

AIX Version 7.3

Commands, r - z



Note

Before using this information and the product it supports, read the information in [“Notices” on page 1465](#).

This edition applies to AIX Version 7.3 and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2021, 2023.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this document.....	xv
Highlighting.....	xv
Case-sensitivity in AIX.....	xv
ISO 9000.....	xv
r.....	1
raddbm Command.....	1
radiusctl Command.....	4
ranlib Command.....	5
raso Command.....	6
ras_logger Command.....	13
rbacqry Command.....	14
rbactoldif Command.....	18
rc Command.....	20
rc.mobip6 Command.....	20
rc.powerfail Command.....	22
rc.wpars Command.....	24
rcp Command	24
rcvdist Command.....	28
rcvpack Command.....	28
rcvstore Command.....	29
rcvtty Command.....	30
rdist Command	31
rdistd Command.....	42
rdump Command.....	43
read Command.....	45
readlvcopy Command.....	47
reboot Command.....	48
rebootwpar Command.....	49
recfgct Command.....	50
recreatevg Command.....	52
recsh Command.....	54
redefinevg Command	55
reducevg Command.....	56
refer Command	58
refile Command.....	60
refresh Command.....	62
refrsrc Command.....	64
refsensor Command.....	66
regcmp Command.....	69
rembak Command.....	70
remove Command.....	72
removevsd Command.....	73
rendev Command.....	74
renice Command.....	75
reorgvg Command.....	76
repl Command.....	78
replacepv Command.....	82
repquota Command.....	83
reset Command.....	84

resetsrc Command.....	85
resize Command.....	89
resource_data_input Information File.....	90
restart-secdapclntd Command.....	94
restbase Command.....	95
restore Command.....	96
restorevgfiles Command.....	107
restvg Command.....	108
restwpar Command.....	112
restwparfiles Command.....	115
resumevsd Command.....	117
rev Command	118
revnetgroup Command.....	119
rexd Daemon.....	120
rexc Command.....	120
rexc Daemon.....	122
rgb Command.....	122
ripquery Command.....	123
rksh Command.....	124
rlogin Command.....	127
rlogind Daemon.....	129
rm Command.....	131
rmail Command.....	133
rmail Command.....	133
rmail Command.....	135
rmC2admin Command.....	139
rmCCadmin Command.....	140
rmcli information file.....	141
rmctrl Command.....	146
rmdomainstatus Command.....	149
rmcifscred Command.....	152
rmcifsmnt Command.....	153
rmclass Command.....	154
rmcluster Command.....	155
rmcomg Command.....	156
rmcondition Command.....	158
rmcondresp Command.....	161
rmcosi Command.....	164
rmdel Command.....	165
rmdev Command	166
rmdir Command.....	168
rmdom Command.....	170
rmf Command.....	170
rmfilt Command.....	172
rmfs Command.....	173
rmgroup Command.....	174
rmiscsi Command.....	176
rmitab Command.....	177
rmkeyserv Command.....	177
rmlpcmd Command.....	178
rmlv Command.....	181
rmlvcopy Command.....	182
rmm Command	184
rmnamsv Command.....	185
rmnfs Command.....	186
rmnfsexp Command.....	186
rmnfsmnt Command.....	188
rmnfsproxy Command.....	188

rmnotify Command.....	189
rmpath Command.....	190
rmprtsv Command.....	192
rmps Command.....	193
rmqos Command.....	194
rmque Command.....	195
rmquedev Command.....	196
rmramdisk Command.....	197
rmresponse Command.....	198
rmrole Command.....	200
rmrpdomain Command.....	202
rmrpnod Command.....	204
rmrset Command.....	207
rmrsrc Command.....	207
rmsecattr Command.....	211
rmsensor Command.....	213
rmserver Command.....	216
rmsmbcmnt Command.....	216
rmsmbcred Command.....	217
rmsock Command.....	218
rmss Command	219
rmssys Command	223
rmt Command.....	224
rmtcpip Command.....	225
rmts Command.....	226
rmtun Command.....	227
rmusil Command.....	228
rmuser Command.....	229
rmvfs Command.....	231
rmvirprt Command.....	231
rmwpar Command.....	232
rmyp Command.....	233
rndc Command.....	234
rndc-confgen Command.....	235
roffbib Command.....	237
rolelist Command.....	238
roleqry Command.....	239
rolerpt Command.....	241
rollback Command.....	242
route Command.....	243
routed Daemon.....	248
rpc.pcnfsd Daemon.....	251
rpcgen Command.....	252
rpcinfo Command.....	253
rpvstat Command.....	256
rpvutil Command.....	261
rrestore Command.....	265
Rsh command.....	268
rsh Command.....	269
rshd Daemon.....	273
rstatd Daemon.....	275
rsyslogd Daemon.....	276
rtcd Daemon.....	278
rtl_enable Command.....	279
runacct Command.....	280
runact Command.....	283
runcat Command.....	287
runlpcmd Command.....	288

rup Command.....	291
ruptime Command.....	292
ruser Command.....	293
rusers Command.....	295
rusersd Daemon.....	296
rvsdrestrict Command.....	296
rwall Command.....	297
rwalld Daemon.....	298
rwho Command.....	299
rwhod Daemon.....	300

S..... 303

sa Command.....	303
sa1 Command.....	305
sa2 Command.....	306
sact Command.....	306
sadc Command	307
sar Command.....	308
savebase Command.....	318
savecore Command	320
savevg Command.....	321
savewpar Command.....	324
scan Command.....	328
sccs Command.....	331
sccsdiff Command.....	334
sccshelp Command.....	335
schedo Command.....	336
scls Command.....	344
script Command.....	345
sctpctrl Command.....	346
sdiff Command.....	354
secldapclntd Daemon.....	357
secldifconv Command.....	359
sectoldif Command.....	361
securetcip Command.....	362
sed Command	363
sedmgr Command.....	368
send Command.....	372
sendbug Command.....	375
sendmail Command.....	376
setclock Command.....	385
setea Command.....	386
setgroups Command.....	387
setkst Command.....	389
setmaps Command.....	391
setrunmode Command.....	394
setsecattr Command.....	395
setsecconf Command.....	401
setsenv Command.....	403
setsyslab Command.....	404
settime Command.....	405
setuname Command.....	407
sh Command.....	408
shconf Command.....	408
shell Command	410
show Command.....	411
showmount Command.....	413

shutacct Command.....	414
shutdown Command.....	415
sisraidmgr Command.....	418
sissasraidmgr Command.....	422
size Command.....	429
skctl Command	431
skulker Command	431
slattach Command.....	432
sleep Command.....	433
slibclean Command.....	434
sliplogin Command.....	434
slocal Command.....	438
slp_srvreg Command.....	439
smbcd Daemon.....	441
smbcstat Command.....	442
smbctune Command.....	444
smdemon.cleau Command.....	446
smit Command	447
smitty Command	450
smrsh Command.....	452
smtctl Command.....	453
snap Command.....	456
snapcore Command.....	464
snapshot Command.....	465
snapsplit Command.....	469
snmpd Daemon.....	470
snmpdv1 Daemon.....	471
snmpdv3 Daemon.....	474
snmpevent Command.....	477
snmpinfo Command.....	480
snmpmibd Daemon.....	483
snmptrap Command.....	486
snmpv3_ssw Command.....	487
sno Command.....	488
sntp Command for NTPv4.....	489
sodebug Command.....	493
soelim Command	495
soestat Command.....	496
sort Command	497
sortbib Command.....	503
sortm Command.....	504
spell Command.....	506
spellin Command.....	508
spellout Command.....	509
splat Command.....	509
split Command	517
splitlvcopy Command.....	518
splitvg Command.....	521
splp Command.....	523
spost Command.....	526
spray Command.....	528
sprayd Daemon.....	529
srcmstr Daemon.....	529
start-secldapclntd Command.....	531
startcondresp Command.....	532
starttrpdomain Command.....	536
starttrpnode Command.....	539
startsrc Command.....	541

startsrc Command.....	545
startup Command.....	547
startvsd Command.....	548
startwpar Command.....	549
startx Command.....	551
statd Daemon.....	553
statvsd Command.....	554
stop-secldapclntd Command.....	556
stopcondresp Command.....	557
stoprpdomain Command.....	560
stoprpnode Command.....	562
stoprsrc Command.....	564
stopsrc Command.....	568
stopvsd Command.....	570
stopwpar Command.....	571
stpinet Method.....	573
strace Command.....	573
strchg Command.....	575
strclean Command.....	576
strconf Command.....	577
strerr Daemon.....	578
strinfo Command.....	579
strings Command.....	580
strip Command.....	582
stripm Command.....	584
strload Command.....	586
strreset Command.....	590
strtune Command.....	591
struct Command.....	593
sttinet Method.....	594
stty-cxma Command.....	594
stty Command.....	598
style Command.....	606
su Command.....	607
subj Command.....	610
sum Command.....	610
suma Command.....	611
suspendvsd Command.....	622
svmon Command.....	623
swap Command.....	642
swapoff Command.....	643
swapon Command.....	644
swcons Command.....	645
swrole Command.....	647
swts Command.....	648
sync Command.....	649
synclvdm Command.....	649
syncroot Command.....	651
syncvg Command.....	652
syncwpar Command.....	655
syscall Command.....	657
sysck Command.....	658
syscorepath Command.....	662
sysdumpdev Command.....	663
sysdumpstart Command.....	669
sysline Command.....	670
syslogd Daemon.....	672

t.....	677
tab Command.....	677
tabs Command.....	677
tail Command.....	681
talk Command.....	683
talkd Daemon.....	684
tapechk Command.....	686
tar Command.....	687
tbl Command	695
tc Command.....	698
tcbck Command.....	700
tcopy Command.....	705
tcpdump Command.....	706
tcptr Command.....	719
tcsd Daemon.....	721
tctl Command.....	722
tee Command.....	725
telinit or init Command	726
telnet Command.....	730
telnetd Daemon.....	743
termdef Command.....	746
test Command	746
tetoldif Command.....	749
tftp Command.....	751
tftpd Daemon.....	756
tic Command.....	759
time Command	759
timed Daemon.....	761
timedc Command.....	763
timex Command.....	765
tip Command.....	766
tnconconsole Command.....	773
tninit Command.....	779
tokstat Command.....	781
topas Command.....	786
topasout Command.....	821
topasrec Command.....	841
topsvcs Command.....	845
topsvcsctrl Command.....	847
touch Command.....	850
tpm_activate Command.....	854
tpm_changeauth Command.....	854
tpm_clear Command.....	856
tpm_clearable Command.....	857
tpm_createek Command.....	858
tpm_enable Command.....	858
tpm_getpubek Command.....	859
tpm_ownable Command.....	860
tpm_present Command.....	860
tpm_restrictpubek Command.....	861
tpm_selftest Command.....	862
tpm_takeownership Command.....	863
tpm_version Command.....	863
tprof Command.....	864
tput Command.....	884
tr Command.....	886

trace Daemon.....	891
traceauth Command.....	901
tracepriv Command.....	902
traceroute Command.....	903
tracesoff Command.....	906
traceson Command.....	907
trbsd Command.....	909
trcctl Command.....	911
trcdead Command	912
trcevgrp Command	914
trcnm Command.....	915
trcrpt Command.....	916
trcstop Command.....	926
trcupdate Command.....	927
troff Command.....	929
trpt Command.....	995
true or false Command.....	999
truss Command.....	1000
trustchk Command.....	1004
tset Command.....	1011
tsh Command.....	1014
tsig-keygen Command.....	1015
tsm Command.....	1016
tsort Command.....	1017
ttt Command.....	1018
tty Command.....	1019
tunchange Command.....	1020
tuncheck Command.....	1022
tundefault Command.....	1024
tunrestore Command.....	1025
tunsave Command.....	1027
turnacct Command.....	1028
turnoff Command.....	1029
turnon Command.....	1029
tvi Command	1030
twconvdict Command.....	1033
twconvfont Command.....	1034
type Command.....	1035

u..... 1037

ucfgif Method.....	1037
ucfginet Method.....	1037
ucfgqos Method.....	1038
ucfgvsd Command.....	1038
uconvdef Command.....	1039
udefif Method.....	1041
udefinet Method.....	1041
udfcheck Command.....	1041
udfcreate Command.....	1042
udflabel Command.....	1043
ugrep Command	1044
uil Command.....	1046
uimx Command.....	1047
ul Command	1049
ulimit Command.....	1049
umask Command.....	1051
umcode_latest Command.....	1053

umount Command.....	1054
umountall Command.....	1057
unalias Command.....	1058
uname Command.....	1059
uncompress Command.....	1061
undefvsvd Command.....	1062
unexpand Command	1063
unfencevsvd Command.....	1064
unget Command (SCCS).....	1066
unifdef Command.....	1067
uniq Command.....	1069
units Command	1071
unlink Command.....	1073
unloadipsec Command.....	1074
unmirrorvg Command.....	1075
unpack Command.....	1077
untab Command	1078
update Command.....	1078
update_iscsi Command.....	1079
updatevsvdnode Command.....	1080
updatevsvdtab Command.....	1082
updatevsvdvg Command.....	1083
uprintfd Daemon.....	1085
uptime Command.....	1086
useradd Command.....	1086
userdel Command.....	1089
usermod Command.....	1091
users Command.....	1093
usrck Command.....	1093
usrprt Command.....	1101
utmpd Daemon.....	1102
uuccheck Command.....	1103
uucico Daemon.....	1104
uuclean Command.....	1106
uucleanup Command.....	1108
uucp Command.....	1110
uucpadm Command.....	1113
uucpd Daemon.....	1116
uudecode Command.....	1116
uudemon.admin Command.....	1117
uudemon.cleanu Command.....	1118
uudemon.hour Command.....	1120
uudemon.poll Command.....	1121
uuencode Command.....	1122
uuid_get command.....	1123
uukick Command.....	1124
uulog Command.....	1125
uuname Command.....	1127
uupick Command.....	1128
uupoll Command.....	1130
uuq Command.....	1131
uusched Daemon.....	1133
uusend Command.....	1134
uusnap Command.....	1135
uustat Command.....	1136
uuto Command	1139
uutry Command	1141
uux Command.....	1142

uuxqt Daemon.....	1146
V.....	1149
vacation Command.....	1149
val Command (SCCS).....	1151
varyoffvg Command.....	1152
varyonvg Command.....	1153
vc Command.....	1157
vfcstat Command.....	1160
vgrind Command	1164
vi Command.....	1166
view Command.....	1189
viosupgrade command.....	1190
vmh Command.....	1202
vmo Command.....	1203
vmstat Command.....	1220
vpdadd Command.....	1234
vpddel Command.....	1236
vsdata1st Command.....	1237
vsdchgserver Command.....	1239
vsdelnode Command.....	1240
vsdelvg Command.....	1241
vsdnode Command.....	1242
vsdsk1st Command.....	1244
vsdvg Command.....	1246
vsdvgs Command.....	1248
W.....	1251
w Command.....	1251
wait Command.....	1252
wall Command.....	1253
wallevent Command.....	1254
watch Command	1256
wc Command.....	1257
what Command.....	1259
whatis Command.....	1260
whatnow Command.....	1261
whereis Command.....	1265
which Command.....	1266
which_fileset Command.....	1267
who Command.....	1268
whoami Command	1271
whodo Command.....	1272
whois Command.....	1274
whom Command.....	1275
wlmassign command.....	1277
wlmcheck command.....	1279
wlmcntrl Command.....	1281
wlmstat Command.....	1284
wol command.....	1289
wparerr Command.....	1290
wparexec Command.....	1291
wpar_reg_script Command.....	1297
wparprnterr Command.....	1298
write Command.....	1298
writesrv Daemon.....	1303
wtmpfix Command.....	1303

wump Command.....	1304
X.....	1307
X Command.....	1307
x_add_fs_fpe Command.....	1320
x_add_nfs_fpe Command.....	1321
x_rm_fpe Command.....	1321
xargs Command	1322
xauth Command.....	1326
xclock Command.....	1330
xcmsdb Command.....	1333
xdm Command.....	1334
xfindproxy Command.....	1349
xfs Command.....	1350
xget Command.....	1352
xhost Command.....	1354
xinit Command.....	1355
xkbcomp Command.....	1357
xkbevd Daemon.....	1359
xkbprint Command.....	1360
xlock Command.....	1362
xlsfonts Command.....	1365
xmbind Command.....	1366
xmkmf Command.....	1367
xmwlm Command.....	1368
xmodem Command.....	1369
xmodmap Command.....	1371
xmpeek Command.....	1373
xmscheck Command.....	1375
xmtopas Command.....	1376
xpr Command.....	1379
xpreview Command.....	1381
xprofiler Command.....	1384
xrdb Command.....	1387
xsend Command.....	1390
xset Command.....	1391
xsetroot Command.....	1395
xss Command.....	1397
xstr Command	1398
xterm Command.....	1399
xwd Command.....	1428
xwud Command.....	1430
y.....	1433
yacc Command.....	1433
yes Command.....	1435
ypbind Daemon.....	1436
ypcat Command.....	1437
ypinit Command.....	1438
ypmatch Command.....	1440
yppasswd Command.....	1441
yppasswdd Daemon.....	1442
yppoll Command.....	1443
yppush Command.....	1444
ypserv Daemon.....	1445
ypset Command.....	1446
ypupdated Daemon.....	1447

ypwhich Command.....	1448
ypxfr Command.....	1450
Z.....	1453
zcat Command.....	1453
zdump Command.....	1454
zic Command.....	1457
Notices.....	1465
Privacy policy considerations.....	1466
Trademarks.....	1467
Index.....	1469

About this document

This document provides users and system administrators with complete information about AIX commands.

Highlighting

The following highlighting conventions are used in this document:

Bold	Identifies commands, subroutines, keywords, files, structures, directories, and other items whose names are predefined by the system. Also identifies graphical objects such as buttons, labels, and icons that the user selects.
<i>Italics</i>	Identifies parameters whose actual names or values are to be supplied by the user.
Monospace	Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or information you should actually type.

Case-sensitivity in AIX

Everything in the AIX operating system is case-sensitive, which means that it distinguishes between uppercase and lowercase letters. For example, you can use the **ls** command to list files. If you type LS, the system responds that the command is not found. Likewise, **FILEA**, **FiLea**, and **filea** are three distinct file names, even if they reside in the same directory. To avoid causing undesirable actions to be performed, always ensure that you use the correct case.

ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.

The following AIX® commands begin with the letter *r*.

raddbm Command

Purpose

Modifies entries in the local database of RADIUS user-authentication information.

Syntax

```
raddbm [ -a Command ] [ -d Database_filename ] [ -e EAP_type ] [ -i Config_filename ] [ -l Load_filename ] [ -n ] [ -p ] [ -t pwd_expire_wks ] [ -u User_ID ] [ -w ]
```

Description

The `raddbm` command is used to create and modify a local database of user-authentication information. The RADIUS server can be configured to use this database as the source of information it uses to authenticate users.

The local database is stored in a file. Data in the file is in a binary tree format to make searches faster. The database file name is specified in the RADIUS `/etc/radius/radiusd.conf` configuration file and has the default value of `dbdata.bin`. You can modify the file name by editing `radiusd.conf` through SMIT.

Each entry has the following fields:

Item	Description
USERID	Specifies the user's ID.
PASSWORD	Specifies the user's password.
PASSWORD_EXPIRATION	Specifies the password expiration time in number of weeks.
EAP_TYPE	Specifies the EAP type allowed for authentication.

Passwords in the database file are not stored in clear text in order to prevent simple password compromise, but the algorithm used to hide the passwords is not considered to be cryptographically secure. The file, `dbdata.bin`, is protected by `root: security` as the owner and group.

Several operations on the local database are supported by the `raddbm` command, including the following:

- Add a user to the database.

To add a user, the command form is:

```
raddbm -a ADD -u User_ID -e EAP_type -t pwd_expire_wks
```

The user's password is prompted from standard input.

The `-e` and `-t` flags are optional. If no value for the `-e` flag is entered, the default value of `none` is used for `EAP_TYPE`, meaning EAP packets are ignored for this user. If no value for the `-t` flag is entered, the default value of `0` is used for `PASSWORD_EXPIRATION`, meaning that password expiration is never checked. The `-p` flag is optional since the `raddbm` command always prompts for a new password when adding a new user.

- Change a user in the database.

To change the user's information in the local database, type the following:

```
raddbm -a CHANGE -u User_ID -p -e EAP_type -t pwd_expire_wks
```

The -e, -p, and -t flags are optional, but at least one must be specified. If the -p flag is used, the raddbm command will prompt for the password.

- Delete a user from the database.

To delete a user's entry from the database, type the following:

```
raddbm -a DELETE -u User_ID
```

- List users in the database.

To list a user's entries in the database, type the following:

```
raddbm -a LIST
raddbm -a LIST -u User_ID
raddbm -a LIST -u User_ID -w
```

The -w and -u flags are optional. If the -w flag is specified, all fields in the user's entry are displayed (except the password, which for security reasons is never displayed).

If the -u flag is specified, the user's information is displayed in colon-separated format. If the -u flag is not specified, all entries in the database are displayed in column format.

- Create a new database.

The RADIUS server ships an empty database in /etc/radius/dbdata.bin. If a user wants to create a new database, at least one user must be added at the time of creation. The form of the command is the following:

```
raddbm -a ADD -u User_ID -e EAP_type -t pwd_expire_wks -n
```

The user's password is prompted from standard input.

The -e and -t flags are optional. They default to EAP_type=NONE and no password expiration checking.

- Load a list of users into the database.

A list of users can be loaded directly into the database using the -l flag. A file must be created for each user that has records in it of the form:

```
"userid" "password"
```

The double quotes must be present.

The file can then be used with the -l flag in the following way:

```
raddbm -l filename
```

Placing user passwords in plain text format in a file is strongly discouraged. This option is provided mainly for testing purposes.

Flags

Item	Description
?	Displays the help screen.
-a <i>Command</i>	Specifies the action to perform. Values are ADD, LIST, DELETE, or CHANGE.
-d <i>Database_filename</i>	Specifies the database file name. Used to override the default database file specified in the radiusd.conf RADIUS configuration file.
-e <i>EAP_type</i>	Specifies the EAP type the user is allowed to use for authentication. Currently, only EAP-TLS, MD5-challenge, or none is supported. The default is none .

Item	Description
-i <i>Config_filename</i>	Specifies the RADIUS configuration file name. Used to override the default <code>/etc/radius/radiusd.conf</code> configuration file.
-l <i>Load_filename</i>	Specifies the file name of the user name and password file to load.
-n	Creates a new database file. Valid only with the ADD command option. If this option is used, all previous information in the database is lost.
-p	Indicates that the user's password is to be changed. For security reasons, the password is prompted from standard input instead of read from the command line.
-t <i>pwd_expire_wks</i>	Specifies the number of weeks the user's password is valid. This flag is valid with the ADD and CHANGE commands. The default is 0, indicating no password expiration. Valid values are from 0 to 52.
-u <i>User_ID</i>	Specifies the user's ID. A valid user ID must be less than 253 characters in length, and can contain letters, numbers, and some special characters. It cannot contain blanks. Duplicate user IDs are not allowed.
-w	Generates a long listing of user information.

Exit Status

This command has the following exit values:

Item	Description
0	The command completed successfully.
>0	An error occurred.

Security

Only the root user or a member of the security group can execute this command.

Examples

1. To create a new local RADIUS database, you must add at least one user. To create the database, type the following:

```
raddbm -a ADD -u user01 -n
```

Note: The `-n` option will overwrite the existing database, destroying the previous contents. The database file created will be named the default name as specified in the `/etc/radius/radiusd.conf` RADIUS configuration file.

2. To add a user to the database, type the following:

```
raddbm -a ADD -u user01
```

The default values of `EAP_TYPE = "none"` and `PASSWORD_EXPIRATION = "0"` are used.

3. To delete a user from the database, type the following:

```
raddbm -a DELETE -u user01
```

4. To change a user's password, type the following:

```
raddbm -a CHANGE -u user01 -p
```

The command prompts for the new password.

5. To display a long listing of all entries in the default database, type the following:

```
raddbm -a LIST -w
```

Passwords are not displayed.

6. To display a particular user's database entry, type the following:

```
raddbm -a LIST -u user01 -w
```

7. To add a list of users from a file, first create the file of users and passwords that has one entry per line and has the form:

```
"userid" "password"
```

Then type the following:

```
raddbm -l Load_filename
```

Restrictions

The RADIUS daemon must be stopped before the `raddbm` command is run. Use the `radiusctl stop` command to stop the daemon. After you have modified the database, restart the daemon with the `radiusctl start` command.

Implementation Specifics

This command is part of the `radius.base` fileset.

Location

`/usr/radius/bin/raddbm`

Standard Input

For security reasons, when a user is added to the database, the user's password is read from standard input instead of from the command line.

Standard Error

If the call to the `raddbm` command fails, an information message is written to standard error.

Files

Item	Description
<code>/usr/radius/bin/raddbm</code>	Location of the <code>raddbm</code> command.
<code>/etc/radius/raddbm.bin</code>	The default database file as specified in the <code>radiusd.conf</code> file.
<code>/etc/radius/radiusd.conf</code>	Specifies the RADIUS configuration values, including the default database file name.

radiusctl Command

Purpose

Starts, stops, or restarts the RADIUS authentication, authorization, and accounting daemons.

Syntax

```
radiusctl start
```

```
radiusctl stop
radiusctl restart
```

Description

The **radiusctl** command starts, stops, or restarts the RADIUS server daemons used for controlling network authentication, authorization, and accounting.

This command enables full EAP-TLS support in the AIX RADIUS server in conjunction with the OpenSSL package shipped on the AIX Expansion Pack media.

The local user database of the AIX RADIUS server can be updated while the server is running, however, new changes take effect only after you restart the system. The **radiusctl** command also makes this possible.

Note: This command deprecates the old method of starting and stopping the AIX RADIUS server (for example, **startsrc -s radiusd**, **stopsrc -s radiusd**, and so on).

Flags

Item	Description
start	Starts running the RADIUS server. Note: If EAP-TLS is enabled through OpenSSL, you are prompted to enter the private key password when you attempt to start or restart the server.
stop	Stops the RADIUS server.
restart	Restarts the RADIUS server whether or not it is currently running. If the server is not running, this flag behaves the same as the start flag.

Examples

1. To start running the AIX RADIUS server, enter the following command:

```
radiusctl start
```

2. To restart an already running AIX RADIUS server, enter the following command:

```
radiusctl restart
```

3. To stop the AIX RADIUS server from running, enter the following command:

```
radiusctl stop
```

ranlib Command

Purpose

Converts archive libraries to random libraries.

Syntax

```
ranlib [ -t ] [ -X {32|64|32_64}] Archive ...
```

Description

The **ranlib** command converts each *Archive* library to a random library. A random library is an archive library that contains a symbol table.

If given the **-t** option, the **ranlib** command only touches the archives and does not modify them. This is useful after copying an archive or using the **-t** option of the **make** command in order to avoid having the **ld** command display an error message about an out-of-date symbol table.

Flags

Item	Description
-t	Touches the named archives without modifying them.
-X mode	Specifies the type of object file ranlib should examine. The <i>mode</i> must be one of the following: 32 Processes only 32-bit object files 64 Processes only 64-bit object files 32_64 Processes both 32-bit and 64-bit object files The default is to process 32-bit object files (ignore 64-bit objects). The <i>mode</i> can also be set with the OBJECT_MODE environment variable. For example, OBJECT_MODE=64 causes ranlib to process any 64-bit objects and ignore 32-bit objects. The -X flag overrides the OBJECT_MODE variable.

Examples

To randomize the archive file `genlib.a`, enter:

```
ranlib genlib.a
```

Files

Item	Description
<code>/usr/ccs/bin/ranlib</code>	Contains the ranlib command.

raso Command

Purpose

Manages Reliability, Availability, Serviceability parameters.

Syntax

```
raso [-p | -r [-K]] [-y] [-o Tunable [= Newvalue] ]
```

```
raso [-p | -r [-K]] [-y] [-d Tunable]
```

```
raso [-p] [-r [-K]] [-y] -D
```

```
raso [-p] [-r | -K] [-F] -a
```

```
raso -h [Tunable]
```

```
raso [-F] [-K] -L [Tunable]
```

```
raso [-F] [-K] -x [Tunable]
```

Note: Multiple [-o](#), [-d](#), [-x](#), and [-L](#) flags can be specified.

Description

Note: The **raso** command requires root authority.

The **raso** command is used to configure Reliability, Availability, Serviceability tuning parameters. The **raso** command sets or displays the current or next-boot values for all RAS tuning parameters. The **raso** command can also be used to make permanent changes or to defer changes until the next reboot. The specified flag determines whether the **raso** command sets or displays a parameter. The **-o** flag can be used to display the current value of a parameter or to set a new value for a parameter.

Understanding the Effect of Changing Tunable Parameters

Misuse of the **raso** command can cause performance degradation or operating system failure. Before modifying any tunable parameter, first read about all of the characteristics of the parameters in the Tunable Parameters section to fully understand the parameter's purpose. Then ensure that the Diagnosis and Tuning sections for this parameter apply to your situation and that changing the value of this parameter could help improve the performance of your system. If the Diagnosis and Tuning sections both contain only N/A, it is recommended that you do not change the parameter unless you are directed to do so by AIX development.

Flags

Table 1. Flags

Item	Description
-a	Displays current value, reboot value (when used with the -x option), Live Update value (when used with the -K option), or permanent values for all tunable parameters (when used with the -p option), one per line in pairs <i>Tunable = Value</i> . For the permanent option, a value is only displayed for a parameter if its reboot and current values are equal. Otherwise, NONE is displayed as the value.
-d <i>Tunable</i>	Resets the <i>Tunable</i> parameters to its default value. If a <i>Tunable</i> parameter, which must be changed because it is not set to its default value, meets one or more of the following sets of criteria, a warning message is displayed and no change is made to the parameter: • The tunable parameter is of type Bosboot or Reboot. • The tunable parameter is of type Incremental and was changed from its default value, and the -x flag is not used in combination. • > The tunable parameter is of type Reboot and is supported across the Live Update operation. <
-D	Resets all <i>Tunable</i> parameters to their default values. If <i>Tunables</i> that need to be changed because they are not set to their default values meet one or more of the following sets of criteria, a warning message is displayed and no change is made: • The tunable is of type Bosboot or Reboot. • The tunable is of type Incremental and was changed from its default value, and -x is not used in combination. • > The tunable parameter is of type Reboot and is supported across the Live Update operation. <
-F	Forces restricted tunable parameters to be displayed when the options -a , -L , or -x are specified alone on the command line. If you do not specify the -F flag, restricted tunables are not included, unless they are named in association with a display option.
-h <i>Tunable</i>	Displays help about the raso command if no <i>Tunable</i> parameter is specified. Displays help about the <i>Tunable</i> parameter if a <i>Tunable</i> parameter is specified.
-K	Sets the tunable parameter value in both <code>/etc/tunables/nextboot</code> and <code>/etc/tunables/nextliveupdate</code> files. The -K flag can be used only with the -x flag. When you specify the -K flag with the -x and -d (or -D) flags, the tunable parameter value is set to its default value in the <code>/etc/tunables/nextboot</code> and <code>/etc/tunables/nextliveupdate</code> files to be used during the next boot or Live Update operations. > When you specify the -K flag with the -L or -x flag, the raso command displays the Live Update values. <

Table 1. Flags (continued)

If you make any change (with **-o**, **-d** or **-D**) to a restricted tunable parameter, it results in a warning message that a tunable parameter of the restricted-use type has been modified. If you also specify the

-r or **-p** options on the command line, you are prompted for confirmation of the change. In addition, at system reboot, the presence of restricted tunables in the `/etc/tunables/nextboot` file, which were modified to a value that is different from their default value (by using a command line that specifies the **-r** or **-p** options), results in an error log entry that identifies the list of these modified tunables.

You can specify a modified tunable value by using the abbreviations K, M, G, T, P, and E to indicate units. The following table shows the prefixes and values that are associated with the number abbreviations.

Table 2. Prefixes and values that are associated with the number abbreviations.

Abbreviation	Prefix	Power of 2
K	Kilo	2 ¹⁰
M	Mega	2 ²⁰
G	Giga	2 ³⁰
T	Tera	2 ⁴⁰
P	Peta	2 ⁵⁰
E	Exa	2 ⁶⁰

Thus, a tunable value of 1024 might be specified as 1 K.

Tunable Parameters type

All the tunable parameters that are manipulated by the tuning commands (**no**, **nfso**, **vmo**, **ioo**, **schedo**, and **raso**) are classified into these categories:

Item	Description
Dynamic	If the parameter can be changed at any time
Static	If the parameter can never be changed
Reboot	If the parameter can be changed only during reboot
Bosboot	If the parameter can be changed only by running bosboot and rebooting the machine
Mount	If changes to the parameter are only effective for future file systems or directory mounts
Incremental	If the parameter can be incremented, except at boot time
Connect	If changes to the parameter are only effective for future socket connections. The parameters must be of type Bosboot.

For parameters of type Bosboot, whenever a change is performed, the tuning commands automatically prompt the user to ask if they want to execute the **bosboot** command. For parameters of type Connect, the tuning commands automatically restart the **inetd** daemon.

The current set of parameters that are managed by the **schedo** command includes only Dynamic and Reboot types.

Compatibility Mode

When running the **raso** command in the pre 5.2 compatibility mode that is controlled by the `pre520tune` attribute of `sys 0`, the reboot values for parameters, except for those of type Bosboot, are not considered because in this mode they are not applied at the boot time. For more information, see [NFS tuning on the client](#) in the *Performance management* guide.

In pre 5.2 compatibility mode, setting reboot values to tuning parameters remains achieved by embedding calls to tuning commands in scripts called during the boot sequence. Therefore, the parameters of type **Reboot** can be set without the **-r** flag so that existing scripts continue to work.

This mode is automatically turned on when a machine is MIGRATED to AIX 5.2. For complete installations, it is turned OFF and the reboot values for parameters are set by applying the content of the `/etc/tunables/nextboot` file during the reboot sequence. Only in that mode the **-r** and **-p** flags are fully functional. For more information, see [Kernel Tuning](#) in the *Performance Tools Guide and Reference*.

Tunable Parameters

For default values and range of values for tunables, refer the **raso** command help (**-h** `<tunable_parameter_name>`).

Table 3. Tunable parameters	
Item	Description
kern_heap_noexec	Purpose Specifies whether no-execute protection must be enabled for the kernel heap. Tuning With protection enabled, any attempt to execute code in the protected heap results in a kernel exception.
kernel_noexec	Purpose Specifies whether no-execute protection must be enabled for kernel data regions. Tuning With protection enabled, any attempt to execute code in the protected regions result in a kernel exception.
> llu_mode <	> Purpose Specifies whether the live library update (LLU) function must be enabled. Tuning The following values are the valid values for the llu_mode tunable parameter: • 0: Specifies that the LLU function is disabled for all the processes regardless of the LLU program attributes. • 1: Specifies that the LLU function is enabled for processes on which the LLU program attributes are enabled. • 2: Specifies that the LLU function is disabled for all the processes unless explicitly enabled with the <code>LDR_CNTRL</code> environment variable For more information, see Live Library Update (LLU). <
mbuf_heap_noexec	Purpose Specifies whether no-execute protection must be enabled for the mbuf heap. Tuning With protection enabled, any attempt to execute code in the protected heap results in a kernel exception.

Table 3. Tunable parameters (continued)

Item	Description
mtrc_commonbufsize	Purpose Specifies the memory trace buffer size for common events of Lightweight Memory Trace (LMT) which provides system trace information for First Failure Data Capture (FFDC). Tuning The default value is based on data generation under a reference system-wide activity, hardware, and system characteristics. The higher limit of the range is based on the hardware and system characteristics and depends on the current value of mtrc_rarebufsize because they share the LMT resource. Recorded events are saved in a system dump, or reported through user commands, or both.
mtrc_enabled	Purpose Defines the Lightweight Memory Trace (LMT) state. Tuning A value of 1 means LMT is enabled. To be effective, any change of state requires a subsequent bosboot and system reboot.
mtrc_rarebufsize	Purpose Specifies the memory trace buffer size for rare events of Lightweight Memory Trace (LMT) which provides system trace information for First Failure Data Capture (FFDC). Tuning The default value is based on data generation under a reference system-wide activity, hardware, and system characteristics. The higher limit of the range is based on the hardware and system characteristics and depends on the current value of mtrace_commonbufsize because they share the LMT resource. Recorded events are saved in a system dump, or reported through user commands, or both.
tprof_cyc_mult	Purpose Specifies the Performance Monitor PM_CYC and software event sampling frequency multiplier as a means to control the trace sampling frequency.
tprof_evt_mult	Purpose Specifies the Performance Monitor PM_* event sampling frequency multiplier as a means to control the trace sampling frequency.

Table 3. Tunable parameters (continued)	
Item	Description
tprof_inst_threshold	Purpose Specifies the minimum number of completed instructions between Performance Monitor event samples as a means to control the trace sampling frequency. Values • The default value is 1000. • The range is 1 to 2G-1. • The type is Dynamic. Diagnosis Not applicable Tuning Not applicable
tprof_evt_system	Purpose Allows or restricts the nonprivileged users from using the system-wide Performance Monitor event-sampling. Values • The default is 0. • The range is 0 - 1. • The type is Dynamic. • The unit is Boolean. Tuning With tprof_evt_system enabled (value 1), the nonprivileged users can use tprof and pmctl commands to perform system-wide Performance Monitor event-sampling. When disabled (value 0), nonprivileged users can perform event-sampling for processes that are started with -y option of tprof and pmctl commands. In the disabled mode, nonprivileged users cannot perform event-sampling of kernel and kernel extensions.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To list the current and reboot value, range, unit, type, and dependencies of all tunable parameters that are managed by the **raso** command, enter the following command:

```
raso -L
```

2. To turn off the Lightweight Memory Trace, enter the following command:

```
raso -r -o mtrc_enabled=0
```

3. To display help for **mtrc_commonbufsize**, enter the following command:

```
raso -h mtrc_commonbufsize
```

4. To set **tprof_inst_threshold** to 10000 after the next reboot, enter the following command:

```
raso -r -o tprof_inst_threshold=10000
```

5. To permanently reset all **raso** tunable parameters to their default values, enter the following command:

```
raso -p -D
```

6. To list the reboot level for all Virtual Memory Manager tuning parameters, enter the following command:

```
raso -r -a
```

ras_logger Command

Purpose

Log an error using the errors template.

Syntax

/usr/lib/ras/ras_logger [**-y** *template-file*]

Description

The **ras_logger** command logs one error, provided in standard input, using the error's template to determine how to log the data. The format of the input is the following:

```
error_label
resource_name
64_bit_flag
detail_data_item1
detail_data_item2
...
```

The **error_label** field is the error's label defined in the template. The **resource_name** field is up to 16 characters in length. The **64_bit_flag** field's values are 0 for a 32-bit error and 1 for a 64-bit error. The **detail_data** fields correspond to the **Detail_Data** items in the template.

Flags

Item	Description
-y <i>template-file</i>	Specifies a template file other than the /var/adm/ras/errtmpl default file.

Examples

1. Log an error. The template is the following:

```
+ F00:
  Catname = "foo.cat"
  Err_Type = TEMP
  Class = 0
  Report = TRUE
  Log     = TRUE
  Alert   = FALSE
  Err_Desc = {1, 1, "Error F00"}
```

```

Prob_Causes = {1, 2, "Just a test"}
User_Causes = {1, 2, "Just a test"}
User_Actions = {1, 3, "Do nothing"}
Detail_Data = 4, {2, 1, "decimal"}, DEC
Detail_Data = W, {2, 1, "hex data"}, HEX
Detail_Data = 100, {2, 1, "long string"}, ALPHA

```

The `ras_logger` input in the **tfile** file appears as follows:

```

F00
resource
0
15
A0
hello world

```

Run the `/usr/lib/ras/ras_logger <tfile` command. This will log the F00 error with **resource** as the resource name. The detail data will consist of 4 bytes set to decimal 15, 4 bytes of hex data set to 0xa0, and the string "hello world". Note that if the value of the 64-bit flag was 1, the hexadecimal data would be 8 bytes set to 0xa0.

2. Multi-item decimal values. The template is the following:

```

+ F00:
    Catname = "foo.cat"
    Err_Type = TEMP
    Class = 0
    Report = TRUE
    Log = TRUE
    Alert = FALSE
    Err_Desc = {1, 1, "Error F00"}
    Prob_Causes = {1, 2, "Just a test"}
    User_Causes = {1, 2, "Just a test"}
    User_Actions = {1, 3, "Do nothing"}
    Detail_Data = 8, {2, 1, "decimal"}, DEC
    Detail_Data = W, {2, 1, "hex data"}, HEX
    Detail_Data = 100, {2, 1, "long string"}, ALPHA

```

The `ras_logger` command enters the following into the **tfile** file:

```

F00
resource
0
15 -15
A0
hello world

```

Note: The decimal data is normally shown by the **errpt** command as two separate values using 4 bytes each. The input therefore contains 15 and -15. This is how it is shown by the **errpt** command.

rbacqry Command

Purpose

Reports a set of used privileges and authorizations for a process.

Syntax

```

/usr/sbin/rbacqry [-T | -C] -n programname [ -i auditfile] -u username [-t timeperiod]
/usr/sbin/rbacqry -c [-s] -u username -S

```

Description

The **rbacqry** command is used as a monitor utility to enable role based access control (RBAC) for applications. The **rbacqry** command reports the privileges and authorizations used by a program after the program is run. It uses the audit subsystem to log the privileges and authorizations of all processes that are created by the program and its spawning process.

The **rbacqry** command operates when the system is operating in the enhanced RBAC mode. The privileges obtained from this report can be assigned to the `innateprivs` and `inheritprivs` attributes for the application by using the **setsecattr** command, which enables the command for RBAC. You can consolidate the privileges for the children of a process and provide it under `inheritprivs` attribute or have separate entries for the children in the `/etc/security/privcmds` file for RBAC enablement.

Notes:

- The **rbacqry** command depends on the audit report that is generated by the AIX auditing subsystem.
- The `rbac` audit class is added to the `/etc/security/audit/config` file when the `rbacqry -c` command is run. The audit class can be configured manually.
- When you are tracing privileges and authorizations by using this utility, assign the `rbac` audit class to a specific user in the `/etc/security/audit/config` file to avoid creating large audit logs.
- The **rbacqry** command does not suggest or provide any RBAC roles as part of the output. The command provides only the privileges and authorizations used by the specified program.
- When you are tracing shell scripts by using the `rbacqry` tool, the shell interpreter (for example: `#!/usr/bin/ksh`) must be mentioned in the first line of the script that is being traced.

Flags

Item	Description
<code>-c</code>	Configures the <code>/etc/security/audit/config</code> file with the <code>rbac</code> class for the specified user.
<code>-C</code>	Provides a set of used privileges and authorization for the process tree in a comma-separated list of the set. This option is mutually exclusive with the <code>-T</code> option.
<code>-i auditfile</code>	Specifies the audit trail file to be processed by the rbacqry command. If not specified, the flag uses the <code>/audit/trail</code> file by default.
<code>-n programname</code>	Specifies the target program name that must be traced for used privileges.
<code>-s</code>	Starts the auditing subsystem if it is turned off. Restarts the audit subsystem if it is already on.
<code>-S</code>	Prints the output in stanza format.
<code>-T</code>	Provides a set of used privileges and authorizations for the processes in a tree format.
<code>-t timeperiod</code>	Accepts a value that is equal to the number of days from when the used privilege report must be generated from the current system date.
<code>-u username</code>	Specifies the user name. This option is required to configure the audit events for the user, and to query the process run by the user.

Exit status

Error Value	Descriptor
<code>= 0</code>	Successful completion
<code>> 0</code>	An error

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To determine the privileges and authorizations that are used by a program, use one of the following methods:

- a. When a program or application is run by a non-root user for which the **rbacqry** command must be run, complete the following steps:

- i) Enable the program for RBAC temporarily under a root or an authorized user's shell, by running the **setsecattr** command:

```
setsecattr -c accessauths=ALLOW_ALL innateprivs=PV_ROOT secflags=FSF_EPS progbath
```

Note: The program path must be a full pathname of the program for which the **rbacqry** command is run.

- ii) Run **setkst -t cmd** (as root or authorized user) to make the changes effective.
- iii) Run the **rbacqry** command under a root or authorized user's shell to configure the user for auditing:

```
rbacqry -c -s -u username
```

- iv) Run the specified program or application as non-root user.
- v) When the program execution completes, run the **rbacqry** command under a root or authorized user's shell to collect used privileges and authorizations:

```
rbacqry -n program -u username (additional options can also be used)
```

- vi) Remove the program entry from the `/etc/security/privcmds` file that was added from step (i) by running the following commands as a root or authorized user:

```
rmsecattr -c progbath; setkst -t cmd
```

- b. When a program or application is executed by a root user (as root login or switching to a root by using the **su** command) and for which the **rbacqry** command must be run, complete these steps:

- i) Run the **rbacqry** command under a root or authorized user's shell to configure the user for auditing:

```
rbacqry -c -s -u root
```

- ii) Run the specified program or application as a root user.
- iii) When the program execution completes, run the **rbacqry** command under a root or authorized user's shell to collect used privileges and authorizations:

```
rbacqry -n program -u root (additional options can also be used)
```

Note: When tracing a program or application that was executed by switching to a root user by using the **su** command after following steps i and ii, run the **rbacqry** command as follows:

```
rbacqry -n program -u user_name (additional options can also be used)
```


- To determine the privileges and authorizations that are used by the **chfs** command (which was executed by user Scooby with aix authorization) and its spawning processes in a tree-formatted output, run the following command:

```
# rbacqry -n chfs -u scooby -T
```

CMD	AUTHORIZATIONS	USED_PRIVS	
chfs	aix.fs.manage.change	PV_FS_RESIZE	
\extendlv	aix.lvm.manage.extend	PV_AU_ADMIN	PV_KER_ACCT
\putlvcb	aix.lvm.manage	PV_FS_MKNOD	PV_PROC_PRIV
		PV_KER_LVM	PV_DEV_QUERY
\lexextendlv	aix.lvm.manage.extend	PV_AU_ADD	PV_AU_PROC
		PV_FS_MKNOD	PV_PROC_PRIV
		PV_KER_ACCT	PV_KER_LVM
		PV_DEV_QUERY	PV_SU_UID
\savebase	aix.system.boot.create	PV_AU_PROC	PV_FS_MKNOD
		PV_PROC_PRIV	PV_KER_ACCT
		PV_KER_LVM	PV_DEV_QUERY
		PV_SU_UID	
\compress	aix.fs.manage.backup	PV_KER_ACCT	PV_SU_UID
.....			

- To display the privileges and authorizations that are used by the **chfs** command (which was executed by user Scooby with aix authorization) from a different audit trail file, run the following command:

```
# rbacqry -u scooby -n chfs -i /audit/trail_example
```

CMD	AUTHORIZATIONS	USED_PRIVS	
chfs	Used_Auth:	PV_DAC_0	PV_FS_CHOWN
	aix.fs.manage.change	PV_FS_RESIZE	
	Checked_Auths:		

- To obtain a comma-separated list of privileges that are used by the **chfs** command (which was executed by user Scooby with aix authorization), run the following command:

```
# rbacqry -n chfs -u scooby -C
```

CMD	AUTHORIZATIONS	USED_PRIVS
chfs	aix.fs.manage.change	PV_FS_RESIZE
extendlv	aix.lvm.manage.extend	PV_AU_ADMIN,PV_KER_ACCT
putlvcb	aix.lvm.manage	PV_FS_MKNOD,PV_PROC_PRIV,PV_KER_LVM,PV_DEV_QUERY
lexextendlv	aix.lvm.manage.extend	PV_AU_ADD,PV_AU_PROC,PV_FS_MKNOD,PV_PROC_PRIV,
savebase	aix.system.boot.create	PV_KER_ACCT,PV_KER_LVM,PV_DEV_QUERY,PV_SU_UID
compress	aix.fs.manage.backup	PV_AU_PROC,PV_FS_MKNOD,PV_PROC_PRIV,PV_KER_ACCT,
		PV_KER_LVM,PV_DEV_QUERY,PV_SU_UID
		PV_KER_ACCT,PV_SU_UID
.....		

This output format is useful when the USED PRIVS set is added to the privileged command in the /etc/security/privcmds database.

Note: The system authorization and custom authorizations can be traced. If the system authorizations must be displayed in the output, a higher authorization (example aix authorization) must be assigned to the user.

5. To configure the user scooby for auditing, run the following command:
- a. To configure the user and to start the auditing for that user, run the following command:

```
#/usr/sbin/rbacqry -c -s -u scooby
```

Audit subsystem started.

- b. To configure the user for auditing without restarting the auditing, run the following command:

```
#/usr/sbin/rbacqry -c -u scooby
```

Note: The user scooby is not traced by the auditing subsystem because the auditing is not restarted. An entry for scooby is made in the /etc/security/audit/config file. You must restart the auditing subsystem manually to allow the auditing to trace the user, or you must run the **rbacqry** command as follows:

```
#/usr/sbin/rbacqry -c -s -u scooby
```

User scooby already configured for audit. Audit subsystem started

6. To show the following stanza for the **-S** format, run the following command:

```
# rbacqry -u scooby -n chfs -S chfs:
      Used_Auth=aix.fs.manage.change
Checked_Auths=
Used_Privs=PV_DAC_0,PV_FS_CHOWN,PV_FS_RESIZE
```

7. To execute the **rbacqry** command without any format options, run the following command:

```
# rbacqry -u scooby -n chfs
CMD      AUTHORIZATIONS      USED_PRIVS
-----
chfs      Used_Auth:
          aix.fs.manage.change  PV_DAC_0      PV_FS_CHOWN
          Checked_Auths:
          PV_FS_RESIZE
```

Note: The *checked_Auths* parameter are blank when no *checked Auths* parameters are present. If not the **rbacqry** command displays the *checked_auths* parameters as follows:

```
# rbacqry -u scooby -n lsuser
CMD      AUTHORIZATIONS      USED_PRIVS
-----
lsuser    Used_Auth:
          ALLOW_ALL          PV_AZ_CHECK   PV_DAC_R
          Checked_Auths:
          aix.security.user.list
          aix.security.user.audit
          aix.security.efs
```

Files

File path

/audit/trail

Description

Specifies the audit file to capture the audit logs.

rbactoldif Command

Purpose

Prints certain role-based access control (RBAC) and Domain role-based access control tables that are defined locally to standard output (**stdout**) in the LDIF format.

Syntax

rbactoldif **-d** *baseDN* [**-s** *tables*]

Description

The **rbactoldif** command reads data from locally defined RBAC tables and prints the result to **stdout** in LDIF format. If redirected to a file, the result can be added to an LDAP server with the **ldapadd** command or the **ldif2db** command.

The **rbactoldif** command reads the **/etc/security/ldap/sectoldif.cfg** file to determine what to name the authorization, role, privileged command, privileged device, and privileged file sub-trees that the data will be exported to. The **rbactoldif** command only exports data to the AUTHORIZATION, ROLE, PRIVCMD, PRIVDEV, and PRIVFILE types defined in the file. The names specified in the file will be used to create sub-trees under the base distinguished name (DN) specified with the **-d** flag. For more information, see the **/etc/security/ldap/sectoldif.cfg** file in *Files Reference* .

Flags

Item	Description
-d <i>baseDN</i>	Specifies the base DN under which the RBAC data is placed.
-s <i>tables</i>	Specifies a set of tables to be read. If you do not specify the -s flag, all of the RBAC and Domain RBAC tables are read. Specify at least one of the following letters, each representing a table name: a Specifies the authorization table. c Specifies the privileged command table. d Specifies the privileged device table. e Specifies the domain table. f Specifies the privileged file table. o Specifies the domain object table. r Specifies the role table. t Specifies the trvi table.

Security

The **rbactoldif** command is owned by root and security group, with mode bits 500.

File Accessed

File	Mode
/etc/security/authorizations	r
/etc/security/roles	r
/etc/security/privcmds	r
/etc/security/privdevs	r

File	Mode
/etc/security/privfiles	r
/etc/security/.rbac_ids	r
/etc/security/domains	r
/etc/security/domobjs	r

Examples

1. To export all of the RBAC and Domain RBAC tables to LDIF format with base DN of cn=aixdata, use the following command:

```
rbactoldif -d cn=aixdata
```

2. To export only the authorization and role tables with base DN of cn=aixdata, use the following command:

```
rbactoldif -d cn=aixdata -s ar
```

3. To export only the domobjs tables with base DN of cn=aixdata, use the following command:

```
rbactoldif -d cn=aixdata -s o
```

rc Command

Purpose

Performs normal startup initialization.

Syntax

```
rc
```

Description

The **rc** command has an entry in the **/etc/inittab** file. The **init** command creates a process for the **rc** command entry in the **/etc/inittab** file. The **rc** command performs normal startup initialization for the system. The contents of **/etc/rc** are installation specific. If all of the necessary operations complete successfully, the file exits with a zero return code that allows the **init** command to start loggers to complete normal initialization and startup.

Note:

1. Many bringup functions such as activating page spaces and mounting filesystems are done by the **rc** command.
2. The root file system is implicitly mounted.

rc.mobip6 Command

Purpose

Enables the system to function as a mobile IPv6 home agent or correspondent node.

Syntax

```
rc.mobip6 { start [ -H ] [ -S ] | stop [ -N ] [ -F ] }
```

Description

The **/etc/rc.mobip6** file is a shell script that, when executed, enables the system to function as a mobile IPv6 home agent or correspondent node. If mobile IPv6 has been configured using system management to start at each system restart, the script will be executed automatically at restart.

Flags

Item	Description
-F	Disables IPv6 forwarding.
-H	Enables the system as a Mobile IPv6 home agent and correspondent node. If this flag is not used, the system will be enabled as a correspondent node only.
-N	Stops the ndpd-router daemon.
-S	Enables checking of IP security authentication.

Exit Status

- 0**
The command completed successfully.
- >0**
An error occurred.

Security

You must have root authority or be a member of the system group to execute this command.

Examples

1. The following example enables the system as a mobile IPv6 home agent and correspondent node:

```
/etc/rc.mobip6 start -H
```

2. The following example enables the system as a mobile IPv6 correspondent node and enables IP security checking:

```
/etc/rc.mobip6 start -S
```

3. The following example disables all mobile IPv6 and IPv6 gateway functionality on the system:

```
/etc/rc.mobip6 stop -N -F
```

4. The following example disables all mobile IPv6 functionality but allows the system to continue functioning as an IPv6 gateway:

```
/etc/rc.mobip6 stop
```

Files

Item	Description
/etc/rc.mobip6	Contains the rc.mobip6 command.

rc.powerfail Command

Purpose

Handles RPA (RS/6000 Platform Architecture) specific EPOW (Environmental and Power Warning) events and shuts down the system if needed, as part of EPOW event handling.

Syntax

rc.powerfail [**-h**] [[**-s**] [**-t** *mm*]] [**-c** [*ss*]]

Description

The `rc.powerfail` command is started by the `/etc/inittab` file when `init` receives a SIGPWR signal from the kernel. The `rc.powerfail` command uses `ioctl()` to determine the state of the system. The `rc.powerfail` command should be called only when an EPOW event has occurred.

The various EPOW events handled by `rc.powerfail` and the corresponding event handling done by `rc.powerfail` are listed in the following table:

EPOW class	Event handling done by rc.powerfail	Example
1 These types of errors are considered non-critical cooling problems by the Operating System.	<code>rc.powerfail</code> warns the users currently logged onto the system through a <code>cron</code> entry which will be walled every 12 hours until the situation disappears.	Redundant Fan Faults. Internal Thermal Problems.
2 These types of errors are considered non-critical power problems by the Operating System.	<code>rc.powerfail</code> warns the users currently logged onto the system through a <code>cron</code> entry which will be walled every 12 hours until the situation disappears.	Redundant AC input fault.
3 These events are critical in nature and the system should be powered down as soon as possible.	<code>rc.powerfail</code> initiates the system shutdown in 10 minutes unless the user has specified some other wait time through the <code>-t</code> option.	Ambient temperature approaching specification limit.
4 These kinds of errors are extreme in nature and need an immediate halting of the system.	<code>rc.powerfail</code> is expected to process this event in 20 seconds. In these cases, <code>rc.powerfail</code> warns the users currently logged onto the system and then immediately halts the system.	Loss of AC input: All the power sources have lost power.
5, 7 These kinds of errors are extreme in nature and should be handled in terms of micro seconds.	Since they should be handled in micro seconds, <code>rc.powerfail</code> will not be handling these events. If <code>rc.powerfail</code> gets control in these conditions, it will continue to wait out the wait time period.	All the fan systems have failed, non redundant power fault.

As previously mentioned, in case of EPOW class 3 events, the `rc.powerfail` command is given approximately 10 minutes prior to shut down of the system. The user can alter this time by using the `-t` option on the `/etc/inittab` file's powerfail entry. Prior to the last 60 seconds, any users still logged-on are sent a message telling them how much time remains until shutdown. If, at any time in the last 60 seconds, the event clears, the system shutdown halts and the users are notified that all errors have cleared. If a shutdown is not desired, the user may add the `-s` option to the command in the `/etc/inittab` file.

Also in case of EPOW class 3 events, `rc.powerfail` will allow executing environment-specific scripts (if any) to be executed before system shutdown. These scripts will be located under `/usr/lib/scripts/`

epow, and `rc.powerfail` will wait for 10 seconds, by default, for their completion. This wait time can be altered using the `-c` option. The value provided through the `-c` option will be taken as the wait time for these scripts, in seconds.

Flags

Item	Description
-h	Gives an information message containing the power status codes and the resulting action. The rc.powerfail -h command shuts down the system if needed, as part of EPOW event handling.
-s	Does not do a system shutdown if there is a power failure in systems with either a battery backup or fan fault. The logged-on users still receive all the appropriate messages, but the actual system shutdown is up to the system administrator. This flag has no effect if a critical power failure is detected.
-t mm	Gives the number of whole minutes until system shutdown in the case of a primary power loss with battery backup or fan fault. This number should be equal to half the length of time guaranteed by the battery backup. This flag has no effect if a critical power failure is detected.
-c ss	Gives the number of seconds to wait for the completion of any environment specify third party scripts to be executed by <code>rc.powerfail</code> , at EPOW 3 situations.

Exit Status

If the system shuts down, no exit value is returned. Otherwise, the **rc.powerfail** command returns the following exit values:

Item	Description
0	Normal condition.
1	Syntax error.
2	halt -q failed
3	shutdown -F failed.
4	An error has occurred. Shut your system down immediately using shutdown -F .
5	An undefined state. Call your Service Representative.

Security

Access Control: root only.

Examples

1. To look at the cause of a power status equal to 3, enter:

```
rc.powerfail -h
```

2. To block system shutdown when non-critical power failures or fan faults occur, enter:

```
chitab "powerfail::powerfail:/etc/rc.powerfail -s >dev/console 2>&1"
```

The next SIGPWR received by **init** will not cause a system shutdown if a non-critical power failure occurs.

3. To change the time until shutdown to 30 minutes, enter:

```
chitab "powerfail::powerfail:/etc/rc.powerfail -t 30 >/dev/console 2>&1"
```

Assuming the condition is not critical, the next SIGPWR received by `init` will have a 30 minute delay until system shutdown.

Files

Item	Description
html	

rc.wpars Command

Purpose

Automatically starts a workload partition.

Syntax

`/etc/rc.wpars`

Description

The `/etc/rc.wpars` command invokes the `startwpars` command on all workload partitions with the `autostart` option (`mkwpar/chwpar -A`) enabled. The `/etc/rc.wpars` command runs automatically each time the system starts.

rcp Command

Purpose

Transfers files between a local and a remote host or between two remote hosts.

Syntax

```
rcp [ -p ] [ -F ] [ -k realm ] [ -m ] { { User@Host:File | Host:File | File } { User@Host:File | Host:File | File |  
User@Host:Directory | Host:Directory | Directory } } [ -r ] { User@Host:Directory | Host:Directory | Directory }  
{ User@Host:Directory | Host:Directory | Directory }
```

Description

The `/usr/bin/rcp` command is used to copy one or more files between the local host and a remote host, between two remote hosts, or between files at the same remote host.

Remote destination files and directories require a specified *Host:* parameter. If a remote host name is not specified for either the source or the destination, the `rcp` command is equivalent to the `cp` command. Local file and directory names do not require a *Host:* parameter.

Note: The `rcp` command assumes that a `:` (colon) terminates a host name. When you want to use a `:` in a filename, use a `/` (slash) in front of the filename or use the full path name, including the `/`.

If a *Host* is not prefixed by a *User@* parameter, the local user name is used at the remote host. If a *User@* parameter is entered, that name is used.

If the path for a file or directory on a remote host is not specified or is not fully qualified, the path is interpreted as beginning at the home directory for the remote user account. Additionally, any metacharacters that must be interpreted at a remote host must be quoted using a `\` (backslash), a `"` (double quotation mark), or a `'` (single quotation mark).

File Permissions and Ownership

By default, the permissions mode and ownership of an existing destination file are preserved. Usually, if a destination file does not exist, the permissions mode of the destination file is equal to the permissions mode of the source file as modified by the **umask** command (a special command in the Korn shell) at the destination host. If the **rcp** command **-p** flag is set, the modification time and mode of source files are preserved at the destination host.

The user name entered for the remote host determines the file access privileges the **rcp** command uses at that host. Additionally, the user name given to a destination host determines the ownership and access modes of the resulting destination file or files.

Using Standard Authentication

The remote host allows access if one of the following conditions is satisfied:

- The local host is included in the remote host **/etc/hosts.equiv** file and the remote user is not the root user.
- The local host and user name is included in a **\$HOME/.rhosts** file on the remote user account.

Although you can set any permissions for the **\$HOME/.rhosts** file, it is recommended that the permissions of the **.rhosts** file be set to 600 (read and write by owner only).

In addition to the preceding conditions, the **rcp** command also allows access to the remote host if the remote user account does not have a password defined. However, for security reasons, the use of a password on all user accounts is recommended.

For Kerberos 5 Authentication

The remote host allows access only if all of the following conditions are satisfied:

- The local user has current DCE credentials.
- The local and remote systems are configured for Kerberos 5 authentication (On some remote systems, this may not be necessary. It is necessary that a daemon is listening to the klogin port).
- The remote system accepts the DCE credentials as sufficient for access to the remote account. See the **kvalid_user** function for additional information.

rcp and Named Pipelines

Do not use the **rcp** command to copy named pipelines, or FIFOs, (special files created with the **mknod -p** command). The **rcp** command uses the **open** subroutine on the files that it copies, and this subroutine blocks on blocking devices like a FIFO pipe.

Restrictions

The SP Kerberos V4 rcp execution path does not support remote-to-remote copy as Kerberos does not support forwarding credentials. The message that you would receive under these circumstances indicates that you do not have tickets and must use **kinit** to log in. The message would be issued from the remote source machine. See the example section for using Kerberos to perform a remote-to-remote copy.

Flags

Item	Description
-p	Preserves the modification times and modes of the source files in the copies sent to the destination only if the user has root authority or is the owner of the destination. Without this flag, the umask command at the destination modifies the mode of the destination file, and the modification time of the destination file is set to the time the file is received. When this flag is not used, the umask being honored is the value stored in the appropriate database. It is not the value that is set by issuing the umask command. The permission and ownership values that result from the umask command do not affect those stored in the database.

Item	Description
-r	Recursively copies, for directories only, each file and subdirectory in the source directory into the destination directory.
-F	Causes the credentials to be forwarded. In addition, the credentials on the remote system will be marked forwardable (allowing them to be passed to another remote system). This flag will be ignored if Kerberos 5 is not the current authentication method. Authentication will fail if the current DCE credentials are not marked forwardable.
-k realm	Allows the user to specify the realm of the remote station if it is different from the local systems realm. For these purposes, a realm is synonymous with a DCE cell. This flag will be ignored if Kerberos 5 is not the current authentication method.
-m	Support for metacharacters in filenames.

Parameters

Item	Description
<i>Host:File</i>	Specifies the host name (<i>Host</i>) and file name (<i>File</i>) of the remote destination file, separated by a : (colon). Note: Because the rcp command assumes that a : (colon) terminates a host name, you must insert a \ (backslash) before any colons that are embedded in the local file and directory names.
<i>User@Host:File</i>	Specifies the user name (<i>User@</i>) that the rcp command uses to set ownership of the transferred file, the host name (<i>Host</i>), and file name (<i>File</i>) of the remote destination file. The user name entered for the remote host determines the file access privileges the rcp command uses at that host.
<i>File</i>	Specifies the file name of the local destination file.
<i>Host:Directory</i>	Specifies the host name (<i>Host</i>) and directory name (<i>Directory</i>) of the remote destination directory. Note: Because the rcp command assumes that a : (colon) terminates a host name, you must insert a \ (backslash) before any colons that are embedded in the local file and directory names.
<i>User@Host:Directory</i>	Specifies the user name (<i>User@</i>) the rcp command uses to set ownership of the transferred file, the host name (<i>Host</i>), and directory name (<i>Directory</i>) of the remote destination directory. The user name entered for the remote host determines the file access privileges the rcp command uses at that host.
<i>Directory</i>	The directory name of the local destination directory.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Security

The remote host allows access only if at least one of the following conditions is satisfied:

- The local user ID is listed as a principal in the authentication database and had performed a **kinit** to obtain an authentication ticket.
- If a **\$HOME/.klogin** file exists, it must be located in the local user's **\$HOME** directory on the target system. The local user must be listed as well as any users or services allowed to **rsh** into this account. This file performs a similar function to a local **.rhosts** file. Each line in this file should contain a principal in the form of "principal.instance@realm." If the originating user is authenticated as one of the principals named in **.klogin**, access is granted to the account. The owner of the account is granted access if there is no **.klogin** file.

For security reasons, any **\$HOME/.klogin** file must be owned by the remote user and only the AIX owner ID should have read and write access (permissions = 600) to **.klogin**.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

In the following examples, the local host is listed in the **/etc/hosts.equiv** file at the remote host.

1. To copy a local file to a remote host, enter:

```
rcp localfile host2:/home/eng/jane
```

The file **localfile** from the local host is copied to the remote host **host2**.

2. To copy a remote file from one remote host to another remote host, enter:

```
rcp host1:/home/eng/jane/newplan host2:/home/eng/mary
```

The file **/home/eng/jane/newplan** is copied from remote host **host1** to remote host **host2**.

3. To send the directory subtree from the local host to a remote host and preserve the modification times and modes, enter:

```
rcp -p -r report jane@host2:report
```

The directory subtree **report** is copied from the local host to the home directory of user **jane** at remote host **host2** and all modes and modification times are preserved. The remote file **/home/jane/.rhosts** includes an entry specifying the local host and user name.

4. This example shows how the root user can issue an **rcp** on a remote host when the authentication is Kerberos 4 on both the target and server. The root user must be in the authentication database and must have already issued **kinit** on the local host. The command is issued at the local host to copy the file, **stuff**, from node **r05n07** to node **r05n05** on an SP.

```
/usr/lpp/ssp/rcmd/bin/rsh r05n07 'export KRBTKFILE=/tmp/rcmdtktt$$; \  
/usr/lpp/ssp/rcmd/bin/rcmdtgt; \  
/usr/lpp/ssp/rcmd/bin/rcp /tmp/stuff r05n05:/tmp/stuff;'
```

The root user sets the **KRBTKFILE** environment variable to the name of a temporary ticket-cache file and then obtains a service ticket by issuing the **rcmdtgt** command. The **rcp** uses the service ticket to authenticate from host **r05n07** to host **r05n05**.

Files

Item	Description
\$HOME/.klogin	Specifies remote users that can use a local user account.
/usr/lpp/ssp/rcmd/bin/rcp	Link to AIX Secure /usr/bin/rsh that calls the SP Kerberos 4 rcp routine if applicable.

Prerequisite Information

Refer to the chapter on security in *IBM Parallel System Support Programs for AIX: Administration Guide* for an overview. You can access this publication at the following Web site: http://www.rs6000.ibm.com/resource/aix_resource

Refer to the "RS/6000 SP Files and Other Technical Information" section of *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for additional Kerberos information. You can access this publication at the following Web site: http://www.rs6000.ibm.com/resource/aix_resource

rcvdist Command

Purpose

Sends a copy of incoming messages to additional recipients.

Syntax

rcvdist [**-form** *File*] *User* ...

Description

The **rcvdist** command forwards copies of incoming messages to users in addition to the original recipient. The **rcvdist** command is not started by a user. The **rcvdist** command is placed in the **.maildelivery** file called by the **/usr/lib/mh/slocal** command.

The **rcvdist** command sends a copy of an incoming message to the user or users specified by the *User* parameter. The default string is located in the **rcvdistcomps** file. This file formats the output from the command and sends it through the **send** command to the ID or alias specified.

You can copy the **rcvdistcomps** file into your local mail directory and change the string to suit your needs. The Message Handler (MH) package uses the **rcvdistcomps** file in your local mail directory first. Otherwise, you can use the **-form** flag to specify a file name that contains the string you want.

Flags

Item	Description
-form <i>File</i>	Specifies the file that formats the command output. The default is the rcvdistcomps file.
-help	Lists the command syntax, available switches (toggles), and version information.

Note: For MH, the name of this flag must be fully spelled out.

Files

Item	Description
\$HOME/.maildelivery	Provides the user with MH instructions for local mail delivery.
\$HOME/.forward	Provides the user with the default message filter.

rcvpack Command

Purpose

Saves incoming messages in a packed file.

Syntax

rcvpack [*File*]

Description

The **rcvpack** command places incoming messages in the packed file specified by the *File* parameter. The **rcvpack** command is not started by the user. The **rcvpack** command is placed in the **\$HOME/.maildelivery** file runs the **rcvpack** command on all incoming messages.

Flags

Item	Description

-help	Lists the command syntax, available switches (toggles), and version information.

Note: For MH, the name of this flag must be fully spelled out.

Files

Item	Description
\$HOME/html	

rcvstore Command

Purpose

Incorporates new mail from standard input into a folder.

Syntax

rcvstore [**+Folder**] [**-create** | **-nocreate**] [**-sequence** *Name*] [**-public** | **-npublic**] [**-zero** | **-nozero**]

Description

The **rcvstore** command adds incoming messages to a specified message directory (a folder). The **rcvstore** command is not started by the user. The **rcvstore** command is placed in the **\$HOME/.maildelivery** file.

You can specify **rcvstore** command flags in the **\$HOME/.mh_profile** file.

Flags

Item	Description
-create	Creates the specified folder in your mail directory if the folder does not exist. This flag is the default.
<u>+Folder</u>	Places the incorporated messages in the specified folder. The default is +inbox .
-help	Lists the command syntax, available switches (toggles), and version information. Note: For MH, the name of this flag must be fully spelled out.
-nocreate	Does not create the specified folder if the folder does not exist.
-npublic	Restricts the specified sequence of messages to your usage. The -npublic flag does not restrict the messages in the sequence, only the sequence. This flag is the default if the folder is write-protected against other users.

Item	Description
-nozero	Appends the messages incorporated by the rcvstore command to the specified sequence of messages. This flag is the default.
-public	Makes the specified sequence of messages available to other users. The -public flag does not make protected messages available, only the sequence. This flag is the default if the folder is not write-protected against other users.
-sequence <i>Name</i>	Adds the incorporated messages to the sequence of messages specified by the <i>Name</i> parameter.
-zero	Clears the specified sequence of messages before placing the incorporated messages into the sequence. This flag is the default.

Profile Entries

Item	Description
Folder-Protect:	Sets the protection level for your new folder directories.
Msg-Protect:	Sets the protection level for your new message files.
Path:	Specifies the <i>UserMHDIRECTORY</i> (the user's MH directory) variable.
Unseen-Sequence:	Specifies the sequences of commands used to keep track of your unseen messages.
Rcvstore:	Specifies flags for the rcvstore program.

Files

Item	Description
\$HOME/.maildelivery	Provides the user with MH instructions for local mail delivery.
\$HOME/.forward	Provides the user with the default message filter.

rcvttty Command

Purpose

Notifies the user of incoming messages.

Syntax

rcvttty [*Command*]

Description

The **rcvttty** command sends the user a message that incoming mail has arrived. The **rcvttty** command is not started by the user. The **rcvttty** command is placed in the **.maildelivery** file.

Flags

Item	Description
-help	Lists the command syntax, available switches (toggles), and version information.
Note: For MH, the name of this flag must be fully spelled out.	

Files

Item	Description
<code>\$HOME/\$HOME/.mh_profile</code>	Contains the MH user profile.

rdist Command

This document describes the old AIX **rdist** command located in the `/usr/bin/rdist` file as well as the new `/usr/sbin/rdist` command which is used with the new **rdistd** daemon.

/usr/bin/rdist Command

Purpose

Remote file distribution client program.

Syntax

To Use a Distribution File

```
rdist [ -n ] [ -q ] [ -b ] [ -D ] [ -R ] [ -h ] [ -i ] [ -v ] [ -w ] [ -y ] [ -f FileName ] [ -d Argument=Value ]  
[ -m Host ] ... [ Name ] ...
```

To Interpret Arguments as a Small Distribution File

```
rdist [ -n ] [ -q ] [ -b ] [ -D ] [ -R ] [ -h ] [ -i ] [ -v ] [ -w ] [ -y ] [ -c Name ... [ Login@ ] Host [ :Destination ]
```

Description



Attention: Do not attempt to use the **rdist** command to send a file larger than 2 Gigabytes in size to a non-AIX machine. Doing so results in undefined behaviors and, in rare cases, the loss of data.

The **rdist** command maintains identical copies of files on multiple hosts. The **rdist** command preserves the owner, group, mode, and modified time of files, if possible, and can update programs that are running. The **rdist** command can receive direction from the following sources:

- The default distribution file, **distfile** file in your **\$HOME** directory.
- A different distribution file, specified by the **-f** flag.
- Command-line arguments that augment or override variable definitions in the distribution file.
- Command-line arguments that serve as a small distribution file.

If you do not use the **-f** flag, the **rdist** command looks for the **distfile** file in your **\$HOME** directory. If it doesn't find a **distfile** file, it looks for **Distfile** file.

The value specified by the *Name* parameter is read as the name of a file to be updated or a subcommand to execute. If you do not specify a value for the *Name* parameter on the command line, the **rdist** command updates all the files and directories listed in the distribution file. If you specify **-** (minus sign) for the *Name* parameter, the **rdist** command uses standard input. If the name of a file specified by the *Name* parameter is the same as the name of a subcommand, the **rdist** command interprets the *Name* parameter as a subcommand.

The **rdist** command requires that a **.rhosts** file be configured on each host. See [File Format for TCP/IP in Files Reference](#) for details.

Note:

1. If the **rdist** command is not present in the `/usr/bin/rdist` directory on a remote machine, create a link from the `/usr/bin/rdist` directory to the actual location of the **rdist** command. This location is usually the `/usr/ucb/rdist` directory.
2. Currently, the **rdist** command can handle only 7-bit ASCII file names.

Flags

Item	Description
-b	Performs a binary comparison and updates files if they differ.
-c	Directs the rdist command to interpret the remaining arguments as a small distribution file. Available arguments are: Name Specifies single name or list of names separated by blanks. The value can be either a file or a subcommand. [Login@]Host Specifies the machine to be updated and, optionally, the login name to be notified of the update. Destination Specifies a file on the remote machine if a single name is specified in the <i>Name</i> argument; specifies a directory if more than one name is specified. Note: Do not use the -c flag with the -f , -d , or -m flag.
-d Argument=Value	Defines the <i>Argument</i> variable as having the value specified by the <i>Value</i> variable. The -d flag defines or overrides variable definitions in the distfile file. The <i>Value</i> variable can be specified as an empty string, one name, or a list of names surrounded by parentheses and separated by tabs or spaces.
-D	Turns on the debugging output.
-f FileName	Specifies the name of the distribution file. If you do not use the -f flag, the default value is the distfile or Distfile file in your \$HOME directory.
-h	Copies the file that the link points to rather than the link itself.
-i	Ignores unresolved links. The rdist command maintains the link structure of files being transferred and warns users if it cannot find all the links.
-m Host	Limits which machines are to be updated. You can use the -m Host option multiple times to limit updates to a subset of the hosts listed in the distfile file.
-n	Prints the subcommands without executing them. Use the -n flag to debug the distfile file.
-q	Operates in quiet mode. The -q option suppresses printing of modified files on standard output.
-R	Removes extraneous files. If a directory is being updated, any files that exist on the remote host but not in the master directory are removed. Use the -R flag to maintain identical copies of directories.
-v	Verifies that the files are up-to-date on all hosts; files that are out-of-date are then displayed. However, the rdist -v command neither changes files nor sends mail. This flag overrides the -b flag when they are used together.
-y	Prevents recent copies of files from being replaced by files that are not as recent. Files are normally updated when their time stamp and size differ. The -y flag prevents the rdist command from updating files more recent than the master file.
-w	Appends the entire path name of the file to the destination directory name. Normally, the rdist command uses only the last component of a name for renaming files, preserving the directory structure of the copied files. When the -w flag is used with a file name that begins with a ~ (tilde), everything except the home directory is appended to the destination name. File names that do not begin with a / (slash) or a ~ (tilde) use the destination user's home directory as the root directory for the rest of the file name.

Distribution File (distfile File)

The distribution file specifies the files to copy, destination hosts for distribution, and operations to perform when updating files to be distributed with the **rdist** command. Normally, the **rdist** command uses the **distfile** file in your **\$HOME** directory. You can specify a different file If you use the **-f** flag.

Entry Formats

Each entry in the distribution file has one of the following formats:

Item	Description
<i>VariableName = NameList</i>	Defines variables used in other entries of the distribution file (<i>SourceList</i> , <i>DestinationList</i> , or <i>SubcommandList</i>).
<i>[Label:] SourceList -> DestinationList SubcommandList</i>	Directs the rdist command to distribute files named in the <i>SourceList</i> variable to hosts named in the <i>DestinationList</i> variable. Distribution file commands perform additional functions.
<i>[Label:] SourceList :: TimeStampFile SubcommandList</i>	Directs the rdist command to update files that have changed since a given date. Distribution file subcommands perform additional functions. Each file specified with the <i>SourceList</i> variable is updated if the file is newer than the time-stamp file. This format is useful for restoring files.

Labels are optional and used to identify a subcommand for partial updates.

Entries

Item	Description
<i>VariableName</i>	Identifies the variable used in the distribution file.
<i>NameList</i>	Specifies a list of files and directories, hosts, or subcommands.

Item	Description
<i>SourceList</i>	Specifies files and directories on the local host for the rdist command to use as the master copy for distribution.
<i>DestinationList</i>	Indicates hosts to receive copies of the files.
<i>SubcommandList</i>	Lists distribution file subcommands to be executed.

The **rdist** command treats new-line characters, tabs, and blanks as separators. Distribution file variables for expansion begin with a \$ (dollar sign) followed by a single character or a name enclosed in {} (braces). Comments begin with a # (pound sign) and end with a new-line character.

Source and Destination List Format

The distribution file source and destination lists comprise zero or more names separated by blanks, as shown in the following format:

[*Name1*] [*Name2*] [*Name3*] ...

The **rdist** command recognizes and expands the following shell metacharacters on the local host in the same way as for the **cs**h command.

- [(left bracket)
-] (right bracket)
- { (left brace)
- } (right brace)
- ((left parenthesis)
-) (right parenthesis)
- * (asterisk)
- ? (question mark)

To prevent these characters from being expanded, precede them with a \ (backslash). The **rdist** command also expands the ~ (tilde) in the same way as for the **cs**h command, but does so separately on the local and destination hosts.

Distribution File Subcommands

Multiple commands to the shell must be separated by a ; (semicolon). Commands are executed in the user's home directory on the host being updated. The **special** subcommand can be used to rebuild private databases after a program has been updated.

The distribution file subcommand list may contain zero or more of the following subcommands:

Item	Description
install <i>Options</i> [<i>OptionalDestName</i>];	Copies out-of-date files and directories. The rdist command copies each source file or directory to each host in the destination list. The available options as specified by the <i>Options</i> variable are the rdist command flags -b , -h , -i , -R , -v , -w , and -y . These options only apply to the files specified by the <i>SourceList</i> variable. When you use the -R flag, nonempty directories are removed if the corresponding file name is absent on the master host. The <i>OptionalDestName</i> parameter renames files. If no install subcommand appears in the subcommand list or the destination name is not specified, the source file name is used. Directories in the path name are created if they do not exist on the remote host. The login name used on the destination host is the same as the local host unless the destination name is of the format <i>login@host</i> .
notify <i>NameList</i> ;	Mails the list of updated files and any errors that may have occurred to the listed names (the <i>NameList</i> parameter). If no @ (at sign) appears in the name, the destination host is appended to the name (<i>name@host</i>).
except <i>NameList</i> ;	Causes the rdist command to update all the files specified by the <i>SourceList</i> entry except for those files specified by the <i>NameList</i> variable.
except_pat <i>NameList</i> ;	Prevents the rdist command from updating any files that contain a string that matches a member of the list specified by the <i>NameList</i> variable.
special <i>NameList</i> " <i>String</i> ";	Specifies shell commands (the " <i>String</i> " variable) to be executed on the remote host after the file specified by the <i>NameList</i> variable is updated or installed. If the <i>NameList</i> variable is omitted, the shell commands are executed for every file updated or installed. The shell variable FILE is set to the current file name before the rdist command executes the " <i>String</i> " variable. The " <i>String</i> " value must be enclosed in " " (double quotation marks) and can cross multiple lines in the distribution file.

Exit Status

This command returns the following exit values:

Item Description

- 0 Successful completion.
- >0 Specifies that an error occurred.

Examples

Examples of the Format: VariableName = NameList

1. To indicate which hosts' files to update, enter a line similar to the following:

```
HOSTS =( matisse root@arpa )
```

where the HOSTS variable is defined to be matisse and root@arpa. The **rdist** command updates files on the hosts matisse and root@arpa. You could use this variable as a destination list.

2. To indicate a name to use as a value for a *SourceList* entry, enter a line similar to the following:

```
FILES = ( /bin /lib/usr/bin /usr/games  
          /usr/include/{*.h,{stand,sys,vax*,pascal,machine}/*.h}  
          /usr/lib /usr/man/man? /usr/ucb /usr/local/rdist )
```

where the FILES value is defined to be the files to be used for the *SourceList* entry.

3. To indicate which files to exclude from the updating process, enter a line similar to the following:

```
EXLIB = ( Mail.rc aliases aliases.dir aliases.pag crontab dshrc  
          sendmail.cf sendmail.fc sendmail.hf sendmail.st uucp vfont)
```

where the EXLIB value is defined as a list of files to exclude from the updating process.

4. To copy all files from **/usr/src/bin** to **arpa** expanding the *namelist* variable so that all files except those present in the *namelist* variable and having .o as an extension are copied:

```
/usr/src/bin ->arpa  
except_pat(\e.o\e ${<namelist> /SCCS\e ${<namelist>}
```

or

```
/usr/src/bin ->arpa  
except_pat(\\.o\e ${<namelist> /SCCS\e ${<namelist>}
```

5. To copy all files from **/usr/src/bin** to **arpa** except those with an .o extension:

```
/usr/src/bin ->arpa  
except_pat(\\.o$ /SCCS$
```

Examples of the Format: [label:] SourceList - DestinationList SubcommandList

1. To copy a source list of files to a destination list of hosts, enter a line similar to the following:

```
${FILES} ->${HOSTS}  
install -R  
except /usr/lib/${EXLIB} ;  
except /usr/games/lib ;  
special /usr/sbin/sendmail "/usr/sbin/sendmail.bz" ;
```

The *[Label:]* entry of the line is optional and not shown here. The \$ (dollar sign) and the {} (braces) cause the file names FILES, HOSTS, and EXLIB to be expanded into the lists designated for them in the previous examples. The rest of the example comprises the subcommand list.

2. To use the *[Label:]* entry, enter the line as follows:

```
srcsL:  
/usr/src/bin -> arpa  
except_pat (\e.o\e$ /SCCS\e$ ) ;
```

The label is `srcsL:` and can be used to identify this entry for updating. The `/usr/src/bin` file is the source to be copied and host `arpa` is the destination of the copy. The third line contains a subcommand from the subcommand list.

3. To use a time-stamp file, enter a line similar to the following:

```
${FILES} :: stamp.cory
notify root@cory
```

The `$` (dollar sign) and `{}` (braces) cause the name specified by `FILES` to be expanded into the list designated for it. The time-stamp file is `stamp.cory`. The last line is a subcommand from the subcommand list.

Files

Item	Description
<code>/usr/bin/rdist</code>	Contains the rdist command.
<code>\$HOME/distfile</code>	Contains a list of subcommands to be read by the rdist command.
<code>/tmp/rdist</code>	Contains an update list. This is a temporary file.

/usr/sbin/rdist Command

This document describes the old AIX **rdist** command located in the `/usr/bin/rdist` file as well as the new `/usr/sbin/rdist` command which is used with the new **rdistd** daemon.

Purpose (/usr/sbin/rdist)

Client program for distributing files remotely.

Syntax (/usr/sbin/rdist)

To Use a Distribution File

```
/usr/sbin/rdist [ -F n ] [ -A num ] [ -a num ] [ -d var=value ] [ -l <local logopts> ] [ -L <remote logopts> ]
[ -f distfile ] [ -M maxproc -m host ] [ -o distops ] [ -t timeout ] [ -p <rdist-path> ] [ -P <transport-path> ]
[ name ... ]
```

To Interpret Arguments as a Small Distribution File

```
/usr/sbin/rdist -Fn -c name ... [ login@ ] host [ :dest ]
```

To Invoke the Old rdist as a Server

```
/usr/sbin/rdist -Server
```

For Version Information

```
/usr/sbin/rdist -V
```

Description (/usr/sbin/rdist)

rdist is a program to maintain identical copies of files over multiple hosts. It preserves the owner, group, mode, and modification time of files if possible and can update programs that are running. The **rdist** command can receive direction from the following sources:

- The distribution file **distfile** in the current directory.
- The standard input if **distfile** is specified as `-`.
- If the **-f** flag is not used, **rdist** looks for the file named *distfile* and *Distfile*.

- If the **-c** flag is used, the trailing arguments are interpreted as a small **distfile**. The equivalent **distfile** is as follows.

```
( filename ... ) -> [user@]host
install          [dest name] ;
```

If no **name** arguments are specified, **rdist** will update all of the files and directories listed in **distfile**. Otherwise, the argument is taken to be the name of a file to be updated or the label of a command to execute. If the label and file names conflict, it is assumed to be a label. These may be used together to update specific files using specific commands.

The **-Server** option provides backward compatibility for older versions of **rdist** which used this option to put **rdist** into server mode. If **rdist** is started with the **-Server** command line option, it will attempt to run the old version of **rdist**. This option will only work if the old **rdist** is located at **/usr/bin/rdist**.

rdist uses an arbitrary transport program to access each target host. The transport program can be specified on the command line with the **-P** flag. If the **-P** flag is not used, **rsh** is taken as the transport program. If the **rsh** method is used and the target host is the string **localhost** and the remote user name is the same as the local user name, **rdist** will attempt to run the following command:

```
/bin/sh -c rdistd -S
```

Otherwise **rdist** will run the following command:

```
rsh host -l remuser rdistd -S
```

In the example preceding, the *host* parameter is the name of the target host, *remuser* is the name of the user to make the connection as and, **rdistd** is the **rdist** server command on the target host.

The transport program must be compatible with the syntax for **rsh**. If not, the transport program should be wrapped in a shell script which understands this command line syntax.

On each target host **rdist** will run the following command:

```
rdistd -S
```

or

```
<rdistd path> -S
```

In the example preceding, the **-p** flag was specified. If **-p** flag is not included, or the *<rdistd path>* is a simple filename, **rdistd** or *<rdistd path>* must be somewhere in the **PATH** of the user running **rdist** on the remote (target) host.

The **rdist** command uses the following environment variables:

Item	Description
TMPDIR	Name of temporary directory to use. Default is /tmp .

Flags (/usr/sbin/rdist)

Item	Description
-A num	Update or install files only if a minimum number of free files (inodes) exists on a filesystem.
-a num	Update or install files only if a minimum amount of free space exists on a filesystem.
-d var = value	Assign <i>value</i> to variable <i>var</i> . This option is used to define or override variable definitions in the distfile . <i>Value</i> can be the empty string, one name, or a list of names surrounded by parentheses and separated by tabs and/or spaces.
-F	Update all clients sequentially without forking child processes.
-f distfile	Use distfile as the distribution file. If distfile is specified as -, read from standard input.
-l logopts	Sets local logging options. See the Message Logging section for more information on the syntax for <i>logopts</i> .
-L logopts	Sets remote logging options. <i>logopts</i> is the same as for local logging except the values are passed to the remote server (rdistd). See the Message Logging section for more information on the syntax of <i>logopts</i> .
-M num	Limit the maximum number of simultaneously running child rdist processes to <i>num</i> . The default is 4.

Item	Description
-m machine	Limits the updating of files to the given machine. Multiple -m arguments can be given to limit updates to a subset of the hosts listed in the distfile .
-n	Display but do not execute commands. Use the -n flag to debug distfile .
-o distopts	Specifies the dist options to enable. <i>distopts</i> is a comma separated list of options. The valid values for <i>distopts</i> are: chknfs If the target filesystem is NFS, do not check or update files. chkreadonly If a file on the target host resides on a read only filesystem, no checking or updating of the file is attempted. chksym If the target on the remote host is a symbolic link, but is not on the master host, the remote target will be a symbolic link. compare Perform a binary comparison and update files if they differ. follow Copy the file that the symbolic link points to rather than the link itself. ignlnks Ignore links which do not resolve. The normal behavior of rdist is to warn the user about unresolved links. nochkowner If the file already exists, do not check user ownership. The file ownership is only set when the file is updated. nochkgroup If the file already exists, do not check group ownership. The file ownership is only set when the file is updated. nochkmode Avoid checking file and directory permission modes. The permission mode is only set when the file is updated. nodescend Do not descend recursively into a directory. Only the existence, ownership, and mode of the directory are checked. noexec Do not check or update executable files that are in a.out format. numchkgroup Use the numeric group id (gid) to check group ownership instead of the group name. numchkowner Use the numeric user id (uid) to check user ownership instead of the user name. quiet Suppress printing files that are being modified on the standard output. remove Remove any files in directories that exist on the remote host that do not exist in the master directory on the local host. savetargets Save files that are updated instead of removing them. Target files that are updated are first renamed from filename to filename.OLD. sparse Enable checking for sparse files. This option adds some additional processing overhead so it should only be enabled for targets likely to contain sparse files.
-o distopts	(<i>dist options, continued</i>): verify Any file on any host that is out of date will be displayed but no file will be changed nor any mail sent. whole The whole file name is appended to the destination directory name. Normally, only the last component of a name is used when renaming files. This will preserve the directory structure of the files being copied instead of flattening the directory structure. For example, rdisting a list of files such as /path/dir1/f1 and /path/dir2/f2 to /tmp/dir would create files /tmp/dir/path/dir1/f1 and /tmp/dir/path/dir2/f2 instead of /tmp/dir/dir1/f1 and /tmp/dir/dir2/f2. younger Files are normally updated if their <i>mtime</i> and <i>size</i> disagree. This option causes rdist not to update files that are younger than the master copy. This can be used to prevent newer copies on other hosts from being replaced. A warning message is printed for files which are newer than the master copy.
-p <rdist-path>	Search for the rdistd server in the given path on the target host.
-P <rdist-path>	Use the transport program as given in <i>transport-path</i> . The <i>transport-path</i> may be a colon separated list of possible pathnames. In this case, the first component of the path to exist is used.
-t timeout	Sets the <i>timeout</i> period (in seconds) for waiting for responses from the remote rdist server. The default is 900 seconds.
-V	Prints the version information and exits.

Message Logging

The **rdist** command provides a set of message facilities, each of which contains a list of message types specifying which types of messages to send to that facility. The local client (**rdist**) and the remote server (**rdistd**) each maintain separate copies of what types of messages to log to what facilities.

The **-l logopts** flag specifies what logging options to use locally on the client. The **-L logopts** flag specifies what logging options to pass to the remote **rdistd** server.

The form of *logopts* should be the following:

```
facility=types:facility= types...
```

The valid facility names are as follows:

stdout

Messages to standard output.

file

Messages are sent to a file. The file name can be specified by the format **file** = *filename* = *types*.

syslog

Messages are sent to the **syslogd** facility.

notify

Messages are sent to the internal **rdistnotify** facility. This facility is used in conjunction with the **notify** ph in a **distfile** to specify what messages are mailed to the **notify** address.

types should be a comma separated list of message types. Each message type specified enables that message level. This is unlike the **syslog** system facility which uses an ascending order scheme. The following are the valid types:

change

Log messages for things that change.

info

Log general information.

notice

Log messages for general info about things that change. This includes things like making directories which are needed in order to install a specific target, but which are not explicitly specified in the **distfile**.

nerror

Log messages for normal errors that are not fatal.

ferror

Log messages for fatal errors.

warning

Log warnings about errors which are not as serious as **nerror** type messages.

verbose

Log messages for more information than normal, but less than debugging level.

debug

Log debugging information.

all

Log all but debug messages.

Distribution File (/usr/sbin/rdist)

The distribution file specifies the files to copy, destination hosts for distribution, and operations to perform when updating files to be distributed with the **rdist** command.

Entry Formats

Each entry in the distribution file has one of the following formats:

VariableName = NameList

Defines variables used in other entries of the distribution file (*SourceList*, *DestinationList*, or *SubcommandList*).

[Label:] SourceList -> DestinationList SubcommandList

Directs the **rdist** command to distribute files named in the *SourceList* variable to hosts named in the *DestinationList* variable.

Distribution file commands perform additional functions.

[Label:] SourceList :: TimeStampFile SubcommandList

Directs the **rdist** command to update files that have changed since a given date. Distribution file subcommands perform additional functions.

Each file specified with the *SourceList* variable is updated if the file is newer than the time-stamp file.

Labels are optional. They are used to identify a command for partial updates.

Entries

Item	Description
<i>VariableName</i>	Identifies the variable used in the distribution file.
<i>NameList</i>	Specifies a list of files and directories, hosts, or subcommands.
<i>SourceList</i>	Specifies files and directories on the local host for the rdist command to use as the master copy for distribution.
<i>DestinationList</i>	Indicates hosts to receive copies of the files.
<i>SubcommandList</i>	Lists distribution file subcommands to be executed.

The **rdist** command treats newline characters, tabs, and blanks as separators. Distribution file variables for expansion begin with a dollar sign followed by a single character or a name enclosed in braces.

Comments begin with a pound sign and end with a newline character.

Source and Destination List Format

The distribution file source and destination lists comprise zero or more names separated by blanks, as shown in the following format:

```
[Name1] [Name2] [Name3] ...
```

The **rdist** command recognizes and expands the following shell metacharacters on the local host in the same way as for the **cs** command.

- [left bracket
-] right bracket
- { left brace
- } right brace
- (left parenthesis
-) right parenthesis
- * asterisk
- ? question mark

To prevent these characters from being expanded, precede them with a backslash. The **rdist** command also expands the tilde in the same way as for the **cs** command, but does so separately on the local and destination hosts. When the **-o whole** option is used with a file name that begins with a tilde, everything except the home directory is appended to the destination name. File names which do not begin with a forward slash or a tilde use the destination user's home directory as the root directory for the rest of the file name.

Distribution File Subcommands

Multiple commands to the shell must be separated by a semicolon. Commands are executed in the user's home directory on the host being updated. The special subcommand can be used to rebuild private databases after a program has been updated.

The distribution file subcommand list may contain zero or more of the following subcommands:

install Options[OptionalDestName];

Copies out-of-date files and directories. The **rdist** command copies each source file or directory to each host in the destination list.

The available options as specified by the *Options* variable are the **rdist** command flags **-b**, **-h**, **-i**, **-R**, **-v**, **-w**, and **-y**.

These options only apply to the files specified by the *SourceList* variable.

When you use the **-R** flag, nonempty directories are removed if the corresponding file name is absent on the master host. The *OptionalDestName* parameter renames files.

If no install subcommand appears in the subcommand list or the destination name is not specified, the source file name is used. Directories in the path name are created if they do not exist on the remote host.

The login name used on the destination host is the same as the local host unless the destination name is of the format login@host.

notify NameList;

Mails the list of updated files and any errors that may have occurred to the listed names (the *NameList* parameter).

If no @ (at sign) appears in the name, the destination host is appended to the name (name@host).

except NameList;

Causes the **rdist** command to update all the files specified by the *SourceList* entry except for those files specified by the *NameList* variable.

except_pat NameList;

Prevents the **rdist** command from updating any files that contain a string that matches a member of the list specified by the *NameList* variable.

special NameList "String";

Specifies shell commands (the "String" variable) to be executed on the remote host after the file specified by the *NameList* variable is updated or installed.

If the *NameList* variable is omitted, the shell commands are executed for every file updated or installed.

The shell variable FILE is set to the current file name before the **rdist** command executes the "String" variable.

The variable REMFILE will contain the full pathname of the remote file that was just updated and the variable BASEFILE will contain the basename of the remote file that was just updated.

The "String" value must be enclosed in double quotation marks and can cross multiple lines in the distribution file.

cmdspecial NameList "String";

The **cmdspecial** command is similar to the **special** command, except it is executed only when the entire command is completed instead of after each file is updated.

The shell variable FILES will contain the list of files. Each file name in the FILES shell variable is separated by a colon.

NFS checks are disabled if a hostname ends in a plus sign. This is equivalent to disabling the **-o chknfs** option just for this one host.

Exit Status (/usr/sbin/rdist)

This command returns the following exit values:

Ite m	Description

0	Successful completion.

>0	An error occurred.

Examples (/usr/sbin/rdist)

1. To indicate which hosts' files to update, enter a line similar to the following:

```
HOSTS =( matisse root@arpa )
```

In the preceding example, the HOSTS variable is defined to be matisse and root@arpa. The **rdist** command updates files on the hosts matisse and root@arpa.

You could use this variable as a destination list.

2. To indicate a name to use as a value for a SourceList entry, enter a line similar to the following:

```
FILES = ( /bin /lib/usr/bin /usr/games  
/usr/include/{*.h,{stand,sys,vax*,pascal,machine}/*.h}  
/usr/lib /usr/man/man? /usr/ucb /usr/local/rdist )
```

In the preceding example, the FILES value is defined to be the files to be used for the *SourceList* entry.

3. To indicate which files to exclude from the updating process, enter a line similar to the following:

```
EXLIB = ( Mail.rc aliases aliases.dir aliases.pag crontab dshrc  
sendmail.cf sendmail.fc sendmail.hf sendmail.st uucp vfont)
```

In the preceding example, the EXLIB value is defined as a list of files to exclude from the updating process.

4. To copy all files from /usr/src/bin to arpa expanding the namelist variable so that all files except those present in the namelist variable and having .o as an extension are copied:

```
/usr/src/bin ->arpa  
except_pat(\e\e.o\e ${<namelist> /SCCS\e ${<namelist>}
```

or

```
/usr/src/bin ->arpa  
except_pat(\\.o\e ${<namelist> /SCCS\e ${<namelist>}
```

5. To copy all files from /usr/src/bin to arpa except those with an .o extension:

```
/usr/src/bin ->arpa  
except_pat(\\.o\ $ /SCCS\ $
```

Examples of the Format: [label:] SourceList - DestinationList SubcommandList

1. To copy a source list of files to a destination list of hosts, enter a line similar to the following:

```
${FILES} ->${HOSTS}  
install -R  
except /usr/lib/${EXLIB} ;  
except /usr/games/lib ;  
special /usr/sbin/sendmail "/usr/sbin/sendmail.bz" ;
```

The [Label:] entry of the line is optional and not shown here. The dollar sign and the braces cause the file names FILES, HOSTS, and EXLIB to be expanded into the lists designated for them in the previous examples.

The rest of the example comprises the subcommand list.

2. To use the [Label:] entry, enter the line as follows:

```
srcsL:  
/usr/src/bin -> arpa  
except_pat (\e\e.o\e $ /SCCS\e $ ) ;
```

The label is srcsL: and can be used to identify this entry for updating. The **/usr/src/bin** file is the source to be copied and host arpa is the destination of the copy.

The third line contains a subcommand from the subcommand list.

3. To use a time-stamp file, enter a line similar to the following:

```
${FILES} :: stamp.cory  
notify root@cory
```

The dollar sign and braces cause the name specified by FILES to be expanded into the list designated for it. The time-stamp file is **stamp.cory**.

The last line is a subcommand from the subcommand list.

Files (/usr/sbin/rdist)

Item	Description
/usr/sbin/rdist	Contains the rdist command at version 6.1.5.
distfile	Contains the input commands.
\$ TMPDIR/rdist*	The temporary file for update lists.

rdistd Command

Purpose

Server program for distributing files remotely.

Syntax

```
rdistd -S  
rdistd -V
```

Description

rdistd is the server program for the **rdist** command. It is normally run by **rdist** through **rsh**.

The **-S** flag ensures that **rdistd** is not accidentally started since it normally resides in a normal user's PATH environment variable.

Flags

Item	Description
-V	Print version information and exit.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Files

Item	Description
<code>/usr/sbin/rdistd</code>	Contains the rdistd server
<code>/usr/bin/rdistd</code>	Symbolic link to <code>/usr/sbin/rdistd</code>

rdump Command

Purpose

Backs up files onto a remote machine's device.

Note: User must have root authority to run this command.

Syntax

rdump [**-b** *Blocks*] [**-B**] [**-c**] [**-d** *Density*] [**-L** *Length*] [**-s** *Size*] [**-u**] [**-w**] [**-W**] [**-Level**] **-f** *Machine:Device* [*FileSystem* | *DeviceName*]

Description

The **rdump** command copies file systems by i-node from your local machine to a remote machine. The files are copied, using the **backup** command format, to a device on the remote machine. The device is accessed by using a remote server on the remote machine. You must have root authority to execute the **rdump** command. You must also define a local machine running the **rdump** command in the `/.rhosts` file of the target remote machine.

To back up a file system, specify the **-Level** and *FileSystem* parameters to indicate the files you want to back up. You can use the **-Level** parameter to back up either all files on the system (a full backup) or only the files that have been modified since a specific full backup (an incremental backup). The possible levels are 0 to 9. If you do not supply a level, the default level is 9. A level 0 backup includes all files on the file system. A level *n* backup includes all files modified since the last level *n* - 1 (*n* minus 1) backup. The levels, in conjunction with the **-u** flag, provide a method of maintaining a hierarchy of incremental backups for each file system.

Note:

1. Use the **-u** flag when you perform an incremental backup (the **-Level** parameter) to ensure that information regarding the last date, time, and level of each incremental backup is written to the `/etc/dumpdates` file.
2. If the **rmt** command on the remote machine is not in `/usr/sbin/rmt`, then a link will need to be created on the remote machine from `/usr/sbin/rmt` to its actual location (usually `/etc/rmt`).

Flags

Item	Description
-b <i>Blocks</i>	Specifies the number of blocks to write in a single output operation. If you do not specify the <i>Blocks</i> variable, the rdump command uses a default value appropriate for the physical device selected. Larger values of the <i>Blocks</i> variable result in larger physical transfers to tape devices.
-B	Terminates the command without querying the user when an error occurs. If you specify the -B flag, the rdump command returns a nonzero value.
-c	Specifies that the tape is a cartridge format, not a 9-track format.

Item	Description
-d <i>Density</i>	Specifies the density of the tape in bits-per-inch (bpi). This value is used in calculating the amount of tape used per volume. If you do not specify a value for the <i>Density</i> variable, the default density is 1600 bpi. When using the -c flag without specifying a tape density, the default density is 8000 bpi.
-f <i>Machine:Device</i>	Specifies the <i>Machine</i> variable as the hostname of the remote machine. To send output to the named device, specify the <i>Device</i> variable as a file name (such as the /dev/rmt0 file). The <i>Device</i> variable should specify only tape devices.
-L <i>Length</i>	Specifies the length of the tape in bytes. This flag overrides the -c , -d , and -s flags. You can specify the size with a suffix of b, k, m, or g to represent Blocks (512 bytes), Kilo (1024 bytes), Mega (1024 Kilobytes), or Giga (1024 Megabytes), respectively. To represent a tape length of 2 Gigabytes, type the following: -L 2g .
-s <i>Size</i>	Specifies the size of the tape in feet using the <i>Size</i> variable. If you do not specify a tape size, the default size is 2300 feet. When using the -c flag without specifying a tape size, the default size is 1700 feet. When the tape drive reaches the specified size, the rdump command waits for the tape to be changed.
-u	Updates the time, date, and level of the remote backup in the /etc/dumpdates file. This file provides the information needed for maintaining incremental backups.
-w	Currently disabled.
-W	Displays the file systems found in the /etc/dumpdates files.
-Level	Specifies the remote backup level (0 to 9). The default value of the <i>Level</i> variable is 9.
-?	Displays the usage message.

Parameters

Item	Description
<i>DeviceName</i>	Specifies the physical device name (the block or raw name).
<i>FileSystem</i>	Specifies the name of the directory on which the file system is usually mounted. The rdump command reads the /etc/filesystems file for the physical device name. If you do not specify a file system, the default is the root (/) file system.

Exit Status

This command returns the following exit values:

Ite m	Description
0	Indicates that the command completed successfully.
>0	Indicates that an error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command

Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To back up files in the **/usr** directory on your local machine to a remote machine, type:

```
rdump -u -0 -fcanine:/dev/rmt0 /usr
```

The **-u** flag tells the system to update the current backup level records in the **/etc/dumpdates** file. The **-Level** flag set to backup level 0 specifies that all the files in the **/usr** directory are to be backed up. The ID of the remote machine is **canine** and the device is the **/dev/rmt0** device.

2. To back up files in the **/usr** directory on your local machine to a remote machine using an 8mm, 2.3GB tape, type:

```
rdump -fcanine:/dev/rmt0 -L 2200m /usr
```

Note: 2.2GB is used here instead of 2.3GB to avoid hitting the actual end of the tape.

3. To back up files in the **/usr** directory on your local machine to a remote machine using 0.25-inch tape, type:

```
rdump -fcanine:/dev/rmt0 -c /usr
```

When using the **-c** flag, the **rdump** command defaults to the correct size and density values for 0.25-inch tape.

Files

Item	Description
/etc/dumpdates	Contains logs of the most recent remote dump dates.
/etc/filesystems	Contains information on file systems.
/dev/rhd4	Contains the device where the default file system (root) is located.
/usr/sbin/rdump	Contains the rdump command.

read Command

Purpose

Reads one line from standard input.

Syntax

```
read [ -p ][ -r ][ -s ][ -u[ n ] ][ VariableName?Prompt ]  
[ VariableName ... ]
```

Description

The **read** command reads one line from standard input and assigns the values of each field in the input line to a shell variable using the characters in the **IFS** (Internal Field Separator) variable as separators. The *VariableName* parameter specifies the name of a shell variable that takes the value of one field from the line of input. The first shell variable specified by the *VariableName* parameter is assigned the value of the first field, the second shell variable specified by the *VariableName* parameter is assigned the value of the second field, and so on, until the last field is reached. If the line of standard input has more fields than there are corresponding shell variables specified by the *VariableName* parameter, the last shell variable

specified is given the value of all the remaining fields. If there are fewer fields than shell variables, the remaining shell variables are set to empty strings.

Note: If you omit the *VariableName* parameter, the variable `REPLY` is used as the default variable name. The setting of shell variables by the **read** command affects the current shell execution environment.

Flags

Item	Description
-p	Reads input from the output of a process run by the Korn Shell using <code> &</code> (pipe, ampersand). Note: An end-of-file character with the -p flag causes cleanup for this process so that another can be spawned.
-r	Specifies that the read command treat a <code>\</code> (backslash) character as part of the input line, not as a control character.
-s	Saves the input as a command in the Korn Shell history file.
-u [<i>n</i>]	Reads input from the one-digit file descriptor number, <i>n</i> . The file descriptor can be opened with the ksh <code>exec</code> built-in command. The default value of the <i>n</i> is 0, which refers to the keyboard. A value of 2 refers to standard error.

Parameters

Item	Description
<i>VariableName?Prompt</i>	specifies the name of one variable, and a prompt to be used. When the Korn Shell is interactive, it will write the prompt to standard error, and then perform the input. If <i>Prompt</i> contains more than one word, you must enclose it in single or double quotes.
<i>VariableName...</i>	specifies one or more variable names separated by white space.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	Detected end-of-file character or an error occurred.

Examples

1. The following script prints a file with the first field of each line moved to the end of the line:

```
while read -r xx yy
do
    print printf "%s %s/n" $yy $xx
done < InputFile
```

2. To read a line and split it into fields, and use "Please enter: " as a prompt, type:

```
read word1?"Please enter: " word2
```

The system displays:

```
Please enter:
You enter:
hello world
```

The value of the *word1* variable should have "hello" and *word2* should have "world".

3. To create a co-process, then use `print -p` to write to the co-process, and use `read -p` to read the input from the co-process, type:

```
(read; print "hello $REPLY")
print -p "world"
read -p line
```

The value of the *line* variable should have "hello world".

4. To save a copy of the input line as a command in the history file, type:

```
read -s line < input_file
```

If *input_file* contains "echo hello world", then "echo hello world" will be saved as a command in the history file.

readlvcopy Command

Purpose

Reads a specific mirror copy of a logical volume.

Syntax

```
readlvcopy -d device [ -c copy | -C copy | -b | -p mirrorpool ] [ -n number_of_blocks ] [ -o outfile ] [ -s skip ] [ -S seek ]
```

Description

Flags

Item	Description
-d <i>device</i>	Logical volume special device file to be read from
-c <i>copy</i>	Requested mirror copy to read from. Valid values are 1, 2, or 3 for the first, second, or third copy of the data. Data is read even if the logical partition has been marked stale. The default is the first copy of the data.
-C <i>copy</i>	Requested mirror copy to read from. Valid values are 1, 2, or 3 for the first, second, or third copy of the data. Stale logical partitions are not read.
-b	Read mirror copy marked as online backup.
-n <i>number_of_blocks</i>	Number of 128K blocks to read
-o <i>outfile</i>	Destination file. The default is <i>stdout</i>
> > -p <i>mirrorpool</i>	Read from the copy of the data that is associated to the specified mirror pool. Data is read even if the logical partition has been marked stale. The default is the first copy of the data. If the specified mirror pool does not exist in the volume group, the readlvcopy command fails.
-s <i>skip</i>	Number of 128K blocks to skip into <i>device</i> .

Item	Description
-S seek	Number of 128K blocks to seek into <i>outfile</i>

Example

>|

To read the first 128 KB block of data from the copy, which is associated with the **mp1** mirror pool, of the device **/dev/rlv00**, run the following command:

```
readlvcopy -d /dev/rlv00 -p mp1 -n 1
```

|<

reboot or fastboot Command

Purpose

Restarts the system.

Syntax

```
{ reboot | fastboot } [ -l ] [ -n ] [ -q ] [ -t mmddHHMM [ yy ] ]
```

Description

The **reboot** command can be used to perform a reboot operation if no other users are logged into the system. The **lsattr** command and enter `lsattr -D -l sys0`. The default value is **true**. To reset the autorestart attribute value to **false**, use the **/var/adm/wtmp**, the login accounting file. These actions are inhibited if the **-l**, **-n**, or **-q** flags are present.

The **fastboot** command restarts the system by calling the **reboot** command. The **fsck** command runs during system startup to check file systems. This command provides BSD compatibility.

Flags

Item	Description
-l	Does not log the reboot or place a shutdown record in the accounting file. The -l flag does not suppress accounting file update. The -n and -q flags imply -l .
-n	Does not perform the sync command. Use of this flag can cause file system damage.
-q	Restarts without first shutting down running processes. Note: A file system synchronization will not occur if the -q flag is used. If you want the file system to be synchronized, manually run the sync command or use the shutdown -r command.

Item	Description
-t	Shuts down the system immediately and then restarts the system on the specified date. A valid date has the following format: <i>mmddHHMM</i> [<i>yy</i>] where: mm Specifies the month. dd Specifies the day. HH Specifies the hour. MM Specifies the minute. yy Specifies the year (optional). The two digit value represents the value of the year in the current century (based on the system time). For example, if the current year based on the systems time is 1985, 99 means 1999 and if the current year is 2005 then 99 means 2099 and 04 means 2004.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

To shut down the system without logging the reboot, enter:

```
reboot -l
```

Files

Item	Description
/etc/rc	Specifies the system startup script.
/var/adm/wtmp	Specifies login accounting file.

rebootwpar Command

Purpose

Stops and restarts a system workload partition.

Restriction: You cannot run the **rebootwpar** command on an application workload partition.

Syntax

```
rebootwpar [ -F | -h ] [ -N | -t seconds ] [ -v ] WparName
```

Description

The **rebootwpar** command stops and restarts the workload partition.

Flags

Item	Description
-F	Specifies a forced stop.
-h	Specifies a hard stop.
-N	Specifies there is no timeout for halt.
-t seconds	Specifies the halt timeout in seconds.
-v	Verbose mode.
WparName	Specifies the workload partition name.

Security

Access Control: Only the root user can run this command.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

To reboot the workload partition called "MyWpar", enter:

```
rebootwpar MyWpar
```

recfgct Command

Purpose

Reconfigures the Reliable Scalable Cluster Technology (RSCT) subsystems.

Syntax

```
/opt/rsct/install/bin/recfgct [ -i Node_ID | -n | -s | -h ]
```

Description

Attention: Use this command with extreme caution.

The **recfgct** command is used to remove all RSCT data under the `/var/ct` directory, generate a new node ID, and make it appear as if the RSCT components are just installed. Because of the destructive nature of this command, it is not normally started by the system administrator. You must use this command *only* if you need to remove a duplicate node ID or if an IBM® service representative instructs you to use it.

When RSCT is first installed, a node ID is automatically generated. The node ID is a true random 64-bit number. Each system where RSCT is installed must have a unique node ID. If a copy of an operating system image (OSI) that has RSCT installed on it is installed on another system, the other system has the same node ID as the system from which the copy is made. This is referred to as *cloning*. For AIX platform, cloning is typically performed using such AIX-supported commands and procedures as **mksysb**. These

commands and procedures call `recfgct` automatically. For other platforms, the `recfgct` command must be run immediately after a cloned OSI is installed.

If the `-s` flag is specified, after all data under the `/var/ct` directory is removed, the node ID contained in the `/etc/ct_node_id` file is used to re-create the `/var/ct/cfg/ct_node_id` file.

Flags

-i Node_ID

Specifies the node ID that must be used. The node ID must contain 9 - 16 hexadecimal characters.

-n

Generates a new node ID. It is the default behavior if no option is specified.

-s

Saves the node ID.

-h

Writes the command usage statement to standard output and then exits.

Restrictions

The `-h` flag is supported on the following RSCT levels:

- RSCT 2.4.9.1 (or later) for AIX 5.3
- RSCT 2.5.1.1 (or later) for AIX 6.1 and all Linux® platforms
- RSCT 3.1.0.0 (or later) for AIX 7.1 and later

If you try to run the `recfgct -h` command on a prior version of RSCT, the `-h` flag is ignored and all RSCT data is removed.

Files

/etc/ct_node_id

Contains a copy of the RSCT node ID

/var/ct/cfg/ct_node_id

Contains the RSCT node ID

Standard output

When the `-h` flag is specified, this command usage statement is written to standard output and then the command exits.

Exit status

0

The command ran successfully.

1

The command did not run successfully.

Security

Privilege control: only the `root` user must have execute (x) access to this command.

Implementation specifics

This command is part of the `rsct.core` fileset for the AIX operating system and `rsct.core-3.1.0.0-0.platform.rpm` package for Linux, Solaris, and Windows operating system, where *platform* is `i386`, `ppc`, `ppc64`, `s390`, or `x86_64`.

Location

/opt/rsct/install/bin/recfgct

Examples

1. After installing a cloned operating system image, enter:

```
/opt/rsct/install/bin/recfgct
```

recreatevg Command

Purpose

Re-creates a volume group that exists on a specified set of disks. Imports and varies on the volume group.

Syntax

recreatevg [-y *VGname*] [-p] [-f] [-Y*Lv_Prefix* | -l *LvNameFile*] [-L *Label_Prefix*] [-n] [-V *MajorNumber*] [-k] [-d] [-O] *PVname...*

Description

The **recreatevg** command re-creates a volume group on a set of disks that are duplicated from another set of disks that belong to a specific volume group. This command overcomes the problem of duplicated Logical Volume Manager (LVM) data structures and identifiers that are caused by a disk duplication process. This command allocates new physical volume identifiers (PVID) for the member disks, as the PVIDs are also duplicated by the disk duplication. Similarly, duplicated logical volume members are given new names with the user-specified prefixes.

Notes:

- The **recreatevg** command removes all logical volumes that fully or partially exist on the physical volumes that are not specified on the command line. Mirrored logical volumes can be an exception (see the **-f** flag).
- The **recreatevg** command warns, if the log for the logical volume of a file system does not exist on the disks that are specified on the command line.
- The **recreatevg** command fails, if the input list does not match the list that is compiled from the Volume Group Descriptor Area (VGDA).
- The set of disks in the list must have consistent VGDA data. The **recreatevg** command does not fix VGDA problems.
- When re-creating a concurrent-capable volume group, the volume group is not varied on when the **recreatevg** command completes. The new volume group must be varied on manually.
- >|The **recreatevg** command keeps the original passphrase authentication methods in all encrypted logical volumes in the volume group.|<
- >|The **recreatevg** command deletes the key file authentication method of the encrypted logical volumes.|<
- >|New encrypted logical volume is unlocked automatically if the original logical volume has key server authentication method. If the original logical volume has Platform keystore (PKS) method and the new logical volume is unlocked, the **recreatevg** command adds the PKS method to the new logical volume if PKS key slots are available.|<
- >|The **recreatevg** command deletes the key server authentication method in all encrypted logical volumes in the volume group. Use the **-k** flag to keep the key server authentication method.|<

Flags

Item	Description
-d	Instead of completely re-creating the VG, the d flag causes the recreatevg command to create only new PVIDs for the specified disks and update the LVM metadata with the new PVIDs. Logical volumes (LVs) names and labels is not changed and the VG is not imported. This flag is incompatible with other flags except the -O flag.
-f	Re-creates a volume group (VG) from a subset of disks. Only those disks and the logical volumes (LVs) that is present entirely on this subset of disks is present in the re-created VG. All other disks and LVs from the original VG is deleted in the re-created VG. For mirrored LVs, only LV mirror copies with physical partitions allocated on the deleted disks are removed. Therefore, a mirrored LV can be re-created with fewer mirror copies when one of copies is present on the subset of disks.
-k	Keeps the key server authentication method in all encrypted logical volumes in the volume group.
-l <i>LvNameFile</i>	Changes logical volume names to the name specified by <i>LvNameFile</i> . Entries must be in the format LV:NEWLV1. All logical volumes that are not included in <i>LvNameFile</i> are re-created with default system generated names. NEWLV1 name can be the same as LV name in the <i>LvNameFile</i> stanza (LV:NEWLV1) to leave the logical volume with the same name.
-L <i>Label_Prefix</i>	Changes the labels of logical volumes on the VG being re-created to this prefix. You must modify the <code>/etc/filesystems</code> file path manually if a simple modification of the mount point is not enough to define the stanza uniquely. Specifying / (slash) as the <i>Label_Prefix</i> , leaves the label in the logical volume unchanged.
-n	Specifies that after recreatevg the volume group is imported but varied off. Default is imported and varies on.
-p	Disables the automatic generation of the new PVIDs. If the -p flag is used, you must ensure that there are no duplicated PVIDs on the system. All the disks that are hardware that is mirrored must have their PVIDs changed to a unique value.
-O	Forces the volume group to be re-created and varied on even if the metadata on the disk indicates that this volume group is varied on in another node. See the <code>varyonvg</code> command for detailed information.
-V <i>MajorNumber</i>	Allows the major number of the volume group to be specified rather than having the major number generated automatically.
-y <i>VGname</i>	Allows the volume group name to be specified rather than having the name generated automatically. Volume group names must be unique system wide and can range from 1 to 15 characters. The name cannot begin with a prefix already defined in the PdDv class in the Device Configuration database for other devices. The new volume group name is sent to standard output.

Item	Description
-Y <i>Lv_Prefix</i>	Causes the logical volumes on the volume group that is being re-created to be renamed with this prefix. The total length of the prefix and the logical volume name must be less than or equal to 15 characters. If the length exceeds 15 characters, the logical volume is renamed with a default name. The default name must comply to the following conditions: • Cannot begin with a prefix that is already defined in the PdDv class of the Device Configuration database. • Cannot use a name that is already used by another system. Specifying NA as the <i>Lv_Prefix</i>, leaves all the logical volume names unchanged.

Security

Access Control: You must have root authority to run this command.

Examples

1. To re-create a volume group that contains three physical volumes, enter the command:

```
recreatevg hdisk1 hdisk2 hdisk3
```

The volume group on hdisk1, hdisk2, and hdisk3 is re-created with an automatically generated name, which is displayed.

2. To re-create a volume group on hdisk1 with the new name testvg, enter the command:

```
recreatevg -y testvg hdisk1
```

3. To re-create a volume group on hdisk14, re-create all logical volumes in that volume group, and rename them with the prefix newlv, enter the command:

```
recreatevg -Y newlv hdisk14
```

Files

Item	Description
/usr/sbin	Directory where the recreatevg command is present.

recsh Command

Purpose

Invokes the recovery shell.

Syntax

recsh

Description

When the **libc.a** library is moved or renamed, an error message Killed will be displayed from the shell as there is no **libc.a** library available for the system to load and run the utilities. The **recsh** command

invokes recovery shell, which provides the ability to rename **libc.a** library if it is accidentally moved. It uses an alternative **libc.a** library that is shipped with the system.

Note: This is a recovery shell and users should not use **recsh** as default shell.

Examples

1. If libc.a is renamed accidentally then the system will be in an unstable state where in execution of any utility will not be possible. To recover at this point, type:

```
recsh; cp -p libc.a.new /usr/lib/libc.a; exit
```

Location

/usr/bin/recsh

Files

Item	Description
/usr/bin/recsh	Specifies the path name to the recovery shell.

redefinevg Command

Purpose

Redefines the set of physical volumes of the given volume group in the device configuration database.

Syntax

redefinevg { **-d** *Device* | **-i** *Vgid* } *VolumeGroup*

Description

During normal operations the device configuration database remains consistent with the Logical Volume Manager (LVM) information in the reserved area on the physical volumes. If inconsistencies occur between the device configuration database and the LVM, the **redefinevg** command determines which physical volumes belong to the specified volume group and re-enters this information in the device configuration database. The **redefinevg** command checks for inconsistencies by reading the reserved areas of all the configured physical volumes attached to the system.

Note: To use this command, you must either have root user authority or be a member of the **system** group.

Flags

Item	Description
-d <i>Device</i>	The volume group ID, <i>Vgid</i> , is read from the specified physical volume device. You can specify the <i>Vgid</i> of any physical volume belonging to the volume group that you are redefining.
-i <i>Vgid</i>	The volume group identification number of the volume group to be redefined.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Example

To redefine rootvg physical volumes in the Device Configuration Database, enter a command similar to the following:

```
redefinevg -d hdisk0 rootvg
```

Files

Item	Description
<code>/usr/sbin/synclvodm</code>	Contains the synclvodm command.

reducevg Command

Purpose

Removes physical volumes from a volume group. When all physical volumes are removed from the volume group, the volume group is deleted.

Syntax

```
reducevg [ -d ] [ -f ] [ -w ] VolumeGroup PhysicalVolume ...
```

>|

```
reducevg [ -d ] -p [ -w ] mirrorpool,...VolumeGroup
```

|<

Description



Attention: You can use the **reducevg** command while the volume group is in concurrent mode. However, if you run this command while the volume group is in concurrent mode and the result is the deletion of the volume group, then the **reducevg** command fails.

The **reducevg** command removes one or more physical volumes that are represented by the *PhysicalVolume* parameter from the *VolumeGroup*. When you remove all physical volumes in a volume group, the volume group is also removed. The volume group must be varied on before it can be reduced.

All logical volumes residing on the physical volumes that are represented by the *PhysicalVolume* parameter must be removed with the **rm1v** command or the -d flag before starting the **reducevg** command.

Note:

1. To use this command, you must either have root user authority or be a member of the system group.
2. Sometimes a disk is removed from the system without first running **reducevg VolumeGroup PhysicalVolume**. The VGDA still has this removed disk in its memory, but the *PhysicalVolume* name no longer exists or reassigned. To remove references to this missing disk you can still use **reducevg**, but with the Physical Volume ID (PVID) instead of the disk name: **reducevg VolumeGroup PVID**.

3. You cannot use the **reducevg** command on a snapshot volume group.
4. You cannot use the **reducevg** command on a volume group that has an active firmware that is assisted dump logical volume.
5. The **reducevg** command discards any background space reclamation process that is running for the physical volumes that are removed from the volume group. To identify whether a space reclamation is running, you can use the **lvmstat** command with **-r** option.
6. [>|](#) Delete the key server and key file authentication methods of all encrypted logical volumes that are configured on the physical volumes before you run the **reducevg** command. The **reducevg** command deletes only the Platform keystore (PKS) authentication method before the encrypted logical volume is deleted.[|<](#)

For volume groups created on AIX 5.3 and varied on without the **varyonvg -M** flag, **reducevg** will dynamically raise the logical track group size for the volume group if necessary to match the common max transfer size of the remaining physical volumes.

You might also use the System Management Interface Tool (SMIT) **smit reducevg fast** path to run this command.

Flags

Table 4. Flags

Item	Description
-d	Deallocates the existing logical volume partitions and then deletes resultant empty logical volumes from the specified physical volumes. User confirmation is required unless the -f flag is added.
	Attention: The reducevg command with the -d flag automatically deletes all logical volume data on the physical volume before removing the physical volume from the volume group. If a logical volume spans multiple physical volumes, the removal of any of those physical volumes may jeopardize the integrity of the entire logical volume.
-f	Removes the requirement for user confirmation when the -d flag is used.
> -p	Removes the physical volumes that are associated with the specified mirror pool. <<
-w	Waits for the completion of the space reclamation process.
	Note: The wait time for space reclaim completion depends on the number of physical partitions.

Security

RBAC users

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To remove physical volume **hdisk1** from volume group **vg01**, enter:

```
reducevg vg01 hdisk1
```

2. To remove physical volume **hdisk1** and all residing logical volumes from volume group **vg01** without user confirmation, enter the following command.



Attention: The **reducevg** command with the **-d** flag automatically deletes all logical volume data before removing the physical volume.

```
reducevg -d -f vg01 hdisk1
```

The physical volume hdisk1 and all residing logical volumes are removed.

3. **>** To remove the physical volumes that are associated with the mp1 mirror pool from the vg00 volume group, run the following command:

```
reducevg -p mp1 vg00
```

|<

Files

Table 5. Files

Item	Description
/usr/sbin/ reducevg	Directory where the reducevg command resides.
/tmp	Directory where the temporary files are stored and while the command is running.

refer Command

Purpose

Finds and inserts literature references in documents.

Syntax

refer [**-b**] [**-e**] [**-P**] [**-S**] [**-a** *Number*] [**-B** *Label.Macro*] [**-c** *Keys*] [**-f** *Number* | **-k** *Label* | **-l** *Letter, Digit*] [**-n**] [**-p** *Reference*] [**-s** *Keys*] [*File ...*]

Description

The **refer** command is a preprocessor for the **nroff** or the **troff** command. The **refer** command finds and formats references for footnotes or endnotes. It is also the basis for a series of programs designed to index, search, sort, and print standalone bibliographies or other data entered in the appropriate form.

Given an incomplete citation with sufficiently precise phs, the **refer** command searches a bibliographic database for references containing these phs anywhere in the title, author, journal, and so on. The input file (or else standard input) is copied to standard output, except for lines enclosed by the .[(period, left bracket) and .] (period, right bracket) delimiters. Lines enclosed by the delimiters are assumed to contain phs and are replaced by information from the bibliographic database. The user can search different databases, override particular fields, or add new fields. The reference data, from whatever source, is assigned to a set of **troff** command strings. Macro packages, such as the **ms** macro package, print the finished reference text from these strings. By default, references are flagged by footnote numbers.

To use your own references, put them in the format described in the Example section. These references can be accessed either by using the **-p** flag or by setting the **REFER** environment variable to those reference files. The references can be searched more rapidly by running the **indxbib** command on them before using the **refer** command. If you do not index, a linear search is made. When the **refer** command is used with any of the preprocessor commands (**eqn**, **neqn**, or **tbl** command), the **refer** command should be issued first, to minimize the volume of data passed through pipes.

Note: Anytime you edit a reference file, you must reissue the **indxbib** command on that file. If you do not use the **indxbib** command, remove any **.ia**, **.ib**, **.ic**, and **.ig** files associated with

that reference file; otherwise, you will get a too many hits error message from the **refer** command.

The **refer** command and associated programs expect input from a file of references composed of records separated by blank lines. A record is a set of fields (lines), each containing one kind of information. Fields start on a line beginning with the **%** (percent sign), followed by a key letter, a space character, and finally the contents of the field, and continue until the next line, starting with a **%** (percent sign). The output ordering and formatting of fields is controlled by the macros specified for the **nroff** and **troff** commands (for footnotes and endnotes), or the **roffbib** command (for standalone bibliographies). For a list of the most common key letters and their corresponding fields, see the **addbib** command.

Flags

Item	Description
-b	Bare mode: do not put any flags in text (either numbers or labels).
-e	Instead of leaving the references where encountered, accumulates them until a sequence of the following form is encountered: <pre>. [\$LIST\$.]</pre> then writes out all references collected so far.
-P	Places punctuation marks after the reference signal, rather than before. The punctuation marks are locale-specific and are defined in the refer message catalog .
-S	Produces references in the natural or social science format.
-a Number	Reverses the first specified number of author names (Jones, J. A. instead of J. A. Jones). If the <i>Number</i> variable is omitted, all author names are reversed.
-B Label.Macro	Specifies bibliography mode. Takes a file composed of records separated by blank lines and turns that file into troff command input. The specified label is turned into the specified macro, with the <i>Label</i> variable value defaulting to %X and the <i>Macro</i> variable value defaulting to .AP (annotation paragraph).
-c Keys	Capitalizes, with SMALL CAPS, the fields whose key letters are in the <i>Keys</i> variable. For example, Jack becomes JACK.
-f Number	Sets the footnote number to the specified number instead of the default of 1. With labels rather than numbers, this flag has no effect. See the -k flag and the -l flag.
-k Label	Instead of numbering references, uses labels as specified in a reference data line beginning with %Label . By default, the <i>Label</i> variable value is L .
-l Letter,Digit	Instead of numbering references, uses labels made from the senior author's last name and the year of publication. Only the first specified letters of the last name and the last specified digits of the date are used. If either the <i>Letter</i> variable or the <i>Digit</i> variable is omitted, the entire name or date, respectively, is used.
-n	Does not search the default /usr/share/dict/papers/Ind file. If the REFER environment variable is set, the specified file is searched instead of the default file. In this case, the -n flag has no effect.
-p Reference	Takes the <i>Reference</i> variable as a file of references to be searched. The default file is searched last.

Item	Description
-s <i>Keys</i>	Sorts references by fields whose key letters are specified by the <i>Keys</i> variable string. Renames reference numbers in text accordingly. Implies the -e flag. The key letters specified by the <i>Keys</i> variable can be followed by a number to indicate how many such fields are used, with q + (plus sign) indicating a very large number. The default value is AD , which sorts first by senior author and then by date. For example, to sort on all authors and then title, enter -sA+T . It is important to note that blank spaces at the end of lines in bibliography fields cause the records to sort and reverse incorrectly. Sorting large numbers of references can cause a core dump.

Example

Following is an example of a **refer** command entry:

%A M.E. Lesk

%T Some Applications of Inverted Indexes on the UNIXSystem

%B *UNIXProgrammer's Manual*

%V 2b

%I Bell Laboratories

%C Murray Hill, NJ

%D 1978

Files

Item	Description
/usr/share/dict/papers/Ind	Contains the default reference file.
/usr/lbin/refer	Contains companion programs.

refile Command

Purpose

Moves files between folders.

Syntax

refile [**-src** *+Folder*] [**-draft**] [**-file** *File*] [*Messages*] [**-nolink** | **-link**] [**-nopreserve** | **-preserve**] **+Folder ...**

Description

The **refile** command moves messages between folders. If you do not specify a source folder, the **refile** command uses the current folder as the source. If you specify a destination folder that does not exist, the system requests permission to create it.

The **refile** command also copies messages from one folder to another. When moving a message, by default, the system does not keep a copy of the message in the original folder. To leave a copy behind, use the **-preserve** flag.

Flags

Item	Description
-draft	Copies the current draft message from your mail directory.
-file <i>File</i>	Copies the specified file. The file must be in valid message format. Use the inc command to format and file new messages correctly.
+Folder	Copies the messages to the specified folder. Any number of folders can be specified.
-help	Lists the command syntax, available switches (toggles), and version information. Note: For MH, the name of this flag must be fully spelled out.
-link	Leaves the messages in the source folder or file after they are copied.
<i>Messages</i>	Specifies the messages to be copied. You can specify several messages, a range of messages, or a single message. Use the following references to specify messages: Number Number of the message. Sequence A group of messages specified by the user. Recognized values include: all All the messages in a folder. cur or . (period) Current message. This is the default. first First message in a folder. last Last message in a folder. next Message following the current message. prev Message preceding the current message. /DT> If the -link and all flags are used together, the current message in the current folder does not change. Otherwise, if a message is specified, the refiled message becomes the current message.
-nolink	Removes the messages from the source folder or file after they are copied. This flag is the default.
-nopreserve	Renums the messages that are copied. Renumbering begins with a number one higher than the last message in the destination folder. This flag is the default.
-preserve	Preserves the message numbers of copied messages. If messages with these numbers already exist, the refile command issues an error message and does not alter the contents of the folders.
-src <i>+Folder</i>	Identifies the source folder. By default, the system uses the current folder.

Profile Entries

The following entries are part of the *UserMHDDirectory/.mh_profile* file:

Item	Description
Current-Folder:	Sets the default current folder.
Folder-Protect:	Sets the protection level for your new folder directories.
Path:	Specifies the <i>UserMhDirectory</i> .
rmmproc:	Specifies the program used to remove messages from a folder.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To refile the current message from the current folder and place it in a new folder called meetings, enter:

```
refile +meetings
```

The system responds with a message similar to the following:

```
Create folder "/home/jeanne/Mail/meetings"?
```

Enter y to create the folder. A copy of the original message is not retained in the current folder.

2. To copy the current message from the current folder and to the meetings folder, enter:

```
refile -link +meetings
```

The original message remains in the current folder.

3. To refile the current message draft into the test folder, enter:

```
refile -draft +test
```

A copy of the message draft is not retained in the current folder.

4. To refile the current message from the current folder and into several folders, enter:

```
refile +tom +pat +jay
```

A copy of the message is not retained in the current folder.

Files

Item	Description
\$HOME/.mh_profile	Sets the MH user profile.
/usr/bin/refile	Contains the refile command.

refresh Command

Purpose

Requests a refresh of a subsystem or group of subsystems.

Syntax

refresh [**-h** *Host*] { **-g** *Group*| **-p** *SubsystemPID*| **-s** *Subsystem*}

Description

The **refresh** command sends the System Resource Controller a subsystem refresh request that is forwarded to the subsystem. The refresh action is subsystem-dependent.

Note: The **refresh** command is unsuccessful if the communication method for the subsystems is signals.

Flags

Item	Description
-g <i>Group</i>	Specifies a group of subsystems to refresh. The refresh command is unsuccessful if the <i>Group</i> name is not contained in the subsystem object class.
-h <i>Host</i>	Specifies the foreign <i>Host</i> machine on which this refresh action is requested. The local user must be running as "root". The remote system must be configured to accept remote System Resource Controller requests. That is, the srcmstr daemon (see /etc/inittab) must be started with the -r flag and the /etc/hosts.equiv or .rhosts file must be configured to allow remote requests.
-p <i>SubsystemPID</i>	Specifies a particular instance of the subsystem to refresh.
-s <i>Subsystem</i>	Specifies a subsystem to refresh. The <i>Subsystem</i> name can be the actual subsystem name or the synonym name for the subsystem. The refresh command is unsuccessful if <i>Subsystem</i> name is not contained in the subsystem object class.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To refresh the a group, like tcpip, enter:

```
refresh -g tcpip
```

2. To refresh a subsystem, like xntpd, enter:

```
refresh -s xntpd
```

Files

Item	Description
/etc/objrepos/SRCsubsys	Specifies the SRC Subsystem Configuration Object Class.
/etc/services	Defines the sockets and protocols used for Internet services.
/dev/SRC	Specifies the AF_UNIX socket file.
/dev/.SRC-unix	Specifies the location for temporary socket files.

refsrc Command

Purpose

Refreshes the resources within the specified resource class.

Syntax

```
refsrc [-h] [-TV] resource_class
```

Description

The `refsrc` command refreshes the resources within the specified resource class. Use this command to force the Resource Monitoring and Control (RMC) subsystem to detect new instances of resources in cases where the configuration could be altered by operating system commands (`mkfs`, for example).

This command makes a request to the RMC subsystem to refresh the configuration of the resources within a resource class. The request is actually performed by the linked resource manager.

Any application that is monitoring resources in the specified resource class may receive events as the configuration is refreshed.

Flags

-h

Writes the command's usage statement to standard output.

-T

Writes the command's trace messages to standard error. For your software-service organization's use only.

-V

Writes the command's verbose messages to standard output.

Parameters

resource_class

Specifies the resource class name.

Security

The user needs read permission for the *Resource_class* specified in `refsrc` to run `refsrc`. Permissions are specified in the access control list (ACL) file on the contacted system. See the *RSCT: Administration Guide* for information about the ACL file and how to modify it.

Exit Status

0

The command has run successfully.

1

An error occurred with RMC.

2

An error occurred with the command-line interface (CLI) script.

3

An incorrect flag was specified on the command line.

4

An incorrect parameter was specified on the command line.

5

An error occurred with RMC that was based on incorrect command-line input.

Environment Variables

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. CT_IP_AUTHENT only has meaning if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon to monitor and control the resources and resource classes. The management scope determines the set of possible target nodes where the resources and resource classes can be monitored and controlled. The valid values are:

0

Specifies *local* scope.

1

Specifies *local* scope.

2

Specifies *peer domain* scope.

3

Specifies *management domain* scope.

If this environment variable is *not* set, *local* scope is used.

Implementation Specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) fileset for AIX.

Standard Output

When the -h flag is specified, this command's usage statement is written to standard output.

The command output and all verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

1. To refresh the configuration of the resources in class IBM.FileSystem, enter:

```
refsrc IBM.FileSystem
```

Location

/opt/rsct/bin/refsrc

refsensor Command

Purpose

Refreshes a sensor or a microsensor defined to the resource monitoring and control (RMC) subsystem.

Syntax

To refresh a sensor:

```
refsensor [-a | -n host1[,host2...] | -N { node_file | "-" }] [-h ] [-v | -V] sensor_name
```

To refresh a microsensor:

```
refsensor -m [-a | -n host1[,host2...] | -N { node_file | "-" }] [-h ] [-v | -V] sensor_name
```

Description

The `refsensor` command refreshes a sensor or microsensor resource that is defined to the RMC subsystem. *Sensors* and *microsensors* are RMC resources with attributes that can be monitored. Sensors and microsensors must be monitored for `refsensor` to run successfully.

A sensor can be refreshed using `refsensor` in one of two ways: either by running the sensor command that is defined for the sensor resource or by specifying values for specific sensor attributes. A microsensor can be refreshed using `refsensor` to query the values of the microsensor's load module. Use the `-m` flag to refresh a microsensor.

When the `refsensor` command runs, it does not affect the interval, if any, that is defined (using `mksensor`) for running the sensor command or for querying the microsensor load module. That is, if a monitored sensor or microsensor is being updated every 60 seconds, running `refsensor` does not cause the interval timer to be reset to 60 seconds.

The `refsensor` command runs on any node. If you want `refsensor` to run on all of the nodes in a domain, use the `-a` flag. If you want `refsensor` to run on a subset of nodes in a domain, use the `-n` flag. Instead of specifying multiple node names using the `-n` flag, you can use the `-N node_file` flag to indicate that the node names are in a file. Use `-N "-"` to read the node names from standard input.

If Cluster Systems Management (CSM) is installed on your system, you can use CSM defined node groups as node name values to refer to more than one node. For information about working with CSM node groups and using the CSM `nodegrp` command, see the *CSM: Administration Guide* and the *CSM: Command and Technical Reference*.

To have `refsensor` update specific sensor attributes, specify one or more `attr=value` parameters. Only the attributes specified will be updated. No other sensor attributes will be updated. The sensor attributes that can be specified as parameters are:

Float32

The type float32 attribute for this sensor resource

Float64

The type float64 attribute for this sensor resource

Int32

The type int32 attribute for this sensor resource

Int64

The type int64 attribute for this sensor resource

Quantum

The type quantum attribute for this sensor resource

String

The type string attribute for this sensor resource

Uint32

The type uint32 attribute for this sensor resource

Uint64

The type uint64 attribute for this sensor resource

For example, to update the Int32 and Float32 sensor attributes for the sensor named Sensor1, enter:

```
refsensor Sensor1 Int32=45 Float32=7.8
```

Microsensor attributes cannot be updated separately.

Flags

-a

Refreshes sensors that match the specified name on all nodes in the domain.

The CT_MANAGEMENT_SCOPE environment variable determines the cluster scope. If CT_MANAGEMENT_SCOPE is not set, first the management domain scope is chosen if it exists, then the peer domain scope is chosen if it exists, and then local scope is chosen, until the scope is valid for the command. The command will run once for the first valid scope found. For example, if both a management domain and a peer domain exist, `refsensor -a` with CT_MANAGEMENT_SCOPE not set will run in the management domain. In this case, to run in the peer domain, set CT_MANAGEMENT_SCOPE to 2.

-m

Specifies that the resource to be refreshed is a microsensor resource.

-n *host1[,host2...]*

Specifies one or more nodes on which the sensor should be refreshed. By default, the sensor is refreshed on the local node. This flag is only appropriate in a management domain or a peer domain.

-N { *node_file* | "-" }

Specifies that node names are read from a file or from standard input.

Use -N *node_file* to indicate that the node names are in a file.

- There is one node name per line in *node_file*
- A number sign (#) in column 1 indicates that the line is a comment
- Any blank characters before a node name are ignored
- Any characters after a node name are ignored

Use -N "-" in a management domain or a peer domain to read the node names from standard input.

-h

Writes the command's usage statement to standard output.

-v | -V

Writes the command's verbose messages to standard output.

Parameters

sensor_name

Specifies the name of the sensor to be refreshed.

attr=value

Specifies which sensor attributes will be refreshed and the values to which they will be set.

Security

To refresh sensors using this command, you need write permission for the IBM.Sensor resource class.

To refresh microsensors using this command, you need write permission for the IBM.MicroSensor resource class.

Permissions are specified in the access control list (ACL) file on the contacted system. See the *RSCT: Administration Guide* for details on the ACL file and how to modify it.

Exit Status

0

The command has run successfully.

1

An incorrect combination of flags and parameters has been entered.

4

The sensor is not monitored and cannot be refreshed.

6

No sensor resources were found.

n

Based on other errors that can be returned by the RMC subsystem.

Environment Variables

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the resource monitoring and control (RMC) daemon on the specified host. If this environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. CT_IP_AUTHENT only has meaning if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon to monitor and control the resources and resource classes. The management scope determines the set of possible target nodes where the resources and resource classes can be monitored and controlled.

The valid values are:

0

Specifies *local* scope.

1

Specifies *local* scope.

2

Specifies *peer domain* scope.

3

Specifies *management domain* scope.

If this environment variable is *not* set, *local* scope is used.

Implementation Specifics

This command is part of the `rsct.core` fileset for AIX and `rsct.core-3.1.0.0-0.platform.rpm` package for Linux, Solaris, and Windows, where *platform* is *i386*, *ppc*, *ppc64*, *s390*, or *x86_64*.

Examples

1. To refresh the sensor called `Sensor1` so that its defined sensor command is run, enter:

```
refsensor Sensor1
```

2. To refresh the sensor called `Sensor1` so that `Int32` is set to 50, `Float32` is set to 123.45, and `String` is set to "test input", enter:

```
refsensor Sensor1 Int32=50 Float32=123.45 String="test input"
```

3. To refresh the sensor called `Sensor1` on the nodes that are listed in the `/u/joe/common_nodes` file so that `Sensor1`'s defined sensor command is run, enter:

```
refsensor -N /u/joe/common_nodes Sensor1
```

where `/u/joe/common_nodes` contains:

```
# common node file
#
node1.myhost.com    main node
node2.myhost.com    backup node
```

4. To refresh the microsensor called `IBM.Sensor1` so that the attribute values are queried using the defined microsensor load module, enter:

```
refsensor -m IBM.Sensor1
```

Location

`/opt/rsct/bin/refsensor`

regcmp Command

Purpose

Compiles patterns into C language **char** declarations.

Syntax

regcmp [-] *File* [*File* ...]

Description

The **regcmp** command compiles the patterns in *File* and places output in a *File.i* file, or a *File.c* file when the **-** option is specified. The resulting compiled patterns are initialized **char** declarations. Each entry in *File* must be a C variable name followed by one or more blanks, followed by a pattern enclosed in **"** (double quotation marks).

The output of the **regcmp** command is C source code. A resulting *File.i* file can be included in C programs, and a resulting *File.c* file can be a file parameter to the **cc** command.

A C language program that uses the output of the **regcmp** command should use the **regex** subroutine to apply it to a string.

In most cases, the **regcmp** command makes unnecessary the use of the **regcmp** subroutine in a C language program, saving execution time and program size.

Flag

Item	Description

-	Places the output in a <i>File.c</i> file. The default is to put the output in <i>File.i</i> .

Examples

1. To compile the patterns in stdin1 and the patterns in stdin2, enter :

```
regcmp stdin1 stdin2
```

This creates the stdin1.i and stdin2.i files.

2. To creates stdin1.c and stdin2.c files, enter:

```
regcmp - stdin1 stdin2
```

Note: Assuming that the same stdin1 and stdin2 files are used in both examples, the resulting stdin1.i and stdin1.c files are identical, and the resulting stdin2.i and stdin2.c files are identical.

File

Item	Description
<code>/usr/ccs/bin/regcmp</code>	Contains the regcmp command.

rembak Command

Purpose

Sends a print job to a queue on a remote server.

Syntax

```
rembak -S Server -P Queue [-R] [-N Filter] [-L] [-p] [-q] [-x] [-# JobNumber] [-u UserName] [-X]  
[-o Option] [-T Timeout] [-C] [-D DebugOutputFile] [File ...]
```

Description

The **rembak** command sends a job to be queued on a remote server. The request can either be a print job, a status request, a job cancel request, or a request to kill the remote queuing system. The server and the queue flags are required. All the other flags are optional, depending on what needs to be done.

This command should only be called by the **qdaemon** command. It is not intended to be entered on the command line by a user. See the **enq** command for details on how to issue a print job request, or use the System Manager Interface Tool (SMIT) to request a print job.

Flags

Item	Description
-# <i>JobNumber</i>	Specifies the <i>JobNumber</i> to cancel.
-C	Sends control file first. The lpd protocol allows two handshaking sequences for processing a print job. The default consists of sending the data file(s) first followed by the control file. The other sequence is to send the control file first followed by the data file(s). If -C is specified, rembak will send the control file first followed by the data file(s).
-D <i>DebugOutputfile</i>	Turns on the debugging option for rembak . If no output file name is specified, or if there are any problems creating or writing to the output file, the debugging option is ignored. If the output file specified already exists, new debugging output is appended to the end of it.
-L	Indicates a long (verbose) status request from the remote queue.

Item	Description
-N <i>Filter</i>	Indicates the machine type of the remote server. The filter name is specified by the s_statfilter attribute in the /etc/qconfig file. Values for the <i>filter</i> variable include the following: /usr/lib/lpd/aixshort Indicates the server is another AIX machine. /usr/lib/lpd/aixv2short Indicates the server is an RT with an AIX Version 2 operating system. /usr/lib/lpd/bsdshort Indicates the server is a bsd machine /usr/lib/lpd/attshort Indicates the server is an AT&T machine
-o <i>Option</i>	Specifies an <i>Option</i> to be sent to the backend on the remote server. (These <i>Options</i> are passed through the rembak command.)
-p	Indicates that the port range used by rembak is restricted to ports less than 1023.
-P <i>Queue</i>	Specifies the name of the <i>Queue</i> on the remote server where the print job is sent.
-q	Indicates a short (abbreviated) status request from the remote queue.
-R	Restarts the remote queuing system. Note: The -R flag is not supported when sending a request to an operating system. The lpd daemon does not support such a request. The -R flag is supported only for compatibility with other systems.
-S <i>Server</i>	Specifies the name of the remote print <i>Server</i> where the print request is sent.
-T <i>Timeout</i>	Sets a timeout period, in minutes, for rembak to wait for acknowledgements from the remote server. If no value is specified, a default timeout of 90 seconds is used. This default is also used if Timeout is 0 or a negative value.
-u <i>UserName@HostName</i>	Cancels a print job for <i>UserName</i> that was submitted from the <i>HostName</i> machine. Note: The queuing system does not support multibyte host names.
-X	Specifies that the rembak command send the -o <i>Option</i> to the remote server, even if the remote server is a non-AIX machine. If the remote is a non-AIX machine, then the <i>Option</i> is sent without the -o flag. Thus, -o abc is sent as -abc. To use the -X flag on a remote queue, the following line for the specific queue must be included in the /etc/qconfig file: <pre>backend = /usr/lib/lpd/rembak -X</pre> The qprt, lpr and other queuing commands are not guaranteed to work when -X is specified on a queue. Use the enq command.
-x	Cancels a job request. Use the -# JobNumber flag or the -u UserName flag to cancel a request.

Examples

1. To print the files `spinach`, `asparagus`, and `broccoli` on the queue `popeye` on the remote server `olive`, which is an RT with an AIX Version 2 operating system, enter:

```
rembak -S olive -P popeye -N /usr/lib/lpd/aixv2short spinach asparagus broccoli
```

2. To issue a verbose status request to `olive` for the queue `popeye`, enter:

```
rembak -S olive -P popeye -N /usr/lib/lpd/aixv2short -L
```

3. To cancel job number 23 on a remote server submitted by user `sweetpea` from machine `bluto`, which is a Version 3 machine, enter:

```
rembak -S olive -P popeye -N /usr/lib/lpd/aixv2short -x -#23 -u sweetpea@bluto
```

Files

Item	Description
<code>/usr/lib/lpd/rembak</code>	Contains the rembak command.
<code>/etc/hosts.lpd</code>	Contains host names that are allowed to do print requests.
<code>/etc/hosts.equiv</code>	Contains host names that are allowed to do print requests.

remove Command

Purpose

Deletes files from **`var/adm/acct/sum`** and **`var/adm/acct/nite`** subdirectories.

Syntax

```
/usr/sbin/acct/remove
```

Description

The **remove** command deletes all **`/var/adm/acct/sum(x)/wtmp*`**, **`/var/adm/acct/sum(x)/pacct*`**, and **`/var/adm/acct/nite(x)/lock*`** files. The **remove** command must be scheduled with the **cron** daemon. Also, the **remove** command should be run at the end of every accounting period, rather than every night.

Security

Access Control: This command should grant execute (x) access only to members of the **adm** group.

Files

Item	Description
<code>/usr/sbin/acct</code>	The path to the accounting commands.
<code>/var/adm/acct/nite</code>	Contains accounting data files.
<code>/var/adm/acct/nitex</code>	Contains accounting data files when user names greater than 8 characters are used.
<code>/var/adm/acct/sum</code>	Cumulative directory for daily accounting records.
<code>/var/adm/acct/sumx</code>	Cumulative directory for daily accounting records when user names greater than 8 character are used.

removevsd Command

Purpose

Removes a set of virtual shared disks.

Syntax

```
removevsd  
    {-v vsd_names | -a} [-f]
```

Description

Use this command to remove the logical volumes associated with the virtual shared disks. Volume groups are not removed with this command.

If the virtual shared disk is configured on any of the nodes on the system partition, this command is unsuccessful, unless the **-f** flag is specified.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit delete_vsd
```

and select the **Remove a Virtual Shared Disk** option.

Flags

-v

Specifies the virtual shared disk name or names that are to be removed by this command.

-a

Specifies that the command should remove all virtual shared disks in the RSCT peer domain.

-f

Forces the system to unconfigure the virtual shared disks and remove them. If **-f** is not specified and any of the virtual shared disks that are to be removed are configured, the command is unsuccessful.

Parameters

vsd_name

Specifies a virtual shared disk. If the virtual shared disk is not in the stopped state, you will get an error message.

Security

You must have `root` authority to run this command.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **startrpdomain** command. To bring a particular node online in an existing peer domain, use the **startrpnode** command. For more information on creating and administering an RSCT peer domain, refer to *RSCT Administration Guide*.

Examples

To unconfigure and remove all defined virtual shared disks in a system or system partition, enter:

```
removevsd -a -f
```

Location

/opt/rsct/vsd/bin/removevsd

rendev Command

Purpose

Renames a device.

Syntax

```
rendev -l Name -n NewName [-u]
```

Description

The **rendev** command enables devices to be renamed. The device to be renamed, is specified with the **-l** flag, and the new wanted name is specified with the **-n** flag.

The new wanted name must not exceed 15 characters in length. Acceptable characters for the new name include alphanumeric characters and underscore (_). If the name is already used or is present in the /dev directory, the operation fails. If the name formed by appending the new name after the character r is already used as a device name, or appears in the /dev directory, the operation fails.

If the device is in the Available state, the **rendev** command must unconfigure the device before renaming it. This operation is similar to the operation performed by the `rmdev -l Name` command. If the unconfigure operation fails, the renaming also fails. If the unconfigure operation succeeds, the **rendev** command will configure the device, after renaming it, to restore it to the Available state. The **-u** flag might be used to prevent the device from being configured again after it is renamed.

Note: Disk drive devices that are members of the root volume group, or that becomes members of the root volume group (by using LVM or install procedures), must not be renamed. Renaming such disk drives might interfere with the ability to recover from certain scenarios, including boot failures.

Some devices might have special requirements on their names for other devices or applications to use them. Using the **rendev** command to rename such a device might result in the device being unusable.

Note: To protect the configuration database, the **rendev** command cannot be interrupted once it is started. Trying to stop this command before completion, might result in a corrupted database.

Flags

Item	Description
-l <i>Name</i>	Specifies the device, indicated by the <i>Name</i> parameter, to be renamed in the customized devices object.
-n <i>NewName</i>	Specifies the new name, indicated by the <i>NewName</i> parameter, to be assigned to the device.
-u	This flag is optional. It indicates that the device is not to be configured after it is renamed.

Examples

1. To rename disk `hdisk5` to `hdisk2`, enter the following command:

```
rendev -l hdisk5 -n hdisk2
```

2. To rename disk `hdisk3` to `ootvg`, enter the following command:

```
rendev -l hdisk3 -n ootvg
```

The second command fails because `ootvg` appended to `r` results in the name `rootvg`, which conflicts with the `rootvg` volume group name.

renice Command

Purpose

Alters the nice value of running processes.

Syntax

renice [**-n** *Increment*] [**-g** | **-p** | **-u**] *ID* ...

Description

The **renice** command alters the nice value of one or more running processes. The *nice value* is the decimal value of the system scheduling priority of a process. By default, the processes affected are specified by their process IDs. When you specify a process group, the request applies to all processes in the process group.

The nice value is determined in an implementation-dependent manner. If the requested increment raises or lowers the nice value of the executed utility beyond implementation-dependent limits, the limit whose value was exceeded is used.

If you do not have root user authority, you can only reset the priority of processes you own and can only increase their priority within the range of 0 to 20, with 20 being the lowest priority. If you have root user authority, you can alter the priority of any process and set the priority to any value in the range -20 to 20. The specified *Increment* changes the priority of a process in the following ways:

Item	Description
1 to 20	Runs the specified processes slower than the base priority.
0	Sets priority of the specified processes to the base scheduling priority.
-20 to -1	Runs the specified processes quicker than the base priority.

The **renice** command maps these values to those actually used by the kernel.

Note:

1. If you do not have root user authority, you cannot increase the nice value of processes (even if you had originally decreased their priorities).
2. You cannot use the **renice** command to change a process to run at a constant priority. To do this, use the **setpriority** system call.

Flags

Item	Description
-g	Interprets all IDs as unsigned decimal integer process group IDs.
-n <i>Increment</i>	Specifies the number to add to the nice value of the process. The value of <i>Increment</i> can only be a decimal integer from -20 to 20. Positive increment values cause a lesser nice value. Negative increment values require appropriate privileges and cause a higher nice value.

Item	Description
-p	Interprets all IDs as unsigned integer process IDs. The -p flag is the default if you specify no other flags.
-u	Interprets all IDs as user name or numerical user IDs.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion
>0	An error occurred.

Examples

1. To alter the system scheduling priority so that process IDs 987 and 32 have low scheduling priorities, enter:

```
renice -n 5 -p 987 32
```

2. To alter the system scheduling priority so that group IDs 324 and 76 have higher scheduling priorities (if the user has the appropriate privileges to do so), enter:

```
renice -n -4 -g 324 76
```

3. To alter the system scheduling priority so that numeric user ID 8 and user sas have low scheduling priorities, enter:

```
renice -n 4 -u 8 sas
```

Files

Item	Description
/usr/sbin/renice	Contains the renice command.
/etc/passwd	Maps user names to user IDs.

reorgvg Command

Purpose

Reorganizes the physical partition allocation for a volume group.

Syntax

```
reorgvg [ -i ] VolumeGroup [ LogicalVolume ... ]
```

Description

The **reorgvg** command reorganizes the placement of allocated physical partitions within the *VolumeGroup*, according to the allocation characteristics of each logical volume. Use the *LogicalVolume* parameter to reorganize specific logical volumes; highest priority is given to the first logical volume name in the *LogicalVolume* parameter list and lowest priority is given to the last logical volume in the parameter

list. The volume group must be varied on and must have free partitions before you can use the **reorgvg** command.

The relocatable flag of each logical volume must be set to **y** with the **chlv -r** command for the reorganization to take effect; otherwise, the logical volume is ignored.

Note:

1. The **reorgvg** command does not reorganize the placement of allocated physical partitions for any striped logical volumes.
2. At least one free physical partition (PP) must exist on the specified volume group for the **reorgvg** command to run successfully. For mirrored logical volumes, one free PP per physical volume (PV) is required in order for the **reorgvg** command to maintain logical volume strictness during execution; otherwise the **reorgvg** command still runs, but moves both copies of a logical partition to the same disk during its execution.
3. To use this command, you must either have root user authority or be a member of the **system** group.
4. If you enter the **reorgvg** command with the volume group name and no other arguments, the entire volume group is reorganized.
5. You cannot use the **reorgvg** command on a snapshot volume group or a volume group that has a snapshot volume group.
6. You cannot use the **reorgvg** command on a volume group that has an active firmware assisted dump logical volume.

You could also use the System Management Interface Tool (SMIT) **smit reorgvg** fast path to run this command.

Flags

Item	Description

-i	Specifies physical volume names read from standard input. Only the partitions on these physical volumes are organized.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To reorganize logical volumes **lv03**, **lv04**, and **lv07** on volume group **vg02**, enter:

```
reorgvg vg02 lv03 lv04 lv07
```

Only the listed logical volumes are reorganized on **vg02**.

2. To reorganize only the partitions located on physical volumes **hdisk4** and **hdisk6** that belong to logical volumes **lv203** and **lv205**, enter:

```
echo "hdisk4 hdisk6" | reorgvg -i vg02 lv203 lv205
```

The partitions located on physical volumes **hdisk4** and **hdisk6** of volume group **vg02**, that belong to logical volumes **lv203** and **lv205**, are reorganized.

Files

Item	Description
<code>/usr/sbin/reorgvg</code>	Directory where the reorgvg command resides.
<code>/tmp</code>	Directory where the temporary files are stored while the command is running.

repl Command

Purpose

Replies to a message.

Syntax

```
repl [ +Folder ] [ -draftfolder +Folder | -nodraftfolder ] [ Message ] [ -draftmessage Message ]  
[ -annotate [ -noinplace | -inplace ] | -noannotate ] [ -cc Names... ] [ -nocc Names... ] [ -query |  
-noquery ] [ -fcc +Folder ] [ -form FormFile ] [ -editor Editor | -noedit ] [ -format | -noformat ]  
[ -filter File ] [ -width Number ] [ -whatnowproc Program | -nowhatnowproc ]
```

Description

The **repl** command starts an interface enabling you to compose a reply to a message. By default, the command drafts a reply to the current message in the current folder. If you do not specify the **-draftfolder** flag, or if the Draft-Folder: entry in the `$HOME/.mh_profile` file is undefined, the **repl** command searches your MH directory for a **draft** file. If you specify a folder, that folder becomes the current folder.

When you enter the **repl** command, the system places the To:, cc:, and In-Reply-To: fields in the draft and prompts you to enter the text of the reply. To exit the editor, press Ctrl-D. After exiting the editor, the **repl** command starts the MH **whatnow** command. You can see a list of available **whatnow** subcommands by pressing the Enter key at the What now? prompt. With these subcommands, you can re-edit, list, and send a reply, or end the processing of the **repl** command.

Note: You must leave a line of dashes or a blank line between the header and the body of the message for the message to be identified when it is sent.

The **repl** command uses the definitions in the `/etc/mh/replcomps` file to format the reply message. You can create a **replcomps** file in your MH directory or use the **-form** flag to define an alternate reply format. To leave a copy of the original message in the reply message, use the **-filter** flag.

To annotate the original message with redistribution information, use the **-annotate** flag. This flag annotates the original message with the Resent: field and the current date and time. A message is annotated only if you send the reply before you exit **repl** command processing.

Flags

Item	Description
-annotate	Annotates the message being replied to with the time and date of the reply. You can use the -inplace flag to preserve links to an annotated message.
-cc Names	Specifies the users who will be listed in the cc: field of the reply. You can specify the following variables for <i>Names</i> : all , to , cc , and me . The default is -cc all .
-draftfolder +Folder	Places the draft message in the specified folder. If <i>+Folder</i> is not specified, then Current-Folder is assumed.

Item	Description
-draftmessage <i>Message</i>	Specifies the draft message. If you specify -draftfolder without the -draftmessage flag, the default message is new. If you specify this flag without the -draftfolder flag, the system creates the draft in the default file, <i>UserMHdirectory/draft</i> .
-editor <i>Editor</i>	Identifies the initial editor for composing the reply. If you do not specify the -editor flag, the comp command selects the default editor specified by the Editor: entry in your \$HOME/.mh_profile file.
-fcc <i>+Folder</i>	Places a file copy of the reply in the specified folder. If you do not specify this flag, the repl command will not produce a file copy.
-filter <i>File</i>	Reformats the message being replied to and places the reformatted message in the body of the reply. You must specify a <i>File</i> variable with this flag. The -filter flag uses the format file acceptable to the mhl command.
+Folder	Identifies the folder that contains the message to reply to. If a folder is not specified, then Current-Folder is used.
-form <i>FormFile</i>	Specifies a reply format. The repl command treats each line in the specified format file as a format string.
-format	Removes duplicate addresses from the To: , cc: , and Bcc: fields and standardizes these fields using the columns specified by the -width flag. The -format flag indicates if Internet style is to be used, which serves as the default.
-help	Lists the command syntax, available switches (toggles), and version information.
-inplace	Forces annotation to be done in place in order to preserve links to the annotated message.
<i>Message</i>	Specifies a message. If you specify both a message to reply to and a message draft, you must use the -draftmessage flag. Use the following to define a message: Number Number of the message. cur or . (period) Current message. The default reply message. first First message in a folder. last Last message in a folder. new New message that is created. The default draft message is new. next Message following the current message. prev Message preceding the current message.
-noannotate	Prevents annotation. This flag is the default.
-nocc <i>Names</i>	Allows you to specify the users who will not be listed in the cc: field of the reply. You can specify the following for <i>Names</i> : all , to , cc , and me .

Item	Description
-nodraftfolder	Places the draft in the file <i>UserMhDirectory/draft</i> .
-noedit	Suppresses the initial edit.
-noformat	Suppresses both removal of duplicate addresses from the To : , cc : , and Bcc : fields, and standardization of these fields.
-noinplace	Prevents annotation in place. This flag is the default.
-noquery	Automatically builds the To : and cc : fields. This flag is the default.
-nowhatnowproc	Prevents interactive processing for the repl command. This flag prevents editing.
-query	Queries you for permission to include each address in the To : and cc : fields.
-whatnowproc <i>Program</i>	Starts the specified command string as the program to guide you through the reply tasks. The default is the whatnow program.
-width <i>Number</i>	Sets the width of the address fields. The default is 72 columns.

Profile Entries

The following entries are entered in the *UserMhDirectory/.mh_profile* file:

Item	Description
Alternate-Mailboxes:	Specifies the mailboxes.
Current-Folder:	Sets the default current folder.
Draft-Folder:	Sets the default folder for drafts.
Editor:	Sets the default editor.
fileproc:	Specifies the program used to refile messages.
mh1proc:	Specifies the program used to filter the message for which you are creating a reply.
Msg-Protect:	Sets the protection level for the new message files.
Path:	Specifies the user's MH directory.
whatnowproc:	Specifies the program used to prompt What now? questions.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To reply to the current message in the current folder, enter:

```
repl
```

The system responds with text similar to the following:

```
To: patrick@venus
cc: tom@thomas
Subject: Re: Meeting on Monday
In-reply-to: (Your message of Thu, 21 Jul 88 13:39:34 CST.)
```



```
<8807211839.AA01868>
```

You can now enter your reply. When you finish entering the body of your reply, press the Ctrl-D key sequence to exit the editor. The system responds with the following:

```
What now?
```

Enter send to send the reply. If you want to see a list of subcommands, press the Enter key. In this example, you are sending a reply to the current message in the current folder.

2. To send a reply to message 4 in the inbox folder, enter:

```
repl +inbox 4
```

The system responds with a message similar to the following:

```
To: dawn@chaucer
cc: jay@venus
Subject: Re: Status Report
In-reply-to: (Your message of Thu, 21 Jul 88 13:39:34 CST.)
             <8807211839.AA01868>
-----
```

You can now enter your reply. When you finish entering the body of your reply, press the Ctrl-D key sequence to exit the editor. The system responds with the following:

```
What now?
```

Enter send to send the reply. If you want to see a list of subcommands, press the Enter key.

3. To keep track of your reply to the current message in the current folder, use the **-annotate** flag to place a copy of the date and time in the message you are replying to, as follows:

```
repl -annotate
```

The system responds with a message similar to the following:

```
To: patrick@venus
cc: tom@thomas
Subject: Re: Meeting on Friday
In-reply-to: (Your message of Mon, 17 Apr 89 13:39:34 CST.)
             <8904171839.AA01868>
-----
```

You can now enter your reply. When you finish entering the body of your reply, press the Ctrl-D key sequence to exit the editor. The system responds with the following:

```
What now?
```

Enter send to send the reply. If you quit the editor without sending the reply, the annotation does not occur.

Files

Item	Description
\$HOME/.mh_profile	Specifies the user's MH profile.
/etc/mh/replcomps	Contains the MH default reply template.
<i>UserMhDirectory/replcomps</i>	Contains the user's default reply form.
/usr/bin/repl	Contains the repl command.
<i>UserMhDirectory/draft</i>	Contains the current message draft.

replacepv Command

Purpose

Replaces a physical volume in a volume group with another physical volume.

Syntax

replacepv [**-f**] {*SourcePhysicalVolume* | *SourcePhysicalVolumeID* } *DestinationPhysicalVolume*

replacepv [**-R**] *dir_name* [*DestinationPhysicalVolume*]

Description

The **replacepv** command replaces allocated physical partitions and the data they contain from the *SourcePhysicalVolume* to *DestinationPhysicalVolume*. The specified source physical volume cannot be the same as *DestinationPhysicalVolume*.

Note:

1. The *DestinationPhysicalVolume* must not belong to a volume group.
2. The *DestinationPhysicalVolume* size must be at least the size of the *SourcePhysicalVolume*.
3. The **replacepv** command cannot replace a *SourcePhysicalVolume* with stale logical volume unless this logical volume has a non-stale mirror.
4. You cannot use the **replacepv** command on a snapshot volume group or a volume group that has a snapshot volume group.
5. Running this command on a physical volume that has an active firmware assisted dump logical volume temporarily changes the dump device to **/dev/sysdumpnull**. After the migration of logical volume is successful, this command calls the **sysdumpdev -P** command to set the firmware assisted dump logical volume to the original logical volume.
6. The VG corresponding to the *SourcePhysicalVolume* is examined to determine if a PV type restriction exists. If a restriction exists, the *DestinationPhysicalVolume* is examined to ensure that it meets the restriction. If it does not meet the PV type restriction, the command will fail.

The allocation of the new physical partitions follows the policies defined for the logical volumes that contain the physical partitions being replaced.

Flags

Item	Description
-f	Forces to replace a <i>SourcePhysicalVolume</i> with the specified <i>DestinationPhysicalVolume</i> unless the <i>DestinationPhysicalVolume</i> is part of another volume group in the Device Configuration Database or a volume group that is active.
-R <i>dir_name</i>	Recovers replacepv if it is interrupted by <ctrl-c>, a system crash, or a loss of quorum. When using the -R flag, you must specify the directory name given during the initial run of replacepv . This flag also allows you to change the <i>DestinationPhysicalVolume</i> .

Security

Access Control: You must have root authority to run this command.

Examples

1. To replace physical partitions from hdisk1 to hdisk6, enter:

```
replacepv hdisk1 hdisk6
```

Files

Item	Description
<code>/usr/sbin</code>	Directory where the replacepv command resides.
<code>/tmp</code>	Directory where the temporary files are stored while the command is running.

repquota Command

Purpose

Summarizes quotas for a file system.

Syntax

```
repquota [ -v ] [ -c ] [ -g ] [ -u ] [ -l ] { -a | FileSystem ... }
```

Description

The **repquota** command prints a summary of quotas and disk usage for a file system specified by the *FileSystem* parameter. If the **-a** flag is specified instead of a file system, the **repquota** command prints the summary for all file systems enabled with quotas in the **/etc/filesystems** file. By default, both user and group quotas are printed.

For each user or group, the **repquota** command prints:

- Number of existing user or group files
- Amount of disk space being used by the user or group
- User or group quotas

Flags

Item	Description
-a	Specifies that quotas are printed for all file systems enabled with quotas in the /etc/filesystems file.
-c	Changes the output of the command to a colon-delineated format.
-g	Specifies that only group quotas are printed.
-l	Enables long user names to be printed on the repquota report. The default behavior of the report will be to truncate the name at 9 characters. If the -l option is specified, the full user name will be used.
-u	Specifies that only user quotas are printed.
-v	Prints a header line before the summary of quotas for each file system.

Security

Access Control: Only the root user can execute this command.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

To print a summary of user quotas in the /u file system, enter:

```
repquota -u /u
```

The system prints the following information:

User		used	Block soft	limits hard	grace	used	File soft	limits hard	grace
root	--	3920	0	0		734	0	0	
dave	+	28	8	30	3 days	3	0	0	
keith	--	48	0	0		7	0	0	

The + printed in the first column next to dave indicates that the user has exceeded established block limits. If there were a + in the second column, it would indicate that the user had exceeded established file limits.

Files

Item	Description
quota.user	Specifies user quotas.
quota.group	Specifies group quotas.
/etc/filesystems	Contains file system names and locations.
/etc/group	Contains basic group attributes.
/etc/passwd	Contains user names and locations.

reset Command

Purpose

Initializes terminals.

Syntax

```
reset [ -e C ] [ -k C ] [ -i C ] [ - ] [ -s ] [ -n ] [ -I ] [ -Q ] [ -m [ Identifier ] [ TestBaudRate ] :Type ] ... [ Type ]
```

Description

The **reset** command is a link to the **tset** command. If the **tset** command is run as the **reset** command, it performs the following actions before any terminal-dependent processing is done:

- Set Cooked and Echo modes to on
- Turn off cbreak and Raw modes
- Turn on new-line translation
- Restore special characters to a sensible state.

Any special character that is found to be NULL or -1 is reset to its default value. All flags to the **tset** command can be used with the **reset** command.

The **reset** command is most useful when a program dies and leaves a terminal in an undesirable state. The sequence <LF>reset<LF> (where <LF> is Ctrl-J, the line feed) may be required to get the **reset** command to run successfully since carriage-return might not work in this state. The <LF>reset<LF> sequence frequently will not be echoed.

Flags

Item	Description
-	The name of the terminal decided upon is output to standard output. This is intended to be captured by the shell and placed in the TERM environment variable.
-e C	Set the erase character to the character specified by the C variable on all terminals. The default is the backspace character on the terminal, usually ^ (cedilla). The character C can either be typed directly or entered using the ^ (cedilla).
-I	Suppresses transmission of terminal initialization strings.
-i C	Is similar to the -e flag, but uses the interrupt character rather than the erase character. The C variable defaults to ^C. The ^ character can also be used for this option.
-k C	Is similar to the -e flag, except uses the line-kill character rather than the erase character. The C variable defaults to ^X. The kill character is not changed if -k is not specified. The ^ character can also be used for this option.
-m IdentifierTestbaudRate:Type	Specifies which terminal type (in the <i>Type</i> parameter) is usually used on the port identified in the <i>Identifier</i> parameter. A missing identifier matches all identifiers. You can optionally specify the baud rate in the <i>TestBaudRate</i> parameter.
-n	On systems with the Berkeley 4.3 tty driver, specifies that the new tty driver modes should be initialized for this terminal. For a CRT, the CRTERASE and CRTKILL modes are set only if the baud rate is 1200 bps or greater. See the tty file for more information.
-Q	Suppresses printing of the Erase set to and Kill set to messages.
-s	Prints the sequence of cs h commands that initialize the TERM environment variable, based on the name of the terminal decided upon.

Files

Item	Description
/usr/share/lib/terminfo/?/*	Contains the terminal capability database.

resetsrc Command

Purpose

Resets a resource that is, forces the resource to move to the offline state.

Syntax

To reset one or more resources, using data entered on the command line:

```
resetsrc -s "selection_string" [-N { node_file | " - " } ] [-h] [-TV] resource_class [arg=value...]
```

```
resetsrc -r [-h] [-TV] resource_handle [arg=value...]
```

To reset one or more resources using command arguments that are predefined in an input file:

```
resetrrsrc -f resource_data_input_file -s "selection_string" [ -N { node_file | "-" } ] [-h] [-TV]
resource_class
```

```
resetrrsrc -f resource_data_input_file -r [-h] [-TV] resource_handle
```

To display the names and data types of the command arguments:

```
resetrrsrc -l [-h] resource_class
```

Description

The `resetrrsrc` command requests that the resource monitoring and control (RMC) subsystem force one or more resources offline. The request is performed by the appropriate resource manager.

To reset one or more resources, use the `-s` flag to force offline all of the resources that match the specified selection string. To reset one specific resource, use the `-r` flag to specify the resource handle that represents that specific resource.

Instead of specifying multiple node names in *selection_string*, you can use the `-N node_file` flag to indicate that the node names are in a file. Use `-N "-"` to read the node names from standard input.

Use the `-l` flag to determine whether the specified resource class accepts any additional command arguments.

The successful completion of this command does not guarantee that the resource is offline, only that the resource manager successfully received the request to force this resource offline. Monitor the resource dynamic attribute `OpState` to determine when the resource is forced offline. Register an event for the resource, specifying the `OpState` attribute, to know when the resource is offline. Or, intermittently run the `lsrrsrc` command until you see that the resource is offline (the value of `OpState` is 2). For example:

```
lsrrsrc -s 'Name == "/filesystem1"' -t IBM.FileSystem Name OpState
```

If Cluster Systems Management (CSM) is installed on your system, you can use CSM defined node groups as node name values to refer to more than one node. For information about working with CSM node groups and using the `CSM nodegrp` command, see the *CSM: Administration Guide* and the *CSM: Command and Technical Reference*.

Parameters

resource_class

Specifies the name of the resource class that contains the resources that you want to force offline.

resource_handle

Specifies the resource handle that corresponds to the resource you want to force offline. Use the `lsrrsrc` command to obtain a list of valid resource handles. The resource handle must be enclosed within double quotation marks, for example:

```
"0x4017 0x0001 0x00000000 0x0069684c 0x0d4715b0 0xe9635f69"
```

arg=value...

Specifies one or more pairs of command argument names and values.

arg

Specifies the argument name.

value

Specifies the value for this argument. The value data type must match the definition of the argument data type.

Command arguments are optional. If any *arg=value* pairs are entered, there must be one *arg=value* pair for each command argument defined for the offline function for the specified resource class.

Use `resetrrsrc -l` to get a list of the command argument names and data types for the specific resource class.

Flags

-f *resource_data_input_file*

Specifies the name of the file that contains resource argument information. The following contents of the file is displayed:

```
PersistentResourceArguments::  
argument1 = value1  
argument2 = value2
```

-l

Lists the command arguments and data types. Some resource managers accept additional arguments that are passed to the offline request. Use this flag to list any defined command arguments and the data types of the command argument values.

-N { *node_file* | "-" }

Specifies that node names are read from a file or from standard input. Use `-N node_file` to indicate that the node names are in a file.

- There is one node name per line in *node_file*
- A number sign (#) in column 1 indicates that the line is a comment
- Any blank characters before a node name are ignored
- Any characters after a node name are ignored

Use `-N "-"` to read the node names from standard input.

The `CT_MANAGEMENT_SCOPE` environment variable determines the scope of the cluster. If `CT_MANAGEMENT_SCOPE` is not set, management domain scope is chosen first (if a management domain exists), peer domain scope is chosen next (if a peer domain exists), and then local scope is chosen, until the scope is valid for the command. The command runs once for the first valid scope it finds. For example, if a management domain and a peer domain both exist and `CT_MANAGEMENT_SCOPE` is not set, this command applies to the management domain. If you want this command to apply to the peer domain, set `CT_MANAGEMENT_SCOPE` to 2.

-r

Forces offline the specific resource that matches the specified resource handle.

-s "*selection_string*"

Specifies the selection string. All selection strings must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'  
-s 'Name != "test"'
```

Only persistent attributes can be listed in a selection string.

-h

Writes the command usage statement to standard output.

-T

Writes the command trace messages to standard error. For your software service organization use only.

-V

Writes the command verbose messages (if there are any available) to standard output.

Environment variables

CT_CONTACT

When the `CT_CONTACT` environment variable is set to a host name or IP address, the command contacts the resource monitoring and control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command

is being run. The resource class or resources that are displayed or modified by the command are on the system to which the connection is established.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. CT_IP_AUTHENT has meaning only if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon to monitor and control the resources and resource classes. The management scope determines the set of possible target nodes where the resources and resource classes can be monitored and controlled. The valid values are:

- 0** Specifies *local* scope.
- 1** Specifies *local* scope.
- 2** Specifies *peer domain* scope.
- 3** Specifies *management domain* scope.

If this environment variable is *not* set, *local* scope is used.

Standard output

When the -h flag is specified, this command usage statement is written to standard output. When the -V flag is specified, this command verbose messages (if there are any available) are written to standard output.

Standard error

All trace messages are written to standard error.

Exit status

- 0** The command ran successfully.
- 1** An error occurred with RMC.
- 2** An error occurred with the command-line interface (CLI) script.
- 3** An incorrect flag was specified on the command line.
- 4** An incorrect parameter was specified on the command line.
- 5** An error occurred with RMC that was based on incorrect command-line input.
- 6** No resources were found that match the specified selection string.

Security

You need write permission for the *resource_class* specified in `resetrsrc` to run `resetrsrc`. Permissions are specified in the access control list (ACL) file on the contacted system. See the *Administering RSCT* guide for information about the ACL file and how to modify it.

Implementation specifics

This command is part of the `rsct.core.rmc` fileset for AIX and `rsct.core-3.1.0.0-0.platform.rpm` package for Linux, Solaris, and Windows, where *platform* is `i386`, `ppc`, `ppc64`, `s390`, or `x86_64`.

Location

`/opt/rsct/bin/resetrsrc`

Examples

Suppose that you have a peer domain called `foo` with three defined nodes: `nodeA`, `nodeB`, and `nodeC`. `nodeA` has two Ethernet cards: `ent0` and `ent1`.

1. Suppose `nodeA` is online and `ent0` (on `nodeA`) is also online. To force `ent0` offline on `nodeA`, run this command on `nodeA`:

```
resetrsrc -s 'Name == "ent0"' IBM.EthernetDevice
```

2. Suppose `nodeA` and `nodeB` are online, `ent0` (on `nodeA`) is also online, and you are currently logged on to `nodeB`. To force `ent0` offline on `nodeA`, run this command on `nodeB`:

```
resetrsrc -s 'NodeName == "nodeA" AND Name == "ent0"' IBM.EthernetDevice
```

3. Suppose `nodeA` and `nodeB` are online and file system `/filesys1` is defined and mounted on `nodeB`. To force `/filesys1` offline on `nodeB`, run this command on `nodeA`:

```
resetrsrc -s 'NodeName == "nodeB" AND Name == "/filesys1"' IBM.FileSystem
```

4. Suppose the resource handle for `ent0` on `nodeA` is:

```
0x406b 0x0001 0x00000000 0x0069564c 0x0dc1f272 0xb9de145e
```

To force `ent0` offline on `nodeA`, run this command on `nodeA`:

```
resetrsrc -r "0x406b 0x0001 0x00000000 0x0069564c 0x0dc1f272 0xb9de145e"
```

5. To reset `ent0` on `nodeA` and `nodeB`, using the `/tmp/common/node_file` file:

```
# common node file
#
nodeA
nodeB
#
```

as input, enter:

```
resetrsrc -s 'Name == "ent0"' -N /tmp/common/node_file \
IBM.EthernetDevice
```

resize Command

Purpose

Sets the **TERMCAP** environment variable and terminal settings to the current window size.

Syntax

resize [**-c** | **-u**] [**-s** [*Rows Columns*]]

Description

The **resize** command utility prints a shell command for setting the **TERM** and **TERMCAP** environment variables to indicate the current size of the xterm window from which the command is run. For this output to take effect, the **resize** command must either be evaluated as part of the command line (usually done with a shell alias or function) or else be redirected to a file that can then be read in. From the C shell (usually known as **/bin/csh**), the following alias could be defined in the user's **.cshrc** file:

```
% alias rs 'set noglob; `eval resize`'
```

After resizing the window, the user would enter:

```
% rs
```

Users of versions of the Bourne shell (usually known as **/bin/sh**) that do not have command functions will need to send the output to a temporary file and then read it back in with the **.** (dot) command:

```
$ resize >/tmp/out
$ . /tmp/out
```

Flags

Item	Description
-c	Indicates that C shell commands should be generated even if the user's current shell is not /bin/csh .
-u	Indicates that Bourne shell commands should be generated even if the user's current shell is not a Bourne shell.
-s [<i>Rows Columns</i>]	Indicates that Sun console escape sequences will be used instead of the special xterm escape code. If the <i>Rows</i> and <i>Columns</i> parameters are given, the resize command will ask the xterm window to resize itself. However, the window manager may choose to disallow the change.

Note: The **-c** or **-u** must appear before **-s** if both are specified.

File

Item	Description
/etc/termcap	Provides modification for the base termcap entry.

resource_data_input Information File

Purpose

Describes how to use an input file for passing resource class information, such as resource attribute names and values, to the resource monitoring and control (RMC) command-line interface (CLI).

Description

You can use the **-f** flag with most RMC commands to specify the name of a resource data input file when you want to pass resource persistent attribute values and other information to the RMC CLI. This is useful when typing information about the command line would be too cumbersome or prone to typographical

errors. The data in this file is used for defining resources or for changing the persistent attribute values of a resource or resource class. The resource data input file, which must be in POSIX format, has no set location. It can be a temporary file or a permanent file, depending on your requirements.

The **chrsrc**, **mkrsrc**, **resetsrc**, **rmrsrc**, **runact**, **startsrc**, and **stopsrc** commands read this file when they are issued with the **-f** flag. The **lsactdef**, **lsrsrc**, and **lsrsrcdef** commands generate a file with this format when they are issued with the **-i** flag.

Keywords are used in the input file to indicate which type of data is listed in the related stanza:

ResourceAction

Resource action element names and values for the resource action when starting an action. The **runact** command reads in the resource action elements. These elements are ignored if the input file is read by **runact -c**.

ResourceClassAction

Resource class action element names and values for the resource class action when starting a class action. The **runact** command reads in the resource action elements.

PersistentResourceArguments

Resource command argument names and values for those commands that accept them: **mkrsrc**, **resetsrc**, **rmrsrc**, **startsrc**, and **stopsrc**. Command arguments are optional and are defined by the resource class. Specify the **-l** option with these commands to see the command arguments for a resource class.

PersistentResourceAttributes

Persistent attribute names and values for one or more resources for a specific resource class used to define a new resource or change attribute values for an existing resource. The persistent resource attributes are read in by the commands **mkrsrc** and **chrsrc**. These attributes are ignored if the input file is read by the **chrsrc** command that is specified with the **-c** flag.

PersistentResourceClassAttributes

Persistent attribute names and values for a resource class used to change the attribute values of an existing resource class. The persistent resource class attributes are read in by the **chrsrc** command only when the **-c** flag is specified.

In general, a *resource_data_input* file is a flat text file with the following format. **Bold** words are literal. Text that precedes a single colon (:) is an arbitrary label and can be any alphanumeric text.

```
PersistentResourceAttributes::
# This is a comment
  label:
    AttrName1 = value
    AttrName2 = value
    AttrName3 = value
  another label:
    Name      = name
    NodeNumber = 1
  :
  ::

PersistentResourceClassAttributes::
# This is a comment
  label:
    SomeSettableAttrName      = value
    SomeOtherSettableAttrName = value
  :
  ::

PersistentResourceArguments::
# This is a comment
  label:
    ArgName1 = value
    ArgName2 = value
    ArgName3 = value
  :
  ::
```

See the Examples section for more details.

Some notes about formatting follow:

- The keywords `PersistentResourceAttributes`, `PersistentResourceClassAttributes`, and `PersistentResourceArguments` are followed by two colons (::).
- The order of the keyword stanzas is not significant in the file. For example, `PersistentResourceClassAttributes` can precede `PersistentResourceClass`. It does not affect the portion of the data that is read in by the calling CLI.
- Individual stanza headings (beneath the keywords) are followed by one colon (:), for example: `c175n05 resource info:`.
- White space at the beginning of lines is not significant. Tabs or spaces are suggested for readability.
- Any line with a pound sign (#) as the first printable character is a comment.
- Each entry on an individual line is separated by white space (spaces or tabs).
- Blank lines in the file are not significant and are suggested for readability.
- There is no limit to the number of resource attribute stanzas included in a particular `PersistentResourceAttributes` section.
- There is no limit to the number of resource class attribute stanzas included in a particular `PersistentResourceClassAttributes` section. Typically, there is only one instance of a resource class. In this case, only one stanza is expected.
- If only one resource attribute stanza is included in a particular `PersistentResourceAttributes` section, the *label:* line can be omitted. This also applies to the `ResourceAction` section.
- If only one resource class attribute stanza is included in a particular `PersistentResourceClassAttributes` section, the *label:* line can be omitted. This also applies to the `ResourceClassAction` section.
- Values that contain spaces must be enclosed in quotation marks.
- A double colon (::) indicates the end of a section. If a terminating double colon is not found, the next Reserved Keyword or end of file signals the end of a section.
- Double quotation marks included within a string that is surrounded by double quotation marks must be escaped. (\").

Note: Double quotation marks can be nested within single quotation marks.

Examples:

- `"Name == \"testing\""`
- `'Name == "testing"'`

This syntax is preferred if your string is a selection string and you are going to cut and paste to the command line.

- Single quotation marks included within a string that is surrounded by single quotation marks must be escaped. (\').

Note: Single quotation marks can be nested within double quotation marks.

Here are some examples:

- `'Isn\'t that true'`
- `"Isn't that true"`

This syntax is preferred if you are going to cut and paste to the command line.

- The format you use to enter data in a *resource_data_input* file might not be the same format used on the command line. The shell you choose to run the commands in has its own rules regarding quotation marks. Refer to the documentation for your shell for these rules, which determine how to enter data on the command line.

Implementation specifics

This information is part of the `rsct.core.rmc` fileset for AIX and `rsct.core-3.1.0.0-0.platform.rpm` package for Linux, Solaris, and Windows, where *platform* is *i386*, *ppc*, *ppc64*, *s390*, or *x86_64*.

Location

/opt/rsct/man/resource_data_input.7

Examples

1. This sample **mkrsrc** command:

```
mkrsrc -f /tmp/my_resource_data_input_file IBM.Example
```

uses the sample input file /tmp/my_resource_data_input_file for the IBM.Example resource class. The contents of the input file look like this:

```
PersistentResourceAttributes::
# Resource 1 - only set required attributes
resource 1:
    Name="c175n04"
    NodeList = {1}
# Resource 2 - setting both required and optional attributes
# mkrsrc -e2 IBM.Example displays required and optional
# persistent attributes
resource 2:
    Name="c175n05"
    NodeList = {1}
    Int32 = -99
    UInt32 = 99
    Int64 = -123456789123456789
    UInt64 = 123456789123456789
    Float32 = -9.89
    Float64 = 123456789.123456789
    String = "testing 123"
    Binary = 0xaabbccddeeff
    RH = "0x0000 0x0000 0x00000000 0x00000000 0x00000000 0x00000000"
    SD = [hello,1,{2,4,6,8}]
    Int32Array = {-4, -3, -2, -1, 0, 1, 2, 3, 4}
    Int64Array = {-4,-3,-2,-1,0,1,2,3,4}
    UInt32Array = {0,1,2,3,4,5,6}
    UInt64Array = {0,1,2,3,4,5,6}
    Float32Array = {-3.3, -2.2, -1.2, 0, 1, 2.2, 3.3}
    Float64Array = {-3.3, -2.2, -1.2, 0, 1, 2.2, 3.3}
    StringArray = {abc,"do re mi", 123}
    BinaryArray = {"0x01", "0x02", "0x0304"}
    RHArray = {"0x0000 0x0000 0x00000000 0x00000000 0x00000000 0x00000000",
               "0xaaaa 0xaaaa 0xbbbbbbbb 0xcccccccc 0xdddddddd 0xeeeeeeee"}
    SDArray = {[hello,1,{0,1,2,3}],[hello2,2,{2,4,6,8}]}
```

2. This sample **chrsrc** command:

```
chrsrc -f /tmp/Example/ch_resources -s 'Name == "c175n05"' IBM.Example
```

uses the sample input file /tmp/Example/ch_resources to change the attribute values of existing IBM.Example resources. The contents of the input file look like this:

```
PersistentResourceAttributes::
# Changing resources that match the selection string entered
# when running chrsrc command.
resource 1:
    String = "this is a string test"
    Int32Array = {10,-20,30,-40,50,-60}
```

3. This sample **rmrsrc** command:

```
rmrsrc -l IBM.Examplebar
```

shows the optional command arguments:

```
rmrsrc IBM.Examplebar ExampleInt32=int32 ExampleUInt32=uint32
```

4. This sample **rmrsrc** command:

```
rmrsrc -f /tmp/Examplebar/rm_resources -s 'Name == "c175n05"' IBM.Examplebar
```

uses the sample input `/tmp/Examplebar/rm_resources` file to specify the optional command arguments for **rmrsrc** command. The contents of the input file look like this:

```
PersistentResourceArguments::  
# Specifying command arguments when running rmrsrc command.  
resource 1:  
    ExampleInt32      =    1  
    ExampleUInt32     =    0
```

restart-secdapclntd Command

Purpose

The **restart-secdapclntd** script is used to stop the currently running **secdapclntd** daemon process and then restart it.

Syntax

```
/usr/sbin/restart-secdapclntd [ -C CacheSize ] [ -p NumOfThread ] [ -t CacheTimeOut ] [ -T HeartBeatIntv ] [ -o ldapTimeOut ]
```

Description

The **restart-secdapclntd** script stops the **secdapclntd** daemon if it is running, and then restarts it. If the **secdapclntd** daemon is not running, it simply starts it.

Flags

By default, the **secdapclntd** daemon reads the configuration information specified in the **/etc/security/ldap/ldap.cfg** file at startup. If the following options are given in command line when starting **secdapclntd** process, the options from the command line will overwrite the values in the **/etc/security/ldap/ldap.cfg** file.

Item	Description
-C <i>CacheSize</i>	Sets the maximum cache entries used by the secdapclntd daemon to CacheSize number of entries. Valid range is 100-10,000 entries for user cache. The default is 1000. The group cache entries will be 10% of the user cache entries.
-o <i>ldapTimeOut</i>	Timeout period in seconds for LDAP client requests to the server. This value determines how long the client will wait for a response from the LDAP server. Valid range is 0 - 3600 (1 hour). Default is 60 seconds. Set this value to 0 to disable the timeout and force the client to wait indefinitely.
-p <i>NumOfThread</i>	Sets the number of thread used by the secdapclntd daemon to NumOfThread threads. Valid range is 1-1000. The default is 10.
-t <i>CacheTimeOut</i>	Sets the cache to expire in CacheTimeOut seconds. Valid range is 60- 3600 seconds. The default is 300 seconds.
-T <i>HeartBeatIntv</i>	Sets the time interval of heartbeat between this client and the LDAP server. Valid values are 60-3,600 seconds. Default is 300.

Security

A user with the **aix.security.ldap** authorization is authorized to use this command.

Examples

1. To restart the **secdapclntd** daemon, type:

```
/usr/sbin/restart-secdapclntd
```

2. To restart the **secdapclntd** with using 30 threads and cache timeout value of 500 seconds, type:

```
/usr/sbin/restart-secdapclntd -p 30 -t 500
```

Files

Item	Description
/etc/security/ldap/ldap.cfg	Contains information needed by the secdapclntd daemon to connect to the server.

restbase Command

Purpose

Reads the base-customized information from the boot image, and restores it into the Device Configuration database used during system boot phase 1.

Syntax

```
restbase [ -o File ] [ -d Path ] [ -v ]
```

Description

The **restbase** command reads the base-customized information from the boot disk and puts it in the specified Device Configuration database directory. By default, the base information is read from the boot disk. If no Device Configuration database directory is specified, then the **restbase** command restores this information into the **/etc/objrepos** directory. You can use the **-o** flag to specify a file, other than the boot disk, from which to read the base-customized information.



Attention: The **restbase** command is intended to be executed only during phase 1 of system boot. Do not execute it in a run-time environment or you could destroy the Device Configuration database.

Flags

Item	Description
-o <i>File</i>	Specifies a file that contains base-customized data.
-d <i>Path</i>	Specifies a directory containing the base Device Configuration database.
-v	Causes verbose output to be written to standard output.

Examples

1. To restore base-customized information and see verbose output, enter:

```
restbase -v
```

2. To restore base information into an alternate device database, enter:

```
restbase -d /tmp/objrepos
```

Files

Item	Description
/usr/lib/objrepos/PdDv	Contains entries for all known device types supported by the system.
/etc/objrepos/CuDv	Contains entries for all device instances defined in the system.
/etc/objrepos/CuAt	Contains customized device-specific attribute information.
/etc/objrepos/CuDep	Describes device instances that depend on other device instances.
/etc/objrepos/CuDvDr	Stores information about critical resources that need concurrency management through the use of the Device Configuration Library routines.

restore Command

Purpose

Extracts files from archives that are created with the **backup** command.

Syntax

To restore files archived by file name

>|

```
restore -f Device -x [ d M n Q v q e ] [ -b Number ] [ -s SeekBackup ] [ -E { force | ignore | warn } ] [ File ... ]
```

|<

To list files archived by file name

>|

```
restore -f Device -T | -t [ a l n q v Q ] [ -b Number ] [ -s SeekBackup ]
```

|<

To restore files archived by file system

>|

```
restore -f Device -r [ B n q v y ] [ -b Number ] [ -s SeekBackup ]
```

|<

To restore files archived by file system

>|

```
restore -f Device -R [ B n v y ] [ -b Number ] [ -s SeekBackup ]
```

|<

To restore files archived by file system

>|

```
restore -f Device -i [ h m n q v y ] [ -b Number ] [ -s SeekBackup ]
```

|<

To restore files archived by file system

>|


```
restore -f Device -x [ B h n q v y ] [ -b Number ] [ -s SeekBackup ] [ File ... ]
```

<

To restore files beginning at a specified volume number

>

```
restore -f Device -X Number [ -M d n q v e Q ] [ -b Number ] [ -s Number ] [ -E { force | ignore | warn } ] [ File ... ]
```

<

To list files archived by file system

>

```
restore -f Device -t | -T [ B a l n h q v y ] [ -b Number ] [ -s SeekBackup ] [ File ... ]
```

<

To restore file attributes archived by file name

>

```
restore -f Device -Pstring [ B d q v Q ] [ -b Number ] [ -s SeekNumber ] [ File ... ]
```

<

To restore file attributes archived by file system

>

```
restore -f Device -Pstring [ h q v ] [ -b Number ] [ -s SeekNumber ] [ File ... ]
```

<

Description

The **restore** command reads archives created by the **backup** command and extracts the files that are stored on them. These archives can be in either file name or file system format. An archive can be stored on disk, diskette, or tape. Files must be restored by using the same method that was used to archive the files. This operation requires that you know the format of the archive. The archive format can be determined by examining the archive volume header information that is displayed when you use the **-T** flag. When the **-x**, **-i**, **-T**, or **-t** flags are used, the **restore** command automatically determines the archive format.

Individual files can be restored from either file name or file system archives by using the **-x** flag and specifying the file name. The file name must be specified as it exists on the archive. Files can be restored interactively from file system archives by using the **-i** flag. The names of the files on an archive can be written to standard output by using the **-T** flag.

Users must have write access to the file system device or have Restore authorization to extract the contents of the archive.

To restore from standard input, specify a **-** (dash) with the **-f** flag. You can also specify a range of devices, such as **/dev/rfd0**.

Notes:

1. If you are restoring from a multiple-volume archive, the **restore** command reads the volume that mounted, prompts you for the next volume, and waits for your response. After the next volume is inserted, press the Enter key to continue restoring files.
2. If an archive, created by using the **backup** command, is made to a tape device with the device block size set to 0, it is necessary for you to have explicit knowledge of the block size that was used when the tape was created to restore from the tape.

- Multiple archives can exist on a single tape. When multiple archives are restored from the tape, the **restore** command expects the input device to be a no-retension-on-open, no-rewind-on-close tape device. Do not use a no-rewind tape device for restoring unless either the **-B**, **-s**, or **-X** flag is specified. For more information on using tape devices, see the **rmt** special file.

File system archives

File system archives are also known as i-node archives because the method used to archive the files. A file system name is specified with the **backup** command, and the files within that file system are archived based on their structure and layout within the file system. The **restore** command restores the files on a file system archive without any special understanding of the underlying structure of the file system.

When you restore the file system archives, the **restore** command creates and uses a file named **restoresymtable**. This file is created in the current directory. The file is necessary for the **restore** command to do incremental file system restores.

Note: Do not remove the **restoresymtable** file if you run incremental file system backups and restores.

The *File* parameter is ignored when you use either the **-r** or the **-R** flag.

File name Archives

File name archives are created by specifying a list of file names to archive to the **backup** command. The **restore** command restores the files from a file name archive without any special understanding of the underlying structure of the file system. The **restore** command allows for metacharacter to be used when you specify files for archive extraction. This process provides the capability to extract files from an archive that is based on pattern matching. A pattern file name must be enclosed in single quotations, and patterns must be enclosed in brackets (...).

About sparse files

Files in the operating system file system that contain long strings of Nulls can be stored efficiently when compared to the other files. If a string of Nulls spans an entire allocation block, that whole block is not stored on disk at all. Files where one or more blocks are omitted in this way are called sparse files. The missing blocks are also known as holes.

Note: Restores the non-sparse files as non-sparse because they were archived by the name format of the **backup** command for both packed and unpacked files. It is necessary to know the sparseness and non-sparseness of the file being restored before you archive the files. This check is required because by enabling the **-e** flag, the flag restores the sparse files as non-sparse. This flag must be enabled only if the files to be restored are non-sparse consisting of more than 4 KB Nulls. If the **-e** flag is specified during the restore operation, it successfully restores all normal files normally and non-sparse database files as non-sparse.

Flags

Table 6. Flags

Item	Descriptor
-a	Specified with the t and T option, the -a option displays the list of files in the archive, along with their permissions.
-B	Specifies that the archive must be read from standard input. Normally, the restore command examines the actual medium to determine the backup format. When you use a (pipe), this examination cannot occur. As a result, the archive is assumed to be in file system format, and the device is assumed to be standard input (-f).

Table 6. *Flags (continued)*

Item	Descriptor
-b <i>Number</i>	Specifies the number of 512-byte blocks for backups done by name. For backups that are done by i-node, the flag specifies the number of 1024-byte blocks to read in a single output. When the restore command reads from tape devices, the default is 100 for backups by name and 32 for backups by i-node. The read size is the number of blocks that are multiplied by the block size. The default read size for the restore command reading from tape devices is 51200 (100 * 512) for backups by name and 32768 (32 * 1024) for backups by i-node. The read size must be an even multiple of the tapes physical block size. If the read size is not an even multiple of the tapes physical block size and it is in fixed block mode (nonzero), the restore command tries to determine a valid value for <i>Number</i>. If successful, the restore command changes <i>Number</i> to the new value, write a message about the change to standard output, and continues. If unsuccessful in finding a valid value for <i>Number</i>, the restore command writes an error message to standard error and exits with a nonzero return code. Larger values for the <i>Number</i> parameter result in larger physical transfers from the tape device. The value of the -b flag is always ignored when the restore command reads from diskette. In this case, the command always reads in clusters that occupy a complete track.
-d	Indicates that, if the <i>File</i> parameter is a directory, all files in that directory must be restored. This flag can be used when the archive is in file name format.
-e	Specifies to not restore sparse files actively. If a file has a block that is aligned and sized areas that are Null populated, then the restore operation creates physical space for those file system blocks to be allocated and filled with Nulls. The file size that is specified in bytes corresponds to the space taken within the file system. This flag must be enabled only if files are to be restored are nonsparse consisting of more than 4 KB Nulls. If the -e flag is specified during restore, it successfully restores all normal files normally and nonsparse database files as nonsparse.
-E	The -E option extracts beginning at a specified volume number and requires one of the following arguments. If you omit the -E option, warn is the default behavior. force Fails the restore operation on a file if the fixed extent size or space reservation of the file cannot be preserved. ignore Ignores any errors in preserving extent attributes. warn Issues a warning if the space reservation or the fixed size of the file cannot be preserved.

Table 6. *Flags (continued)*

Item	Descriptor
-f <i>Device</i>	Specifies the input device. To receive input from a named device, specify the <i>Device</i> variable as a path name such as <code>/dev/rmt0</code>. To receive input from the standard output device, specify a - (minus sign). The - (minus) feature allows to pipe the input of the restore command from the dd command. You can also specify a range of archive devices. The range specification must be in the following format: <pre style="background-color: #f0f0f0; padding: 5px;">/dev/deviceXXX-YYY</pre> where <i>XXX</i> and <i>YYY</i> are whole numbers, and <i>XXX</i> must always be less than <i>YYY</i>; for example, <code>/dev/rfd0-3</code>. All devices in the specified range must be of the same type. For example, you can use a set of 8 mm, 2.3GB tapes or a set of 1.44MB diskettes. All tape devices must be set to the same physical tape block size. If the <i>Device</i> variable specifies a range, the restore command automatically goes from one device in the range to the next. After all the specified devices are exhausted, the restore command halts and requests that new volumes be mounted on the range of devices.
-h	Restores only the actual directory, not the files that are contained in it. This flag can be used when the archive is in file system format. This flag is ignored when used with the -r or -R flags.

Table 6. *Flags (continued)*

Item	Descriptor
-i	Restores the selected files interactively from a file system archive. The following are the subcommand for the -i flag: cdDirectory Changes the current directory to the specified directory. add [File] Specifies that the <i>File</i> parameter is added to the list of files to extract. If <i>File</i> is a directory, that directory and all the files that are contained in it are added to the extraction list (unless the -h flag is used). If <i>File</i> is not specified, the current directory is added to the extraction list. delete [File] Specifies that the <i>File</i> parameter is to be removed from the list of files to be extracted. If <i>File</i> is a directory, that directory and all the files that are contained in it are removed from the extraction list (unless the -h flag is used). ls [Directory] Displays the directories and files that are contained within the <i>Directory</i> parameter. Directory names are displayed with a / (slash) after the name. Files and directories, within the specified directory, that are on the extraction list are displayed with an * (asterisk) before the name. If verbose mode is on, the i-node number of the files and directories is also displayed. If the <i>Directory</i> parameter is not specified, the current directory is used. extract Restores all the directories and files on the extraction list. pwd Displays the full path name of the current directory. verbose Causes the ls subcommand to display the i-node number of files and directories. More information about each file is also displayed as it is extracted from the archive. setmodes Sets the owner, mode, and time for all directories added to the extraction list. quit Causes restore to exit immediately. Any files on the extraction list are not restored. help Displays a summary of the subcommand.
-l	Specified with the -t and -T option. When specified, displays a detailed list of files, which includes the timestamp, file permissions, file size, owner, and group. The -l option overrides the -a option.

Table 6. Flags (continued)

Item	Descriptor
-M	Sets the access and modification times of restored files to the time of restoration. If a restored file is an archive that is created by the ar command, the modification times in all the member headers are also set to the time of restoration. You can specify the -M flag only when you are restoring individually named files and only if the -x or -X flags are also specified. When the -M flag is not specified, the restore command maintains the access and modification times as displayed on the backup medium. The -M flag is used when the data is in the AIX 4.2 backup by-i-node or by-name format.
-m	Renames restored files to the file's i-node number as it exists on the archive. This function is useful if a few files are being restored and you want these files that are restored under a different file name. Since any restored archive members are renamed to their i-node numbers, directory hierarchies and links are not preserved. Directories and hard links are restored as regular files. The -m flag is used when the archive is in file system format.
-n	By default the restore command restores any ACLs, PCLs, or named extended attributes in the archive. The -n flag causes the restore command to ignore any ACLs, PCLs, or named extended attributes in the archive and not restore them. When the archived files contain Encrypted file system (EFS) information, the EFS extended attributes are restored even if the -n flag is specified. For more information about EFS restoration, see Backup and restore in Security.
-Pstring	Restore only the file attributes. Does not restore the file contents. If the file specified does not exist in the target directory path, the files are not created. This flag restores file attributes selectively depending on the flags that are specified in the string parameter. String parameter can be a combination of the following characters: A restore all attributes. a restore only the permissions of the files. o restore only the ownership of the files. t restore only the timestamp of the files. c restore only the ACL attributes of the files. Note: Among the existing options for the restore command, options v, h, b, s, f, B, d, and q are valid with the P option. The P option can be used with both file name and file system archives. If the File argument is a symbolic link, then the metadata of the target file is modified and not that of the symbolic link.  Warning: Usage of the -P flag overwrites the attributes of files that are owned by another user when run by the superuser.
-Q	Specifies that the command must exit when an error is encountered, for backups done by name. This process does not attempt to recover and continue processing the archive, when an error occurs.

Table 6. *Flags (continued)*

Item	Descriptor
-q	Specifies that the first volume is ready to use and that the restore command cannot prompt you to mount the volume and hit Enter. If the archive spans multiple volumes, the restore command prompts you for the subsequent volumes.
-r	Restores all files in a file system archive. The -r flag is only used to restore complete level 0 backups or to restore incremental backups after a level 0 backup is restored. The <code>restoresymtable</code> file is used by restore to pass information between incremental restores. This file must be removed when the last incremental backup is restored. The <i>File</i> parameter is ignored when use the -r flag.
-R	Requests a specific volume of a multiple-volume, file system archive. The -R flag allows a previously interrupted restore to be restarted. The <i>File</i> parameter is ignored when you use the -R flag. When the restore command is restarted, it functions similar to the -r flag.
-s <i>SeekBackup</i>	Specifies the backup to seek and restore on a multiple-backup tape archive. The -s flag is only applicable when the archive is written to a tape device. To use the -s flag properly, a no-rewind-on-close and no-retension-on-open tape device, such as <code>/dev/rmt0.1</code> or <code>/dev/rmt0.5</code> , must be specified. If the -s flag is specified with a rewind tape device, the restore command displays an error message and exits with a nonzero return code. If a no-rewind tape device is used and the -s flag is not specified, a default value of -s1 is used. The value of the <i>SeekBackup</i> parameter must be in the range of 1 to 100 inclusive. It is necessary to use a no-rewind-on-close, no-retension-on-open tape device because of the behavior of the -s flag. The value that is specified with -s is relative to the position of the tapes read/write head and not to an archives position on the tape. For example, to restore the first, second, and fourth backups from a multiple-backup tape archive, the respective values for the -s flag would be -s1, and -s2.
-t	Displays information about the backup archive. If the archive is in file system format, a list of files that are found on the archive is written to standard output. The name of each file is preceded by the i-node number of the file as it exists on the archive. The file names that are displayed are relative to the root (/) directory of the file system that was backed up. If the <i>File</i> parameter is not specified, all the files on the archive are listed. If the <i>File</i> parameter is used, then just that file is listed. If the <i>File</i> parameter refers to a directory, all the files that are contained in that directory are listed. If the archive is in file name format, information that is contained in the volume header is written to standard error. This flag can be used to determine whether the archive is in the file name or the file system format.
-T	Displays information about the backup archive. If the archive is in file name format, the information that is contained in the volume header is written to standard error, and a list of files that are found on the archive is written to standard output. The <i>File</i> parameter is ignored for file name archives. If the archive is in file system format, the behavior is identical to the -t flag.

Table 6. *Flags (continued)*

Item	Descriptor
-v	Displays information when the file name is restored . If the archive is in file name format and either the -x or -T flag is specified, the size of the file as it exists on the archive is displayed in bytes. Directory, block, or character device files are archived with a size of 0. Symbolic links are listed with the size of the symbolic link. Hard links are listed with the size of the file, which is how they are archived. Once the archive is read, a total of these sizes is displayed. If the archive is in file system format, directory and nondirectory archive members are distinguished.
-x	Restores individually named files that are specified by the <i>File</i> parameter. If the <i>File</i> parameter is not specified, all the archive members are restored. If the <i>File</i> parameter is a directory and the archive is in file name format, only the directory is restored. If the <i>File</i> parameter is a directory and the archive is in file system format, all the files that are contained in the directory are restored. The file names that are specified by the <i>File</i> parameter must be the same as the names shown by the restore-T command. Files are restored with the same name they were archived with. If the file name was archived by using a relative path name (./filename), the file is restored relative to the current directory. If the archive is in file system format, files are restored relative to the current directory. The restore command automatically creates any needed directories. When you use this flag to restore file system backups, you are prompted to enter the beginning volume number. The restore command allows for shell-style pattern matching metacharacters to be used when files for archive extraction is specified . The rules for matching metacharacters are the same as used in shell pathname "globbing," namely: * (asterisk) Matches zero or more characters, but not a '.' (period) or '/' (slash). ? (question mark) Matches any single character, but not a '.' (period) or '/' (slash). [] (brackets) Matches any one of the characters that are enclosed within the brackets. If a pair of characters that are separated by a dash is contained within the brackets, the pattern matches any character that lexically falls between the two characters in the current local. Additionally, a '.' (period) or a '/' (slash) within the brackets does not match a '.' (period) or a '/' (slash) in a file name. \ (backslash) Matches the immediately following character, preventing its possible interpretation as a metacharacter.
-XVolumeNumber	Begins restoring from the specified volume of a multiple-volume, file name backup. When the restore command is started, the command behaves similar to the -x flag. The -X flag applies to file name archives only.
-y	Continues restoring when tape errors are encountered. Normally, the restore command request input to continue. In either case, all data in the read buffer is replaced with zeros. The -y flag applies only when the archive is in file system format.
-?	Displays a usage message.

Exit Status

This command returns the following exit values:

Table 7. Exit Status

Item	Descriptor
0	Successful completion.
>0	An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To list the names of files in either a file name or file system archive on the diskette device `/dev/rmt0`, type:

```
restore -Tqf /dev/rmt0
```

The archive is read from the specified `/dev/rmt0` restore device. The names of all the files and directories that are contained in the archive are displayed. For file system archives, the file name is preceded by the i-node number of the file as it exists on the archive. The `-q` flag tells the **restore** command that the first volume is available and is ready to be read. As a result, you are not prompted to mount the first volume.

2. To restore a specific file, type:

```
restore -xvqf myhome.bkup system.data
```

This command extracts the file `system.data` into the current directory from the archive `myhome.bkup`. The archive in this example is in the current directory. File and directory names must be specified as they are displayed when the `-T` flag is used. The `-v` flag displays information during the extraction. This example applies to both file name and file system archives.

3. To restore a specific directory and the contents of that directory from a file name archive, type:

```
restore -xdvqf /dev/rmt0 /home/mike/tools
```

The `-x` flag tells **restore** to extract files by their file name. The `-d` tells **restore** to extract all the files and subdirectories in the `/home/mike/tools` directory. File and directory names must be specified as they are displayed when you use the `-T` flag. If the directories do not exist, they are created.

4. To restore a specific directory and the contents of that directory from a file system archive, type:

```
restore -xvqf /dev/rmt0 /home/mike/tools
```

This command extracts files by file name. File and directory names must be specified as they are displayed when you use the `-T` flag. If the directories do not exist, they are created.

5. To restore an entire file system archive, type:

```
restore -rvqf /dev/rmt0
```

This command restores the entire file system that is archived on the tape device, `/dev/rmt0`, into the current directory. This example assumes you are in the root directory of the file system to be

restored. If the archive is part of a set of incremental file system archives, the archives must be restored in increasing backup-level order beginning with level 0 (for example, 0, 1, and 2).

6. To restore the fifth and ninth backups from a single-volume, multiple-backup tape, type:

```
restore -xvqs 5 -f/dev/rmt0.1
restore -xvqs 4 -f/dev/rmt0.1
```

The first command extracts all files from the fifth archive on the multiple-backup tape that is specified by `/dev/rmt0.1`. The `.1` designator specifies the tape device that is not retensioned when it is opened and rewound when it is closed. It is necessary to use a no-rewind-on-close, no-retension-on-open tape device because of the behavior of the `-s` flag. The second command extracts all the files from the fourth archive (relative to the current location of the tape head on the tape). After the fifth archive is restored, the tape read/write head is in a position to read the archive. To extract the ninth archive on the tape, you must specify a value of 4 with the `-s` flag. This is because the `-s` flag is relative to your position on the tape and not to an archive's position on the tape. The ninth archive is the fourth archive from your current position on the tape.

7. To restore the fourth backup, which begins on the sixth tape on a 10-tape multiple-backup archive, put the sixth tape into the tape drive and type:

```
restore -xcs 2 -f /dev/rmt0.1 /home/mike/manual/chap3
```

Assuming the fourth backup is the second backup on the sixth tape, specifying `-s 2` advances the tape head to the beginning of the second backup on this tape. The **restore** command then restores the specified file from the archive. If the backup continues onto subsequent volumes and the file is not restored, the **restore** command instructs you to insert the next volume until the end of the backup is reached. The `-f` flag specifies the no-rewind, no-retension tape device name.

Note: The `-s` flag specifies the backup number relative to the tape inserted in the tape drive, not to the overall 10-tape archive.

8. To improve the performance on streaming tape devices, pipe the `dd` command to the **restore** command by typing:

```
dd if=/dev/rmt0 bs=64b | restore -xf- -b64
```

The `dd` command reads the archive from the tape by using a block size of 64 512-byte blocks and writes the archive to standard output. The **restore** command reads the standard input by using a block size of 64 512-byte blocks. The value of the block size that is used by the `dd` command to read the archive from the tape must be an even multiple of the block size that was used to create the tape with the **backup** command. For example, the following **backup** command cannot be used to create the archive that this example extracts:

```
find /home -print | backup -ivqf/dev/rmt0 -b64
```

This example applies to archives in file name format only. If the archive was in file system format, the **restore** command must include the `-B` flag.

9. To improve the performance of the **restore** command on the 9348 Magnetic Tape Unit Model 12, you can change the block size by typing:

```
chdev -l DeviceName -a BlockSize=32k
```

10. To restore non-sparse database files, type:

```
restore -xef /dev/rmt0
```

11. To restore files that were sparse before archive as sparse, type:

```
restore -xf /dev/rmt0
```

12. To restore only the permissions of the files from the archive, type:

```
restore -Pa -vf /dev/rmt0
```

13. To restore only the ACL attributes of the files from the archive, type:

```
restore -Pc -vf /dev/rmt0
```

14. To view the table of contents along with the file permissions, type:

```
restore -Ta -vf /dev/rmt0
```

15. To view the table of contents of file name archive along with the timestamps and file permissions, type:

```
restore -Tl -vf /dev/rmt0
```

16. To view the table of contents of file system archive along with the timestamps and file permissions, type:

```
restore -tl -vf /dev/rmt0
```

Files

Table 8. Files

Item	Descriptor
/usr/sbin/restore	Contains the restore command.

restorevgfiles Command

Purpose

Restores files from a backup source.

Syntax

```
restorevgfiles [ -b blocks ] [ -f device ] [ -a ] [ -n ] [ -s ] [ -d path ] [ -D ] [ file_list ]
```

Description

The **restorevgfiles** command restores files from tape, file, CD-ROM, or their volume group backup source. The **restorevgfiles** command also works for multi-volume backups such as multiple CDs, DVDs, USB disks, or tapes.

The **restorevgfiles** and **listvgbackup -r** commands perform identical operations and should be considered interchangeable. The **restorevgfiles** command automatically applies the **-r** flag. The **-r** flag, while redundant, is retained for compatibility purposes and will cause no unusual behavior if specified. For a complete description of the **-r** flag, see the **listvgbackup** command.

Flags

Item	Description
-b <i>blocks</i>	Specifies the number of 512-byte blocks to read in a single input operation, as defined by the <i>blocks</i> parameter. If the <i>blocks</i> parameter is not specified, the number of blocks read will default to 100.
-f <i>device</i>	Specifies the type of device containing the backup (file, tape, CD-ROM, or other source) as defined by the <i>device</i> parameter. When -f is not specified, <i>device</i> will default to /dev/rmt0 .

Item	Description
-a	Verifies the physical block size of the tape backup, as specified by the -b block flag. You may need to alter the block size if necessary to read the backup. The -a flag is valid only when a tape backup is used.
-n	Does not restore ACLs, PCLs, or extended attributes.
-s	Specifies that the backup source is a user volume group and not rootvg .
-d path	Specifies the directory path to which the files will be restored, as defined by the <i>path</i> parameter. If the -d parameter is not used, the current working directory is used. This can be a problem if the current working directory is root. We recommend writing to a temporary folder instead of to root.
-D	Produces debug output.

Parameters

Item	Description
<i>file_list</i>	Identifies the list of files to be restored. The full path of the files relative to the current directory should be specified in the space-separated list. All files in the specified directory will be restored unless otherwise directed. If you are restoring all files in a directory, we recommend writing to a temporary folder instead of to root.

Examples

1. To read the backup stored at **/dev/cd1** and restore all files to the **/data/myfiles** directory, enter:

```
restorevgfiles -f /dev/cd1 -s -d /data/myfiles
```

2. To read the user vg backup from the default device at 20 512-byte blocks at a time and restore the **/myapp/app.h** file to the current directory, enter:

```
restorevgfiles -b 20 -s ./myapp/app.h
```

3. To read the backup stored at **/dev/cd1** and restore the **/myapp/app.c** file to the **/data/testcode** directory, enter:

```
restorevgfiles -f /dev/cd1 -s -d /data/testcode ./myapp/app.c
```

4. To read the backup stored at **/dev/usbms0** and restore all files to the **/data/myfiles** directory, enter the following command:

```
restorevgfiles -f /dev/usbms0 -s -d /data/myfiles
```

Files

Item	Description
/usr/bin/restorevgfiles	Contains the restorevgfiles command

restvg Command

Purpose

Restores the user volume group and all its containers and files.

Syntax

```
restvg [ -b Blocks ] [ -d FileName ] [ -f Device ] [ -l ] [ -q ] [ -r ] [ -s ] [ -n ] [ -P PPsize ] [ -S ] [ DiskName ... ]
```

Description

The **restvg** command restores the user volume group and all its containers and files as specified in the `/tmp/vgdata/vgname/vgname.data` file, where *vgname* is the name of the volume group. The `/tmp/vgdata/vgname/vgname.data` file is contained in the backup image that is created by the **savevg** command.

The **restvg** command restores a user volume group. The **bosinstall** routine reinstalls the root volume group (rootvg). If the **restvg** command encounters a rootvg volume group in the backup image, the **restvg** command exits with an error.

If a **yes** value has been specified in the EXACT_FIT field of the logical_volume_policy stanza of the `/tmp/vgdata/vgname/vgname.data` file, the **restvg** command uses the map files to preserve the placement of the physical partitions for each logical volume. The target disks must be of the same size or larger than the source disks specified in the source_disk_data stanzas of the *vgname.data* file.

Notes:

- To view the files in the backup image, use **restore** command with the **-T** flag.
- To restore individual files from the backup image, use the **restore** command with the **-x** flag.
- When you run the **varyonvg** command on the volume group, the logical track group (LTG) size is set to the common max transfer size of the disks.

Flags

Table 9. Flags

Item	Description
-b <i>Blocks</i>	Specifies the number of 512-byte blocks to read in a single input operation. If this parameter is not specified, the default of 100 is used by the restore command. Larger values result in larger physical transfers to tape devices.
<i>DiskName...</i>	Specifies the names of disk devices to be used instead of the disk devices listed in the <i>vgname.data</i> file. The target disk devices must be defined as empty physical volumes, which means that they must contain a physical volume identifier and must not belong to a volume group. If the target disk devices are new, they must be added to the system by using the mkdev command. If the target disk devices belong to a volume group, they must be removed from the volume group by using the reducevg command.
-d <i>FileName</i>	The -d flag is an optional flag. When the -d flag is specified, it must be followed by a file name. This file is used as the <i>vgname.data</i> file instead of the one contained within the backup image that is restored. The file name can be specified by either a relative or an absolute path name.
-f <i>Device</i>	Specifies the device name of the backup media. The default is <code>/dev/rmt0</code> .

Table 9. Flags (continued)

Item	Description
-l	Displays useful information about a volume group backup. This flag requires the -f device flag. This flag causes restvg to display information such as volume group, date and time when the backup was made, uname output from backed up system, oslevel, recommended maintenance and technology levels, backup size in megabytes, and backup shrink size in megabytes. The shrink size is the size of the data on all filesystems. The full size is the total size of each filesystem (unused + data). The -l flag also displays the logical volume and filesystem information of the backed up volume group, equivalent to running <code>lsvg -l vname</code>.
-n	Specifies that the existing MAP files are ignored. The -n flag overrides the value of the EXACT_FIT field in the logical_volume_policy stanza of the <code>vname.data</code> file.
-P PPSize	Specifies the number of megabytes in each physical partition. If not specified, restvg uses the best value for the <i>PPsize</i>, dependent upon the largest disk that is restored. If the <i>PPsize</i> value is not the same as the size specified in the <code>vname.data</code> file, the number of partitions in each logical volume is altered as per the new <i>PPsize</i>. If the value of the <i>PPsize</i> that is specified is smaller than the appropriate disk size value, the larger <i>PPsize</i> is used. If the value of the <i>PPsize</i> that is specified is larger than the appropriate disk size value, the specified larger <i>PPsize</i> value is used.
-q	Specifies that the usual prompt is not displayed before the restoration of the volume group image. If this flag is not specified, the prompt displays the volume group name and the target disk-device names.
-r	Re-creates only a volume group structure. The -r flag allows restvg command to create (for the specified backup <i>FileName</i> or <i>Device</i>) the volume group, logical volumes, and filesystems, from the backup, without restoring any files or data. This feature is useful for users who use third-party software for restoring data and need all the AIX logical volume structure in place. Note: The -r flag can be used with either the -f Device flag or the -d FileName flag. This restriction is because the restvg command requires a backup image or <code>vname.data</code> file to get all the information it needs to re-create the logical volume structure of the volume group wanted.

Table 9. Flags (continued)

Item	Description
-s	Specifies that the logical volumes be created at the minimum size possible to accommodate the file systems. This size is specified by the value of LV_MIN_LPS field in the lv_data stanza of the <i>vgname</i>.data file where <i>vgname</i> is the name of the volume group. The -s flag overrides the values of the SHRINK and EXACT_FIT fields in the logical_volume_policy stanza of the <i>vgname</i>.data file. The -s flag causes the same effect as values of SHRINK=yes and EXACT_FIT=no would cause.
> > -s	> Specifies that the target physical volumes must have the same encryption policy as the source physical volumes. The source physical volumes are considered encrypted when at least one of the source physical volumes is encrypted. However, the target physical volumes are considered encrypted when all the target physical volumes are encrypted. < > If the source physical volumes are not encrypted and target physical volumes are encrypted, the restvg command asks to disable the target physical volumes encryption. If the source physical volumes are encrypted and the target physical volumes are not encrypted, the restvg command asks to enable target physical volumes encryption. < Notes: • When you use the restvg command with the -S flag, out of all the authentication types, the passphrase and PKS authentication type is only restored in the user volume group. You can add other authentication types after the restoration of the user volume group is complete. • > The encryption policy applies to passphrase authentication and Platform keystore (PKS) encryption authentication method. < • > The restvg -S command does not remove the additional encryption methods from the target physical volumes when the backup image of the volume group contains physical volumes that are encrypted only with a passphrase. If the target physical volumes are encrypted by using methods such as key files, key servers, or any other encryption mechanisms, these additional encryption methods remains on the target physical volumes. < < <

Examples

1. To restore the volume group image from the /dev/rmt1 device, onto the hdisk2 and hdisk3 disks, enter the following command:

```
restvg -f/dev/rmt1 hdisk2 hdisk3
```

2. To restore the volume group image that is saved in `/mydata/myvg` file onto the disks that are specified in the `vgname . data` file that is contained within the backup image, enter the following command:

```
restvg -f/mydata/myvg
```

3. To re-create the volume group logical volume structure without restoring any files by using only the `vgname . data` file `/home/my_dir/my_vg . data`, enter the following command:

```
restvg -r -d /home/my_dir/my_vg.data
```

Note: `vgname . data` files can be created for a volume group by using the **mkvgdata** command.

4. To re-create the volume group logical volume structure without restoring any files by using the `vgname . data` file inside the volume group backup that is on the tape in `/dev/rmt0`, enter the following command:

```
restvg -r -f /dev/rmt0
```

5. To display volume group information about the volume group backed up on the tape in `/dev/rmt0`, enter the following command:

```
restvg -l -f /dev/rmt0
```

6. To restore the volume group image from the `/dev/usbms0` device, onto the disks that are specified in the `vgname . data` file that is contained within the backup image, enter the following command:

```
restvg -f /dev/usbms0
```

Note: For more information about backing up a volume group, see the **listvgbackup** command. To restore individual files from a volume group backup, see the **restorevgfiles** command.

restwpar Command

Purpose

Restores a workload partition.

Syntax

```
restwpar [ -a ] [ -A ] [ -b Blocks ] [ -B devexportsFile ] [ -C ] [ -d Directory ] [ -f Device ] [ -F ] [ -h  
hostName ] [ -i imagedataFileName ] [ -k ] [ -K ] [ -M mkwparFlags ] [ -n WparName ] [ -r ] [ -s ] [ -S { a | A  
| f | F | n } ] [ -U ] [ -w wparSpecificationFile ]
```

Description

The **restwpar** command creates a workload partition from a workload partition backup image that was created by the **savewpar**, **mkcd**, or **mkdvd** command.

Warning: The **restwpar** command should not be run while an AIX Live Update operation is in progress.

A workload partition backup image contains an `image . data` file and a workload partition specification file that is used to establish the characteristics of workload partition `WparName`. You can use the `-i` and `-w` flags to override these default files.

If you do not specify the `-f` flag, the `/dev/rmt0` device is used as the input device.

If you specify a value of Yes in the EXACT_FIT field of the `logical_volume_policy` stanza of the `/tmp/wpardata/WparName/image . data` file, the **restwpar** command uses the map files to preserve the placement of the physical partitions for each logical volume.

If user volume groups are configured with a rootvg WPAR, then they are not automatically imported after restoring a rootvg WPAR.

Notes:

- To view the files in the backup image or to restore individual files from the backup image, use the **lssavewpar**, **restwparfiles**, or **restore** command with the **-T** or the **-x** flag.
- For shared workload partitions (WPARs), if the installation history of the source system is different from the installation history of the target system, the **restwpar** command and the **syncroot** command might fail for few filesets. You might see a failure message that is similar to the following example for the **syncroot** command, at the end of the **restwpar** operation:

```
syncroot: Error synchronizing installp software
syncroot: Returns Status = FAILURE
```

You must restore or migrate the shared WPAR to a logical partition (LPAR) that has the installation history similar to the installation history of the source LPAR.

Flags

Item	Description
-a	Automatically resolves conflicting static settings if required. Resolvable settings are name, host name, base directory, and network configuration.
-A	Starts the workload partition each time when the <code>/etc/rc.wpars</code> command is run, which is added to the global <code>/etc/inittab</code> to run on each system start. The default is not to start the workload partition automatically.
-b <i>Blocks</i>	Specifies the number of 512-byte blocks to read in a single input operation. If you do not specify the <i>Blocks</i> parameter, the default value of 100 is used by the restore command. Larger values result in larger physical transfers to tape devices.
-B <i>devexportsFile</i>	Specifies a substitute file that can be used as the master device exports file. This file must match the format of a Device exports File. If you do not specify a file name, the <code>/etc/wpars/devexports</code> file is used.
-C	Forces the creation of the named workload partition, even when a compatibility check fails between the system from the backup image and the system where the backup is being restored. If the workload partition is not compatible with the target system. It might not be operable. If the operating system of the global system is at a later technology level or service pack level than the WPAR that has different modification or fix levels in the VRMF (version, release, modification and fix level), the workload partition (WPAR) can be synchronized with the new global system. Different factors affect the success of the synchronization. Review the logs after the synchronization operation is complete. Any updates that are applied to the new global system must be committed, and the updates to the WPAR must be committed before you back up the WPAR. If the new global system is installed on a system that is running AIX 6100-08 or 7100-02 technology levels, or earlier, you must run the cp_bos_updates command before you restore the workload partition for the synchronization to work.
-d <i>Directory</i>	Specifies a base directory for the workload partition. If you do not specify a directory name, the directory name from the WPAR specification file is used.
-f <i>Device</i>	Specifies the device name of the backup media. The default value is <code>/dev/rmt0</code> .

Item	Description
-F	Forces the creation of the named workload partition. If the named workload partition exists, it is stopped if active, and then removed, before the new workload partition is created.
-h <i>hostname</i>	Specifies a host name for the workload partition. If not specified, the mkwpar command uses the workload partition name for the host name.
-i <i>imagedataFileName</i>	An optional flag that specifies a file name. The file is used as the <code>image.data</code> file instead of the one contained within the backup image that is being restored.
-k	Creates logical volumes with minimum sizes from the backup.
-K	Creates the post-installation customization script.
-M <i>mkwparFlags</i>	Specifies the flags to pass directly to the mkwpar command to create the workload partition. The -M flag is used to pass other flags to the mkwpar command. If a flag is passed through its own option and through the -M flag, both flags are passed to the mkwpar command. Note: The <i>mkwparFlags</i> value cannot include the -i and -f flags as these flags are reserved for use by the restwpar command. Specifying the -i or -f flag as the <i>mkwparFlags</i> value causes an error.
-n <i>WparName</i>	Specifies the name for the workload partition to be created. If you do not specify the -n flag, the <i>WparName</i> is taken from the WPAR specification file.
-r	Duplicates the network name resolution configuration from the global system. The following files, if they exist, are copied into the workload partition: • /etc/resolv.conf • /etc/hosts • /etc/netsvc.conf • /etc/irs.conf • /etc/networks If the NSORDER environment variable is defined in the calling environment, it is added to the /etc/environment file of the workload partition.
-s	Starts the workload partition after it is created.

Item	Description
-S { a A f F n }	Specifies the type of synchronization to use after files are restored from the backup to synchronize the levels of software in the workload partition with the levels of the software in the global environment. a Causes additional installations with no removal of software. This option is the default.
	A Causes additional installations with no removal of software, and ignores any errors in synchronization.
	Important: If you specify -S A, the workload partition might be in an unusable state.
	f Causes additional installations, software rejection, and deinstallation.
	F Causes additional installations, software rejection, and deinstallation. This option ignores any errors in synchronization.
	Important: If you specify -S F, the workload partition might be in an unusable state.
	n Prevents the synchronization processing after the files are restored.
	Important: If you specify -S n, the workload partition might be in an unusable state.
-U	Specifies that the existing MAP files are ignored. The -U flag overrides the value of the EXACT_FIT field in the logical_volume_policy stanza of the <i>WparName</i> .data file.
-w <i>wparSpecificationFile</i>	An optional flag that specifies a file name. The file is used as the WPAR specification file rather than the version in the WPAR backup image by the mkwpar command.

Examples

1. To restore the workload partition image from the /dev/rmt1 device, enter the following command:

```
restwpar -f/dev/rmt1
```

2. To restore the workload partition image that is saved in the /mydata/wpar.img file with name mywpar and base directory /wpars/mywpar, enter the following command:

```
restwpar -f/mydata/wpar.img -n mywpar -d /wpars/mywpar
```

3. To restore the workload partition image from the /dev/usbms0 device, enter the following command:

```
restwpar -f/dev/usbms0
```

restwparfiles Command

Purpose

Restores files from a workload partition backup source.

Syntax

restwparfiles [**-b** *blocks*] [**-f** *device*] [**-a**] [**-m**] [**-n**] [**-d** *path*] [**-D**] [**-V**] [*file_list*]

Description

The **restwparfiles** command restores files from tape, file, CD-ROM, or other workload partition backup source. The **restwparfiles** command also works for multivolume backups such as multiple CDs, DVDs, USB disks, or tapes.

Flags

Item	Description
-a	Verifies the physical block size of the tape backup, as specified by the -b <i>blocks</i> flag. You might need to alter the block size to read the backup. The -a flag is valid only when you specify the device in the -f flag as tape.
-b <i>blocks</i>	Specifies the number of 512-byte blocks to read in a single input operation, as defined by the <i>blocks</i> parameter. If you do not specify the <i>blocks</i> parameter, the default number of blocks to read is 100.
-d <i>path</i>	Specifies the directory path where the files are restored, as defined by the <i>path</i> parameter. If you do not specify the -d flag, the current working directory is used. Restriction: The directory path where the files are restored must not be root (/) in the global environment, either through the use of -d / or if the current working directory is / and the -d flag is not specified.
-D	Produces debug output.
-f <i>device</i>	Specifies the device containing the backup (file, tape, CD-ROM, or other source) as defined by the <i>device</i> parameter. When you do not specify the -f flag, the default device is /dev/rmt0 .
-m	Restores only informational and control files from the image. Use the flag to restore the image.data and wpar.spec files from the backup image. Files are restored under the ./savewpar_dir/ directory.
-n	Specifies that ACLs, PCLs, or extended attributes are not to be restored.
-V	Verifies a tape backup. The -V flag requires the -f <i>device</i> flag and can be used to specify only tape devices. The -V flag causes the restwparfiles command to verify the readability of each file header on the volume group backup and print any errors that occur to the standard error log (stderr) file.

Parameters

Item	Description
<i>file_list</i>	Identifies the list of files to be restored. Specify the full path of the files relative to the current directory in the space-separated list. All files in the specified directory are restored unless directed. If you are restoring all files in a directory, write to a temporary folder instead of the root directory.

Examples

1. To read the backup stored on the **/dev/cd1** device and restore all files to the **/data/myfiles** directory, enter the following command:

```
restwparfiles -f /dev/cd1 -d /data/myfiles
```

2. To read the backup from the default device at twenty 512-byte blocks at a time and restore the **/myapp/app.c** file to the current directory, enter the following command:

```
restwparfiles -b 20 ./myapp/app.h
```

3. To read the backup stored on the **/dev/cd1** device and restore the **/myapp/app.c** file to the **/data/testcode** directory, enter the following command:

```
restwparfiles -f /dev/cd1 -d /data/testcode ./myapp/app.c
```

4. To read the backup stored at **/dev/usbms0** and restore all files to the **/data/myfiles** directory, enter the following command:

```
restwparfiles -f /dev/usbms0 -d /data/myfiles
```

resumevsd Command

Purpose

Activates an available virtual shared disk.

Syntax

```
resumevsd [-p | -b | -l server_list] {-a | vsd_name ...}
```

Description

The **resumevsd** command brings the specified virtual shared disks from the suspended state to the active state. The virtual shared disks remains available. Read and write requests which had been held while the virtual shared disk was in the suspended state are resumed.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit vsd_mgmt
```

and select the **Resume a Virtual Shared Disk** option.

Under normal circumstances, you should not issue this command. The recoverable virtual shared disk subsystem uses this command to manage shared disks in a controlled manner. If you issue this command, the results may be unpredictable.

Flags

-p

Specifies that the primary server node defined for the global volume group is to be the active server. The -p flag is not valid for CVSD.

-b

Specifies that the secondary server node defined for the global volume group is to be the active server. The -b flag is not valid for CVSD.

-a

Specifies that all the virtual shared disks that have been defined are to be resumed.

-l

Passes the `server_list` to the driver.

Parameters

`vsd_name`

Specifies a virtual shared disk.

Security

You must have `root` authority to run this command.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **`starttrdomain`** command. To bring a particular node online in an existing peer domain, use the **`starttrnode`** command. For more information on creating and administering an RSCT peer domain, refer to *RSCT Administration Guide* .

Under normal circumstances, you should not issue this command. The recoverable virtual shared disk subsystem uses this command to manage shared disks in a controlled manner. If you issue this command, the results may be unpredictable.

Examples

To bring the virtual shared disk **`vsd1vg1n1`** from the suspended state to the active state, enter:

```
resumevsd vsd1vg1n1
```

Location

`/opt/rsct/vsd/bin/resumevsd`

rev Command

Purpose

Reverses characters in each line of a file.

Syntax

`rev` [*File ...*]

Description

The **`rev`** command copies the named files to standard output, reversing the order of characters in every line. If you do not specify a file, the **`rev`** command reads standard input.

Examples

To reverse characters in each line of a file, type:

```
rev file
```

If the `file` file contains the text:

```
abcdefghi  
123456789
```

then the **rev** command displays:

```
ihgfedcba
987654321
```

Files

Item	Description
/usr/bin/rev	Contains the rev command.

revnetgroup Command

Purpose

Reverses the listing of users and hosts in network group files in NIS maps.

Syntax

```
/usr/sbin/revnetgroup { -h | -u } [ File ]
```

Description

The **revnetgroup** command reverses the order in which hosts and users are listed in the **/etc/netgroup** file. The **revnetgroup** command is called from the **/var/yp/Makefile** file to produce output for creating either the **netgroup.byuser** or **netgroup.byhost** NIS map. Each line in the output file begins with a key formed by concatenating the host or user name with the domain name. Following the key is a list of groups to which the host or user belongs. The list is preceded by a tab, and each group is separated by a comma.

Note: The list of groups does not use the names of universal groups (groups that include all users in the network). Universal groups are listed under * (asterisk).

The **revnetgroup** command takes an optional file name if the default **/etc/netgroup** file is not desired. This feature provides users with flexibility to create custom network group maps.

Flags

Item	Description
-h	Produces output for creating the netgroup.byhost map.
-u	Produces output for creating the netgroup.byuser map.

Examples

1. To cause the **/etc/netgroup** file to list user names before host names, modify the appropriate stanza in the **/var/yp/Makefile** to read:

```
revnetgroup -u
```

2. To create a new network group file, called newgroup, in the **/etc** directory, modify the appropriate stanza in the **/var/yp/Makefile** to read:

```
revnetgroup -h newgroup
```

The **-h** flag used in this example causes the new **/etc/newgroup** file to list host names before user names.

Files

Item	Description
<code>/etc/netgroup</code>	Contains lists of users and hosts in network groups.
<code>/var/yp/Makefile</code>	Contains rules for making NIS maps.

rexcd Daemon

Purpose

Executes programs for remote machines.

Syntax

`/usr/sbin/rpc.rexd`

Description

The **rexcd** daemon executes programs for remote machines when a client issues a request to execute a program on a remote machine. The **inetd** daemon starts the **rexcd** daemon from the `/etc/inetd.conf` file.

Noninteractive programs use standard file descriptors connected directly to TCP connections. Interactive programs use pseudo-terminals, similar to the login sessions provided by the **rlogin** command. The **rexcd** daemon can use the network file system (NFS) to mount the file systems specified in the remote execution request. Diagnostic messages are normally printed on the console and returned to the requester.

Note: A root user cannot execute commands using **rexcd** client programs such as the **on** command.

Files

Item	Description
<code>/tmp_rex/rexd</code>	Contains temporary mount points for remote file systems.
<code>/etc/exports</code>	Lists the directories that the server can export.
<code>inetd.conf</code>	Starts RPC daemons and other TCP/IP daemons.
<code>/etc/passwd</code>	Contains an entry for each user that has permission to log in to the machine.

rexec Command

Purpose

Executes commands one at a time on a remote host.

Syntax

rexec [-a] [-d | -n] [-i] *Host Command*

Description

The `/usr/bin/rexec` command executes a command on the specified remote host.

The **rexec** command provides an automatic login feature by checking for a `$HOME/.netrc` file that contains the user name and password to use at the remote host. If such an entry is not found or if your system is operating in secure mode (see the **securetcip** command), the **rexec** command prompts for a

valid user name and password for the remote host. In both cases, **rexec** causes **rexecd** on the remote system to use the default compat login authentication method for the user. **rexecd** does not look at the **/etc/security/user** file on the remote system for alternative authentication methods. You can also override the automatic login feature by specifying the **-n** flag on the **rexec** command line.

Flags

Item	Description
-a	Indicates the standard error of the remote command is the same as standard output. No provision is made for sending arbitrary signals to the remote process.
-d	Enables socket-level debugging.
-i	Prevents reading the stdin.
-n	Prevents automatic login. With the -n flag specified, the rexec command prompts for a user name and password to use at the remote host, rather than searching for a \$HOME/.netrc file.

Parameters

Item	Description
<i>Command</i>	Specifies the command, including any flags or parameters, to be executed on the remote host.
<i>Host</i>	Specifies in alphanumeric form the name of the host where the command is to be executed.

Examples

1. To execute the **date** command on a remote host, enter:

```
rexec host1 date
```

The output from the **date** command is now displayed on the local system. In this example, the **\$HOME/.netrc** file on the local host contains a user name and password valid at the remote host.

If you do not have a valid entry in the **\$HOME/.netrc** file for the remote host, you will be prompted for your login ID and password. After you have entered the requested login information, the output from the **date** command is displayed on the local system.

2. To override the automatic login feature and execute the **date** command on a remote host, enter:

```
rexec -nhost1 date
```

Enter your name and password when prompted.

The output from the **date** command is now displayed on the local system.

3. To list the directory of another user on a remote host, enter:

```
rexec host1 ls -l /home/karen
```

The directory listing of user **karen** on remote host **host1** is displayed on the local system.

If you do not have a valid entry in the **\$HOME/.netrc** file for the remote host, you will be prompted for your login ID and password. After you have entered the requested login information, the directory listing of user **karen** on remote host **host1** is displayed on the local system.

rexecd Daemon

Purpose

Provides the server function for the **rexec** command.

Syntax

Note: The **rexecd** daemon is normally started by the **/etc/inetd.conf** or **kill -1 InetdPID** command to inform the **inetd** daemon of the changes to its configuration file.

Note: The **rexecd** daemon ignores invalid options and if the **syslog** facility is enabled, the information will be logged to the system log.

Flags

Item	Description

-s	Enables socket-level debugging.
-c	Prevents reverse name resolution. When the -c flag is not specified, the rexecd daemon will fail if the reverse name resolution of the client fails.

Service Request Protocol

When the **rexecd** daemon receives a request, it initiates the following protocol:

1. The server reads characters from the socket up to a null (\0) byte and interprets the resulting string as an ASCII number (decimal).
2. If the number received is nonzero, the **rexecd** daemon interprets it as the port number of a secondary stream to be used for standard error output. The **rexecd** daemon then creates a second connection to the specified port on the client machine.
3. The **rexecd** daemon retrieves a null-terminated user name of up to 16 characters on the initial socket.

Security

The **rexecd** daemon is a PAM-enabled application with a service name of *rexec*. System-wide configuration to use PAM for authentication is set by modifying the value of the **auth_type** attribute, in the **usw** stanza of **/etc/security/login.cfg**, to **PAM_AUTH** as the root user.

The authentication mechanisms used when PAM is enabled depend on the configuration for the **rexec** service in **/etc/pam.conf**. The **rexecd** daemon requires **/etc/pam.conf** entries for the **auth**, **account**, and **session** module types. Following is a recommended configuration in **/etc/pam.conf** for the **rexec** service:

```
#
# AIX rexec configuration
#
rexec auth      required      /usr/lib/security/pam_aix
rexec account   required      /usr/lib/security/pam_aix
rexec session   required      /usr/lib/security/pam_aix
```

rgb Command

Purpose

Creates the database used by the X-Window system server for colors.

Syntax

rgb [*DatabaseName*] [*<InputfileName*]

Description

The **rgb** command reads lines from standard input and inserts them into its database to associate color names with specific red, green, and blue (RGB) values.

The **rgb** command produces two output files: *DatabaseName.dir* and *DatabaseName.pag*. If you do not specify a database file name, the default names **rgb.dir** and **rgb.pag** are used.

Each color entry is in the form:

```
Red Green Blue Colorname
```

where the *Red*, *Green*, and *Blue* elements are integer values ranging from 0-255. The actual color is determined by how the elements are combined. Each element can range from no intensity (0) to full intensity (255). The *Colorname* parameter can be descriptive or fanciful. For example, the sequence 250 250 250 could be named white or snow. Two or more entries can share the same element numbers or names.

Parameters

Item	Description
<i>DatabaseName</i>	Specifies the database to create for the output data.
<i><InputFileName</i>	Specifies the name of the input file.

Examples

1. The following example shows a portion of an input file:

```
248 248 255    ghost white
245 245 245    white smoke
255 250 240    floral white
253 245 230    old lace
250 240 230    linen
255 218 185    peach puff
255 248 220    consilk
255 250 205    lemon chiffon
245 255 250    mint cream
240 255 255    azure
```

2. The following example generates the output files **Newcolor.dir** and **Newcolor.pag**.

```
rgb Newcolor < rgb.txt
```

where **Newcolor** is the *DatabaseName* and **rgb.txt** is the *InputFileName*.

Files

Item	Description
/usr/lib/X11/rgb.txt	The default rgb database input file.

ripquery Command

Purpose

Queries the RIP gateways.

Syntax

ripquery [-1] [-2] [-[a5] *authkey*] [-n] [-N *dest[/mask]*] [-p] [-r] [-v] [-w *time*] *gateway*...

Description

The **ripquery** command is used to request all routes known by a RIP *gateway* by sending a RIP **REQUEST** or **POLL** command. The routing information in any routing packets returned is displayed numerically and symbolically. The **ripquery** command is intended to be used as a tool for debugging *gateways*, not for network management. SNMP is the preferred protocol for network management.

Flags

Item	Description
-1	Send the query as a version 1 packet.
-2	Send the query as a version 2 packet (default).
-[a5] <i>authkey</i>	Specifies the authentication password to use for queries. If -a is specified, an authentication type of SIMPLE will be used, if -5 is specified, an authentication type of MD5 will be used, otherwise the default is an authentication type of NONE. Authentication fields in incoming packets will be displayed, but not validated.
-n	Prevents the address of the responding host from being looked up to determine the symbolic name.
-N <i>dest[/mask]</i>	Specifies that the query should be for the specified <i>dest/mask</i> instead of complete routing table. The specification of the optional mask implies a version 2 query. Up to 23 requests about specific destinations may be included in one packet.
-p	Uses the RIP POLL command to request information from the routing table. This is the default. If there is no response to the RIP POLL command, the RIP REQUEST command is tried. gated responds to a POLL command with all the routes learned via RIP.
-r	Uses the RIP REQUEST command to request information from the <i>gateway</i> 's routing table. Unlike the RIP POLL command, all <i>gateways</i> should support the RIP REQUEST . If there is no response to the RIP REQUEST command, the RIP POLL command is tried. gated responds to a REQUEST command with all the routes he announces out the specified interface.
-v	Version information about ripquery is displayed before querying the <i>gateways</i> .
-w <i>time</i>	Specifies the time in seconds to wait for the initial response from a <i>gateway</i> . The default value is 5 seconds.

rksh Command

Purpose

Invokes the restricted version of the Korn shell.

Syntax

rksh [-i] [{ + | - } { a e f h k m n p t u v x }] [-o *Option ...*] [-c *String* | -s | *File* [*Parameter*]]

Note: Preceding a flag with + (plus) rather than - (minus) turns off the flag.

Description

The `rksh` command invokes a restricted version of the Korn shell. It allows administrators to provide a controlled shell environment to the users. There is also a restricted version of `rksh` available for the enhanced Korn shell, called `rksh93`.

With a restricted shell a user cannot:

- Change the current working directory.
- Set the value of the `SHELL`, `ENV`, or `PATH` variable.
- Specify the pathname of a command that contains a `/` (slash).
- Redirect output of a command with `>` (right caret), `>|` (right caret, pipe symbol), `<>` (left caret, right caret), or `>>` (two right carets).

Flags

Item	Description
-a	Exports automatically all subsequent parameters that are defined.
-c <i>String</i>	Causes the Korn shell to read commands from the <i>String</i> variable. This flag cannot be used with the -s flag or with the <i>File[Parameter]</i> parameter.
-e	Executes the ERR trap, if set, and exits if a command has a nonzero exit status. This mode is disabled while reading profiles.
-f	Disables file name substitution.
-h	Designates each command as a tracked alias when first encountered.
-i	Indicates that the shell is interactive. An interactive shell is also indicated if shell input and output are attached to a terminal (as determined by the ioctl subroutine). In this case, the TERM environment variable is ignored (so that the kill 0 command does not kill an interactive shell) and the INTR signal is caught and ignored (so that a wait state can be interrupted). In all cases, the QUIT signal is ignored by the shell.
-k	Places all parameter assignment arguments in the environment for a command, not just those arguments that precede the command name.
-m	Runs background jobs in a separate process and prints a line upon completion. The exit status of background jobs is reported in a completion message. On systems with job control, this flag is turned on automatically for interactive shells.
-n	Reads commands and checks them for syntax errors, but does not execute them. This flag is ignored for interactive shells.

Item	Description
-o <i>Option</i>	Prints the current option settings and an error message if you do not specify an argument. You can use this flag to enable any of the following options:
allexport	Same as the -a flag.
errexit	Same as the -e flag.
bgnice	Runs all background jobs at a lesser priority. This is the default mode.
emacs	Enters an emacs-style inline editor for command entry.
gmacs	Enters a gmacs-style inline editor for command entry.
ignoreeof	Does not exit the shell when it encounters an end-of-file character. You must use the exit command, or override the flag and exit the shell by pressing the Ctrl-D key sequence more than 11 times.
keyword	Same as the -k flag.
markdirs	Appends a / (slash) to all directory names that are a result of filename substitution.
monitor	Same as the -m flag.
noclobber	Prevents redirection from truncating existing files. When you specify this option, use the redirection symbol > (right caret, pipe symbol) to truncate a file.
noexec	Same as the -n flag.
noglob	Same as the -f flag.
nolog	Prevents function definitions from being saved in the history file.
nounset	Same as the -u flag.
privileged	Same as the -p flag.
verbose	Same as the -v flag.
trackall	Same as the -h flag.
vi	Enters the insert mode of a vi-style inline editor for command entry. Entering escape character 033 puts the editor into the move mode. A return sends the line.
viraw	Processes each character as it is typed in vi mode.
xtrace	Same as the -x flag.
	You can set more than one option on a single rksh command line.

Item	Description
-p	Disables the processing of the \$HOME/.profile file when you use the shell as a login shell.
-s	Causes the rksh command to read commands from the standard input. Shell output, except for the output of the special commands, is written to file descriptor 2. This parameter cannot be used with the -c flag or with the <i>File[Parameter]</i> parameter.
-t	Exits after reading and executing one command.
-u	Treats unset parameters as errors when substituting.
-v	Prints shell input lines as they are read.
-x	Prints executed commands and their arguments.

Files

Item	Description
/usr/bin/rksh	Contains the path name to the restricted Korn shell.
/tmp/sh*	Contains temporary files that are created when a shell is opened.

rlogin Command

Purpose

Connects a local host with a remote host.

Syntax

rlogin *RemoteHost* [**-e** *Character*] [**-8**] [**-l** *User*] [**-f** | **-F**] [**-k** *realm*]

Description

The **/usr/bin/rlogin** command logs into a specified remote host and connects your local terminal to the remote host.

The remote terminal type is the same as that given in the **TERM** local environment variable. The terminal or window size is also the same, if the remote host supports them, and any changes in size are transferred. All echoing takes place at the remote host, so except for delays, the terminal connection is transparent. The Ctrl-S and Ctrl-Q key sequences stop and start the flow of information, and the input and output buffers are flushed on interrupts.

Remote Command Execution

When using the **rlogin** command, you can create a link to your path using a host name as the link name. For example:

```
ln -s /usr/bin/rsh HostName
```

Entering the host name specified by the *HostName* parameter with an argument (command) at the prompt, automatically uses the **rsh** command to remotely execute the command specified on the command line of the remote host specified by the *HostName* parameter.

Entering the host name specified by the *HostName* parameter without an argument (command) at the prompt, automatically uses the **rlogin** command to log in to the remote host specified by the *HostName* parameter.

In addition to the preceding conditions, the **rlogin** command also allows access to the remote host if the remote user account does not have a password defined. However, for security reasons, the use of a password on all user accounts is recommended.

The **rlogin** command execs (using the **exec** command) the **/usr/sbin/login** file to validate a user. This 1) allows all user and device attributes to take effect on telnet connections and 2) causes remote logins to count against the maximum number of login sessions allowable at a time (determined by the **maxlogins** attribute). Attributes are defined in the **/etc/security/user** and **/etc/security/login.cfg** files.

POSIX Line Discipline

The **rlogind** and **telnetd** daemons use POSIX line discipline to change the line discipline on the local TTY. If POSIX line discipline is not used on the local TTY, echoing other line disciplines may result in improper behavior. TCP/IP must have POSIX line discipline to function properly.

Flags

Item	Description
-8	Allows an 8-bit data path at all times. Otherwise, unless the start and stop characters on the remote host are not Ctrl-S and Ctrl-Q, the rlogin command uses a 7-bit data path and parity bits are stripped.
-e Character	Changes the escape character. Substitute the character you choose for <i>Character</i> .
-f	Causes the credentials to be forwarded. This flag will be ignored if Kerberos 5 is not the current authentication method. Authentication will fail if the current DCE credentials are not marked forwardable.
-F	Causes the credentials to be forwarded. In addition, the credentials on the remote system will be marked forwardable (allowing them to be passed to another remote system). This flag will be ignored if Kerberos 5 is not the current authentication method. Authentication will fail if the current DCE credentials are not marked forwardable.
-k realm	Allows the user to specify the realm of the remote station if it is different from the local systems realm. For these purposes, a realm is synonymous with a DCE cell. This flag will be ignored if Kerberos 5 is not the current authentication method.
-l User	Changes the remote user name to the one you specify. Otherwise, your local user name is used at the remote host.

Security

There are multiple authentication methods, each requiring different things to be set in order to allow the connection.

For Standard Authentication

The remote host allows access only if one or both of the following conditions is satisfied:

- The local host is included in the remote **\$HOME/.rhosts** file in the remote user account.

Although you can set any permissions for the **\$HOME/.rhosts** file, it is recommended that the permissions of the **.rhosts** file be set to 600 (read and write by owner only).

Note: The **AUTHSTATE** environment variable indicates the registry to which the user authenticates. For example, an LDAP user that is defined on the LDAP server has the **AUTHSTATE** set to LDAP if the user logs in to the remote system with a password. But if a user is authenticated through an entry in the **\$HOME/.rhosts** and **/etc/hosts.equiv** files, the **AUTHSTATE** environment variable for that user is set to **compat** regardless of where the user ID is defined.

For Kerberos 5 Authentication

The remote host allows access only if all of the following conditions are satisfied:

- The local user has current DCE credentials.
- The local and remote systems are configured for Kerberos 5 authentication (On some remote systems, this may not be necessary. It is necessary that a daemon is listening to the klogin port).
- The remote system accepts the DCE credentials as sufficient for access to the remote account. See the html

rlogind Daemon

Purpose

Provides the server function for the **rlogin** command.

Syntax

Note: The **rlogind** daemon is normally started by the **inetd** daemon. It can also be controlled from the command line, using SRC commands.

/usr/sbin/rlogind [**-a**] [**-c**] [**-l**] [**-n**] [**-s**]

Description

The **/usr/sbin/rlogind** daemon is the server for the **rlogin** remote login command. The server provides a remote login facility.

Changes to the **rlogind** daemon can be made using the System Management Interface Tool (SMIT) or System Resource Controller (SRC), by editing the **/etc/inetd.conf** or **/etc/services** file. Entering **rlogind** at the command line is not recommended. The **rlogind** daemon is started by default when it is uncommented in the **/etc/inetd.conf** file.

The **rlogind** daemon ignores unrecognized options and log this information through the **syslog** service if the **syslog** service is enabled in the system.

The **inetd** daemon get its information the **/etc/inetd.conf** file and the **/etc/services** file.

After changing the **/etc/inetd.conf** or **/etc/services** file, run the **refresh -s inetd** or **kill -1 InetdPID** command to inform the **inetd** daemon of the changes to its configuration file.

Service Request Protocol

When the **rlogind** daemon receives a service request, the daemon initiates the following protocol:

1. The **rlogind** daemon checks the source port number for the request. If the port number is not in the range 512-1023, the **rlogind** daemon terminates the connection.
2. The **rlogind** daemon uses the source address of the initial connection request to determine the name of the client host. If the name cannot be determined, the **rlogind** daemon uses the dotted-decimal representation of the client host address.

Error Messages

The following error messages are associated with the **rlogind** daemon:

Item	Description
Try again.	A fork command made by the server has failed.
/usr/bin/shell:	No shell. The shell specified for the shell variable cannot be started. The shell variable may also be a program.

Flags

Item	Description
-a	Disables pty speed enhancement feature.
-c	Suppresses the sanity check of a host name lookup.
-l	Prevents any authentication based on the user's \$HOME/.rhosts file. However, a root user is automatically logged in when there is a .rhosts file in root's home directory as specified by the /etc/passwd file.
-n	Disables transport-level keep-alive messages. The messages are enabled by default.
-s	Turns on socket level debugging.

Security

The **rlogind** daemon is a PAM-enabled application with a service name of *rlogin*. System-wide configuration to use PAM for authentication is set by modifying the value of the **auth_type** attribute, in the **usw** stanza of **/etc/security/login.cfg**, to **PAM_AUTH** as the root user.

The authentication mechanisms used when PAM is enabled depend on the configuration for the **rlogin** service in **/etc/pam.conf**. The **rlogind** daemon requires **/etc/pam.conf** entries for the **auth**, **account**, **password**, and **session** module types. Following is a recommended configuration in **/etc/pam.conf** for the **rlogin** service:

```
#
# AIX rlogin configuration
#
rlogin auth      sufficient /usr/lib/security/pam_rhosts_auth
rlogin auth      required   /usr/lib/security/pam_aix

rlogin account   required   /usr/lib/security/pam_aix

rlogin password  required   /usr/lib/security/pam_aix

rlogin session   required   /usr/lib/security/pam_aix
```

Examples

Note: The arguments for the **rlogind** daemon can be specified by using SMIT or by editing the **/etc/inetd.conf** file.

1. To start the **rlogind** daemon, enter the following:

```
startsrc -t rlogin
```

This command starts the **rlogind** subserver.

2. To stop the **rlogind** daemon normally, enter the following:

```
stopsrc -t rlogin
```

This command allows all pending connections to start and existing connections to complete but prevents new connections from starting.

3. To force stop the **rlogind** daemon and all **rlogind** connections, enter the following:

```
stopsrc -f -t rlogin
```

This command terminates all pending connections and existing connections immediately.

4. To display a short status report about the **rlogind** daemon, enter the following:

```
lssrc -t rlogin
```

This command returns the daemon's name, process ID, and state (active or inactive).

rm Command

Purpose

Removes (unlinks) files or directories.

Syntax

```
rm [ -f ] [ -r ] [ -R ] [ -i ] [ -e ] File ...
```

Description

The **rm** command removes the entries for the specified *File* parameter from a directory. If an entry is the last link to a file, the file is then deleted. If you do not have write permission for a file and the standard input is a terminal, you are prompted with the file name and ask to confirm that you want to delete the file. If you type a y (for yes), the file is deleted, type any other character and the file is not deleted. You do not need read or write permission for the file you want to remove. However, you must have write permission for the directory containing the file.

If the file is a symbolic link, the link is removed, but the file or directory that the symbolic link refers to remains. You do not need write permission to delete a symbolic link, if you have write permission in the directory.

If either of the files `.` (dot) or `..` (dot, dot) are specified as the base name portion of the *File* parameter, the **rm** command writes a diagnostic message to standard error and does nothing more with such parameters.

The **rm** command writes a prompt to standard error and reads a line from standard input if the **-f** flag is not specified, and either the *File* parameter does not have write permission and the standard input is a workstation, or the **-i** flag is specified. If the response is not affirmative, the **rm** command does nothing more with the current file and proceeds to the next file.

The files owned by other users cannot be removed if the sticky bit of the directory is set and the directory is not owned by the user.

Note: The **rm** command supports the `--` (dash, dash) parameter as a delimiter that indicates the end of the flags.

An attempt to remove a file or directory that has been exported for use by the NFS version 4 server will fail with a message saying that the resource is busy. The file or directory must be unexported for NFS version 4 use before it can be removed.

Flags

It e m	Description
-e	Displays a message after each file is deleted.
-f	Does not prompt before removing a write-protected file. Does not display an error message or return error status if a specified file does not exist. If both the -f and -i flags are specified, the last one specified takes affect.
-i	Prompts you before deleting each file. When you use the -i and -r flags together, the rm command also prompts before deleting directories. If both the -i and -f flags are specified, the last one specified takes affect.

Ite m	Description

-r	Permits recursive removal of directories and their contents when the <i>File</i> parameter is a directory. This flag is equivalent to the -R flag.
-R	Permits recursive removal of directories and their contents when the <i>File</i> parameter is a directory. This flag is equivalent to the -r flag.

Exit Status

This command returns the following exit values:

Ite m	Description

0	If the -f flag was not specified, all the named directory entries were removed; otherwise, all the existing named directory entries were removed.
>0	An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To delete a file, enter:

```
rm myfile
```

If there is another link to this file, then the file remains under that name, but the name `myfile` is removed. If `myfile` is the only link, the file itself is deleted.

2. To delete a file without first receiving a confirmation prompt, enter:

```
rm -f core
```

No confirmation prompt is issued before the **rm -f** command attempts to remove the file named `core`. However, an error message displays if the `core` file is write-protected and you are not the owner of the file or you do not have root authority. No error message displays when the **rm -f** command attempts to remove nonexistent files.

3. To delete files one by one, enter:

```
rm -i mydir/*
```

After each file name is displayed, enter `y` to delete the file, or press the Enter key to keep it.

4. To delete a directory tree, enter:

```
rm -ir manual
```

This command recursively removes the contents of all subdirectories of the `manual` directory, prompting you regarding the removal of each file, and then removes the `manual` directory itself, for example:

```
You: rm -ir manual
System: rm: Select files in directory manual? Enter y for yes.
You: y
System: rm: Select files in directory manual/draft1? Enter y for yes.
You: y
```

```
System: rm: Remove manual/draft1?
You: y
System: rm: Remove manual/draft1/chapter1?
You: y
System: rm: Remove manual/draft1/chapter2?
You: y
System: rm: Select files in directory manual/draft2? Enter y for yes.
You: y
System: rm: Remove manual/draft2?
You: y
System: rm: Remove manual?
You: y
```

Here, the **rm** command first asks if you want it to search the manual directory. Because the manual directory contains directories, the **rm** command next asks for permission to search manual/draft1 for files to delete, and then asks if you want it to delete the manual/draft1/chapter1 and manual/draft1/chapter2 files. The **rm** command next asks for permission to search the manual/draft2 directory. Then asks for permission to delete the manual/draft1, manual/draft2, and manual directories.

If you deny permission to remove a subdirectory (for example, manual/draft2), the **rm** command does not remove the manual directory. Instead, you see the message: rm: Directory manual not empty.

Files

Item	Description
/usr/bin/rm	Contains the rm command.

rmail Command

Purpose

Handles remote mail received through Basic Networking Utilities (BNU).

Syntax

rmail *User*

Description

The **rmail** command interprets incoming mail received through the **uucp** command. It collapses From header lines in the form generated by the **bellmail** command into a single line of the form:

```
return-path!sender
```

The **rmail** command passes the processed mail on to the **sendmail** command. The *User* parameter must specify a user recognized by the **sendmail** command.

rmail Command

Purpose

Removes one or more user-defined authorizations.

Syntax

rmail [**-R** *load_module*] [**-h**] *Name*

Description

The **rmauth** command removes the user-defined authorization identified by the *Name* parameter. The command only removes existing user-defined authorizations in the authorization database. You cannot remove system-defined authorizations with this command. If an authorization is being referenced in the privileged command database, it cannot be removed until the authorization is no longer referenced by the database.

By default, the **rmauth** command only attempts to remove the specified authorization from the authorization database. You must remove authorizations from the lowest level of a hierarchy before the higher level can be removed. If you specify a higher level authorization and low-level authorizations still exist, the command fails. To remove a hierarchy of authorizations, specify the **-h** flag. With the **-h** flag, any low-level authorization beneath the specified authorization is also removed. If any of the low-level authorizations is being referenced in the privileged command database, no authorizations are removed and the entire operation fails.

If the system is configured to use databases from multiple domains, the **rmauth** command finds the first match from the database domains in the order that was specified by the **secorder** attribute of the authorizations stanza in the **/etc/nscontrol.conf** file. Meanwhile, the **rmauth** command removes that authorization entry from the domain. If any matching authorizations from the rest of the domains exist, they are not affected. Use the **-R** flag to remove an authorization from a specific domain.

When the system is operating in enhanced role based access control (RBAC) mode, modifications made to the authorization database are not used for security considerations until the database is sent to the kernel security tables using the **setkst** command.

Flags

Item	Description
-h	Allows removal of a hierarchy of authorizations.
-R <i>load_module</i>	Specifies the loadable module to use for the authorization deletion.

Parameters

Item	Description
<i>Name</i>	Specifies the authorization to remove.

Security

The **rmauth** command is a privileged command. You must have the **aix.security.role.remove** authorization to run the command:

Item	Description
aix.security.auth.remove	Required to run the command.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Files Accessed

File	Mode
/etc/security/authorizations	rw

Examples

1. To remove the `custom.test` authorization, use the following command:

```
rmauth custom.test
```

2. To remove the custom authorization and all of its children authorizations, use the following command:

```
rmauth -h custom
```

3. To remove the `custom.test` authorization from LDAP, use the following command:

```
rmauth -h custom.test
```

rmaudrec Command

Purpose

Removes records from the audit log.

Syntax

```
rmaudrec [-a | -n node_name1[,node_name2]...] [-S subsystem_name]  
-s selection_string [-h] [-V]
```

Description

The `rmaudrec` command is used to delete records in the audit log. The audit log is a facility for recording information about the system's operation. It can include information about the normal operation of the system as well as failures and other errors. It augments the error log functionality by conveying the relationship of the error relative to other system activities. All detailed information about failures is still written to the AIX error log.

Records are created in the audit log by subsystems that have been instrumented to do that. For example, the event response subsystem runs in the background to monitor administrator-defined conditions and then invokes one or more actions when a condition becomes true. Because this subsystem runs in the background, it is difficult for the operator or administrator to understand the total set of events that occurred and the results of any actions that were taken in response to an event. Because the event response subsystem records its activity in the audit log, the administrator can easily view its activity as well as that of other subsystems. In addition, records may sometimes need to be removed explicitly, which can be done using this command.

Each record in the audit log contains named fields. Each field contains a value that provides information about the situation corresponding to the record. For example, the field named `Time` indicates the time at which the situation occurred. Each record has a set of common fields and a set of subsystem-specific fields. The common fields are present in every record in the audit log. The subsystem-specific fields vary from record to record. Their names are only significant when used with a subsystem name because they may not be unique across all subsystems. Each record is derived from a template that defines which subsystem-specific fields are present in the record and defines a format string that is used to generate a message describing the situation. The format string may use record fields as inserts. A subsystem typically has many templates.

The field names can be used as variables in a *selection string* to choose which records are deleted. The selection string is matched against each record using the referenced fields of each record to perform the match. Any records that match will be removed. The selection string is specified with the `-s` flag.

A selection string is an expression composed of field names, constants, and operators. The syntax of a selection string is very similar to an expression in the C programming language. For information on how to specify selection strings, see the *Administering RSCT* guide.

The common field names are:

Field	Description
Time	Specifies the time when the situation occurred that the record corresponds to. The value is a 64-bit integer and represents the number of microseconds since UNIX Epoch (00:00:00 GMT January 1, 1970). See the constants for specifying the time in more user-friendly formats.
Subsystem	Specifies the subsystem that generated the record. This is a string.
Category	Indicates the importance of the situation corresponding to the audit record, as determined by the subsystem that generated the record. The valid values are: 0 (informational) and 1 (error).
SequenceNumber	Specifies the unique 64-bit integer that is assigned to the record. No other record in the audit log will have the same sequence number.
TemplateId	Specifies the subsystem-dependent identifier that is assigned to records that have the same content and format string. This value is a 32-bit unsigned integer.
NodeName	Specifies the name of the node from which the record was obtained. This field name cannot be used in a selection string.

In addition to the constants in expressions, you can use the following syntax for dates and times with this command:

#mmdhmmYYYY

This format consists of a sequence of decimal characters that are interpreted according to the pattern shown. The fields in the pattern clockwise are: *mm* = month, *dd* = day, *hh* = hour, *mm* = minutes, *yyyy* = year. For example, #010523042002 corresponds to January 5, 11:04 PM, 2002. The fields can be omitted anticlockwise. If not present, the following defaults are used: year = the current year, minutes = 0, hour = 0, day = 1, and month = the current month.

#-mmdhmmYYYY

This format is similar to the previous one, but is relative to the current time and date. For example, the value #-0001 corresponds to one day ago and the value #-010001 corresponds to one month and one hour ago. Fields at the end can be omitted and are replaced by 0.

The audit records considered for deletion and matched against the selection string can be restricted to a specific subsystem by using the -S flag. If this flag is specified, the subsystem-specific field names can be used in the selection string in addition to the common field names.

The nodes from which audit log records are considered for deletion can be restricted to a set of specific nodes by using the -n flag. If this flag is specified, the search will be limited to the set of nodes listed. Otherwise, the search will be performed for all nodes defined within the current management scope as determined by the CT_MANAGEMENT_SCOPE environment variable.

It is advisable to first use the `lsaudrec` command with the same -s and -n flag values to list the records that will be deleted. This minimizes the possibility of the selection string matching more records than intended.

Flags

-a

Specifies that records from all nodes in the domain are to be removed. If both the -n and the -a flags are omitted, records from the local node only are removed.

-n node_name1[,node_name2]...

Specifies the list of nodes containing audit log records that will be examined and considered for deletion if they meet the other criteria, such as matching the specified selection string. Node group names can also be specified, which are expanded into a list of node names. If both the -n and the -a flags are omitted, records from the local node only will be deleted.

-S subsystem_name

Specifies a subsystem name. If this flag is present, only records identified by *subsystem_name* are considered for deletion. The records to be deleted can be further restricted by the *-s* flag. If the subsystem name contains any spaces, it must be enclosed in single or double quotation marks.

For backward compatibility, the subsystem name can be specified using the *-n* flag *only* if the *-a* and the *-S* flags are *not* specified.

-s selection string

Specifies a selection string. This string is evaluated against each record in the audit log. If the evaluation results in a non-zero result (TRUE), the record is removed from the audit log. If the selection string contains any spaces, it must be enclosed within single or double quotation marks. For information on how to specify selection strings, see the *RSCT: Administration Guide*.

The names of fields within the record can be used in the expression. If the *-S* flag is not specified, only the names of common fields can be used. See the **Description** for a list of the common field names and their data types. If the *-S* flag is specified, the name of any field for the specified subsystem as well as the common field names can be used.

If this flag is not specified, no records will be removed from the audit log.

-h

Writes the command's usage statement to standard output.

-V

Writes the command's verbose messages to standard error.

Parameters***field_name1 [field_name2...]***

Specifies one or more fields in the audit log records to be displayed. The order of the field names on the command line corresponds to the order in which they are displayed. If no field names are specified, Time, Subsystem, Severity, and Message are displayed by default. If the management scope is not local, NodeName is displayed as the first column by default. See the **Description** for information about these and other fields.

Security

In order to remove records from an audit log when the *-S* flag is omitted, a user must have write access to the target resource class on each node from which records are to be removed. When the *-S* flag is specified, the user must have write access to the audit log resource corresponding to the subsystem identified by the *-S* flag on each node from which records are to be removed.

Authorization is controlled by the RMC access control list (ACL) file that exists on each node.

Exit Status**0**

The command ran successfully.

1

An error occurred with RMC.

2

An error occurred with a command-line interface script.

3

An incorrect flag was entered on the command line.

4

An incorrect parameter was entered on the command line.

5

An error occurred that was based on incorrect command-line input.

Environment Variables

CT_CONTACT

Determines the system where the session with the resource monitoring and control (RMC) daemon is established. When CT_CONTACT is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If CT_CONTACT is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that can be affected by this command.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. CT_IP_AUTHENT only has meaning if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines (in conjunction with the -a and -n flags) the management scope that is used for the session with the RMC daemon. The management scope determines the set of possible target nodes where audit log records can be deleted. If the -a and -n flags are not specified, local scope is used. When either of these flags is specified, CT_MANAGEMENT_SCOPE is used to determine the management scope directly. The valid values are:

0

Specifies *local* scope.

1

Specifies *local* scope.

2

Specifies *peer domain* scope.

3

Specifies *management domain* scope.

If this environment variable is *not* set, *local* scope is used.

Implementation Specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) filesset for AIX.

Standard Output

When the -h flag is specified, this command's usage statement is written to standard output.

Standard Error

If the -V flag is specified and the command completes successfully, a message indicating the number of records that were deleted will be written to standard error.

Examples

1. To remove all records from the audit log on every node in the management scope defined by the CT_MANAGEMENT_SCOPE environment variable, enter:

```
rmaudrec -s "Time > 0"
```

or

```
rmaudrec -s "SequenceNumber >= 0"
```

2. To remove all records more than a week old on every node in the management scope defined by the CT_MANAGEMENT_SCOPE environment variable, enter:

```
rmaudrec -s "Time < #-0007"
```

3. To remove all records that are more than a day old and created by the abc subsystem on nodes mynode and yournode, enter:

```
rmaudrec -S abc -s "Time < #-0001" -n mynode,yournode
```

Location

/opt/rsct/bin/rmaudrec

rmC2admin Command

Purpose

Remove the configuration files for a distributed C2 System host.

Syntax

rmC2admin [**-m**]

Description

The **rmC2admin** command replaces the distributed C2 System symbolic links with the actual files. The directory **/etc/data.shared** will be removed. When the **-m** flag is used, the **hd10sec** file system and **/etc/data.master** directory will be removed as well. This option should only be used after all other hosts in the C2 System have replaced their administrative host with another system or removed the C2 configuration files as well.

The entries for the system initialization scripts in **/etc/inittab** will also be removed, and rebooting this system will result in the system not being configured for C2 mode.

Executing this command in multi-user mode will result in the user definitions from the C2 System being retained. Executing this command in single-user mode will result in the user definitions from the C2 System being removed and the root user being the only valid user ID.

The system should be rebooted immediately after executing this command so that the changes may take effect.

Flags

Item	Description
-m	The host was configured as the administrative master

Exit Status

- 0** The C2 System administrative host information has been successfully removed.
- 1** The system was not configured to operate in C2 mode.
- 2** The system was not installed with the C2 option.
- 3** An error occurred removing the C2 System administrative host information.

4

An invalid command line option was used.

Files

Item	Description
<code>/usr/sbin/rmC2admin</code>	Contains the rmC2admin command.

rmCCadmin Command

Purpose

Remove the configuration files for a distributed Common Criteria enabled System host.

Syntax

rmCCadmin [**-m**]

Description

The **rmCCadmin** command replaces the distributed Common Criteria enabled System symbolic links with the actual files. The directory **/etc/data.shared** will be removed. When the **-m** flag is used, the **hd10sec** file system and **/etc/data.master** directory will be removed as well. This option should only be used after all other hosts in the Common Criteria enabled System have replaced their administrative host with another system or removed the Common Criteria enabled configuration files as well.

The entries for the system initialization scripts in **/etc/inittab** will also be removed, and rebooting this system will result in the system not being configured for Common Criteria enabled mode.

Executing this command in multi-user mode will result in the user definitions from the Common Criteria enabled System being retained. Executing this command in single-user mode will result in the user definitions from the Common Criteria enabled System being removed and the root user being the only valid user ID.

The system should be rebooted immediately after executing this command so that the changes may take effect.

Flags

Item	Description
-m	The host was configured as the administrative master

Exit Status

0

The Common Criteria enabled System administrative host information has been successfully removed.

1

The system was not configured to operate in Common Criteria enabled mode.

2

The system was not installed with the Common Criteria enabled option.

3

An error occurred removing the Common Criteria enabled System administrative host information.

4

An invalid command line option was used.

Files

Item

`/usr/sbin/rmCCadmin`

Description

Contains the **rmCCadmin** command.

rmccli information file

Purpose

Provides general information about resource monitoring and control (RMC) and related commands.

Description

The general information about RMC and related commands, including data types, terminology, and references to related information follows.

Command structure and use

The RMC commands might be grouped into categories that represent the different operations that can be run on resource classes and resources:

- Creating and removing resources: **mkrsrc**, **rmrsrc**
- Modifying resources: **chrsrc**, **refrsrc**
- Viewing definitions and data: **lsrsrc**, **lsrsrcdef**
- Viewing actions: **lsactdef**
- Running actions: **runact**

The RMC commands can be run directly from the command line or called by user-written scripts. In addition, the RMC commands are used as the basis for higher-level commands, such as the event response resource manager (ERRM) commands.

Data display information

The flags that control the display function for the RMC CLI routines, in order of precedence are:

1. **-l** for long display. This flag is the default display format.

For example, the command:

```
lsrsrc -s 'Name == "c175n05"' IBM.Foo Name NodeList SD Binary RH Int32Array
```

produces the following output:

```
Persistent Attributes for Resource: IBM.Foo
resource 1:
  Name       = "c175n05"
  NodeList   = {1}
  SD         = ["testing 1 2 3",1,{0,1,2}]
  Binary     = "0xaabbcc00 0xeeff"
  RH         = "0x0000 0x0000 0x00000000 0x00000000 0x00000000 0x00000000"
  Int32Array = {1,5,-10,1000000}
```

2. **-t** for tabular display.

For example, the command:

```
lsrsrc -s 'Name ?= "Page"' -t IBM.Condition Name EventExpression
```

produces the following output:

```
Persistent Attributes for Resource: IBM.Condition

Name                               EventExpression
"Page space out rate"              "VMPgSpOutRate > 500"
"Page fault rate"                  "VMPgFaultRate > 500"
"Page out rate"                    "VMPgOutRate > 500"
```

```
"Page in rate"      "VMPgInRate > 500"
"Page space in rate" "VMPgSpInRate > 500"
```

- x for suppressing headers when printing.
- d for colon (:) delimited display.

For example, the command:

```
lsrsrc -xd -s 'Name == "c175n05"' IBM.Foo Name Int32 UInt32Array SD Binary
```

produces the following output:

```
c175n05:-100:{}:["hel lo1",1,{0,1,2}]:"0xaabbcc00 0xeeff":
```

Note the use of the `-x` flag along with the `-d` flag.

- Ddelimiter* for string-delimited display.

For example, the command:

```
lsrsrc -xD:: -s 'Name == "c175n05"' IBM.Foo Name Int32 UInt32Array SD Binary
```

produces the following output:

```
c175n05::-100::{}::["hel  lo1",1,{0,1,2}]::"0xaabbcc00 0xeeff"::
```

Note the use of the `-x` flag along with the `-DDelimiter` flag.

en output of any list command **lsrsrc lsrsrcdef** is displayed in the tabular output format, the
nting column width might be truncated. If more characters need to be displayed (as in the case of
ngs) use the **-l** flag to display the entire field.

Data input formatting

Binary data for attributes of binary type can be entered in the following formats:

- "0xxxxxxxxx 0x xxxxxxxx 0x xxxxx..."
- "0xxxxxxxxxxxxxxxxxxxxxxxxxxx..."
- 0x xxxxxxxxxxxxxxxxxxxxxxxxxx...

Integer data for attributes of one of the integer types can be entered as:

- A decimal constant that begins with a non-zero digit (Int32=45, for example)
- An octal constant that begins with a prefix of 0, which is optionally followed by a combination of decimal numbers in the range 0 to 7 (Int32=055, for example)
- A hexadecimal constant that begins with a prefix of 0x or 0X followed a combination of decimal numbers in the range a to f and A to F (Int32=0x2d, for example)

Be careful when you specify strings as input data. Strings that contain:

- No white space or non-alphanumeric characters can be entered as input without enclosing quotation marks
- White space or other alphanumeric characters must be enclosed in quotation marks
- Single quotation marks (') must be enclosed by double quotation marks ("), as shown in this example: "this is a string with 'single quotation marks'"

Selection strings must be enclosed in double quotation marks, unless the selection string itself contains double quotation marks, in which case the selection string must be enclosed in single quotation marks. For information about how to specify selection strings, see the *Administering RSCT* Guide.

- Sample selection string input: "NodeNumber == 1"
- Selection string input where double quotation marks are part of the selection string: 'Name == "c175n05" '

Structured data (SD) types must be enclosed in square brackets: `[hello,1,{2,4,6,8}]`

When structured data (SD) is supplied as command-line input to the RMC commands, enclose the SD in single quotation marks: `SD= '[hello,1,{2,4,6,8}]'`

Arrays of any type must be enclosed in braces `{}`:

- Array of integers: `{-4, -3, -2, -1, 0, 1, 2, 3, 4}`
- Array of strings: `{abc, "do re mi", 123}`
- Array of structured data: `{[hello,1,{0,1,2,3}], [hello2,2,{2,4,6,8}]}`

Arrays of any type with more than one element must be enclosed in quotation marks. For example:

- **mkrsrc** IBM.Foo Name=testing NodeList={1} Uint32Array='{1,2,3}'
- **mkrsrc** IBM.Foo Name=testing NodeList='{1}' Uint32_array='{1,2,3}'

Arrays of strings and arrays of structured data must always be enclosed in quotation marks.

When arrays of structured data or arrays that contain strings enclosed in quotation marks are supplied as command-line input to the RMC commands, enclose the entire array in single quotation marks:

- Array of strings: **mkrsrc** IBM.Foo Name="c175n05" NodeList={1} StringArray='{ "a string", "a different string" }'
- Array of structured data: **mkrsrc** IBM.Foo Name="c175n05" NodeList={1} SDArray='{ ["string 1",1,{1,1}], ["string 2",2,{1,2,3}] }'

For more examples, see the `resource_data_input`.

Data output formatting

String data is always displayed in either double or single quotation marks as:

- A description attribute that equals the string "This is a string that contains white space" is displayed in the long format as:

```
Description = "This is a string that contains white space"
```

- A description attribute value that equals an empty string "" is displayed in long format as:

```
Description = ""
```

- A description attribute value that equals a string that contains a new-line character at the end of the string is displayed in long format as:

```
Description = "This string ends with a new-line character..."
```

- A selection string that contains double quotation marks is displayed in long format as:

```
SelectionString = 'Name == "c175n05"'
```

- A name attribute value that equals the string "c175n05" is displayed in long format as:

```
Name = "c175n05"
```

Binary data is displayed as follows:

```
"0x nnnnnnnnn 0x nnnnnnnnn 0x nnnnnnnnn 0x nnnnnnnnn"
```

Naming conventions

The following variable names are used throughout the RMC command man pages:

Variable	Description
<i>attr</i>	The name of a resource class or a resource attribute
<i>resource_class</i>	The name of a resource class

Node groups

If Cluster Systems Management (CSM) is installed on your system, you can use CSM defined node groups as node name values to refer to more than one node. For information about working with CSM node groups and by using the CSM **nodegrp** command, see the *CSM: Administration Guide* and the *CSM: Command and Technical Reference*.

Terminology

attribute

Attributes are either persistent or dynamic. A resource class is defined by a set of persistent and dynamic attributes. A resource is also defined by a set of persistent and dynamic attributes. Persistent attributes define the configuration of the resource class and resource. Dynamic attributes define a state or a performance-related aspect of the resource class and resource. In the same resource class or resource, an attribute name can be specified as either persistent or dynamic, but not both.

resource

An entity in the system that provides a set of services. Examples of hardware entities are processors, disk drives, memory, and adapters. Examples of software entities are database applications, processes, and file systems. Each resource in the system has one or more attributes that define the state of the resource.

resource class

A broad category of system resource, for example: node, file system, adapter. Each resource class has a container that holds the functions, information, dynamic attributes, and conditions that apply to that resource class. For example, the `"/tmp space used"` condition applies to a file system resource class.

resource manager

A process that maps resource and resource-class abstractions into calls and commands for one or more specific types of resources. A resource manager can be a stand-alone daemon, or it can be integrated into an application or a subsystem directly.

To see all of the resource classes that are defined in the system, run the **lsrsrc** command without any flags or parameters. To see all of the resources that are defined in the system for the `IBM.FileSystem` resource class, enter:

```
lsrsrc IBM.FileSystem
```

selection string

Must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks, for example:

```
-s 'Name == "testing"'
-s 'Name != "test"'
```

Only persistent attributes can be listed in a selection string.

Flags

-h

Writes the command usage statement to standard output.

-T

Writes the command trace messages to standard error. For your software service organization use only.

-V

Writes the command verbose messages (if there are any available) to standard output.

All RMC commands include a `-T` flag and a `-V` flag. Use the `-T` flag only when your software service organization instructs you to turn on tracing. Trace messages are not translated. Use the `-V` flag, which indicates "verbose" mode, to see more information about the command. Verbose messages (if there are

any available) are contained in message catalogs and are translated based on the locale in which you are running and other criteria.

Environment variables

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the resource monitoring and control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are on the system to which the connection is established.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. The CT_IP_AUTHENT environment variable is valid, if the CT_CONTACT environment variable is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon to monitor and control the resources and resource classes. The management scope determines the set of possible target nodes where the resources and resource classes can be monitored and controlled. The valid values are:

- 0**
Specifies *local* scope.
- 1**
Specifies *local* scope.
- 2**
Specifies *peer domain* scope.
- 3**
Specifies *management domain* scope.

If this environment variable is *not* set, *local* scope is used.

Standard output

When the -h flag is specified, this command usage statement is written to standard output. When the -V flag is specified, these command verbose messages (if there are any available) are written to standard output.

Standard error

All trace messages are written to standard error.

Exit status

- 0**
The command ran successfully.
- 1**
An error occurred with RMC.
- 2**
An error occurred with the command-line interface (CLI) script.
- 3**
An incorrect flag was specified on the command line.
- 4**
An incorrect parameter was specified on the command line.

5

An error occurred with RMC that was based on incorrect command-line input.

6

No resources were found that match the specified selection string.

Security

Permissions are specified in the access control list (ACL) file on the contacted system.

Implementation specifics

This information is part of the `rsct.core.rmc` fileset for AIX and `rsct.core-3.1.0.0-0.platform.rpm` package for Linux, Solaris, and Windows, where *platform* is `i386`, `ppc`, `ppc64`, `s390`, or `x86_64`.

Location

/opt/rsct/man/rmccli

/opt/rsct/man/rmccli.7 - For Linux platform.

rmcctl Command

Purpose

Manages the resource monitoring and control (RMC) subsystem.

Syntax

```
rmcctl { -a | -A | -b | -B | -d | -k | -K | -m {R | E | D} | -M {R | E | D} | -p | -P | -q | -Q | -s |  
-t n | -T | -u n | -U | -v n | -V | -w n | -W | -x | -X | -z | -h }
```

Description

The `rmcctl` command controls the operation of the resource monitoring and control (RMC) subsystem. The subsystem is under the control of the system resource controller (SRC) with a subsystem name of `ctrmc` and a subsystem group name of `rsct`. The RMC subsystem definition is added to the subsystem object class and then started when Reliable Scalable Cluster Technology (RSCT) is installed. In addition, an entry is made in the `/etc/inittab` file so that the RMC subsystem is started automatically when the system is started.

Note: While the RMC subsystem can be stopped and started by using the `stopsrc` and `startsrc` commands, you can use the `rmcctl` command to perform these functions.

Flags

-a

Adds the RMC subsystem to the subsystem object class and places an entry at the end of the `/etc/inittab` file.

-A

Adds and starts the RMC subsystem.

-b

Sets the idle timeout for the RMC API client session to *n* seconds. If the RMC daemon finds no activity in the session for the last *n* seconds, it is closed.

-B

Sets the idle timeout for the RMC API client session to a default value of 0 seconds (that is it is disabled).

- d**
Deletes the RMC subsystem from the subsystem object class and removes the RMC entry from the `/etc/inittab` file.
- k**
Stops the RMC subsystem.
- K**
Stops the RMC subsystem and all resource managers.
- m**
Specifies the RMC subsystem client message policy. This policy applies to messages sent between the RMC subsystem and any command that is listed in the *RSCT: Technical Reference*, when the command is run on a different node than the RMC subsystem (in other words, the `CT_CONTACT` environment variable is set). These messages are sent by using TCP/IP.

This flag is supported on RSCT version 2.3.1.0 or later. The "Enabled" policy must be used if the commands are from an earlier version of RSCT.

R
Indicates that the client message policy is "Required". "Required" means that the connection remains open only if message authentication can (and will) be used.

E
Indicates that the client message policy is "Enabled". "Enabled" is the default; message authentication is used if both sides of the connection support it.

D
Indicates that the client message policy is "Disabled". "Disabled" means that message authentication is not used.
- M**
Specifies the RMC subsystem daemon message policy. This policy applies to messages sent between the RMC subsystem daemons within a management domain cluster. These messages are sent by using the User Datagram Protocol (UDP).

This flag is supported on RSCT release 2.4.1.0 or later. When specified, the indicated message policy takes effect the next time the RMC subsystem is started.

R
Indicates that the daemon message policy is "Required". "Required" means that two daemons communicate only if message authentication can (and will) be used.

E
Indicates that the daemon message policy is "Enabled". "Enabled" is the default; message authentication is used if the sending and receiving daemons support it.

D
Indicates that the daemon message policy is "Disabled". "Disabled" means that message authentication is not used. Disabling message authentication can result in the loss of function if all of the nodes in the cluster are not configured the same.
- p**
Enables remote client connections.
- P**
Disables remote client connections.
- q**
Enables remote client connections the next time the RMC subsystem is started.
- Q**
Disables remote client connections the next time the RMC subsystem is started.
- s**
Starts the RMC subsystem.

-t *n*

Sets the client message timeout value to *n* seconds. This timeout value must include the following actions:

- Receiving the first message of the start session protocol after the RMC subsystem accepts a client connection.
- Receiving the complete client message by the RMC subsystem, after the initial message is received

If either of these time limits is exceeded, the client session is closed. The minimum acceptable value is 10; the maximum is 86400.

When specified, this value takes effect the next time the RMC subsystem is started.

-T

Sets the client message timeout to the default value of 10 seconds.

When specified, this value takes effect the next time the RMC subsystem is started.

-u *n*

Sets the start session timeout value to *n* seconds. Within this amount of time, the start session processing must complete for a new client session; otherwise, the session is closed. The minimum acceptable value is 60; the maximum is 86400.

When specified, this value takes effect the next time the RMC subsystem is started.

-U

Sets the start session timeout value to the default value of 300 seconds.

When specified, this value takes effect the next time the RMC subsystem is started.

-v *n*

Sets the first command timeout value to *n* seconds. If a first command timer is set when a client session is established with the RMC subsystem, the first command must arrive within the specified number of seconds after the start session processing completes; otherwise, the session is closed. The minimum acceptable value is 10; the maximum is 86400.

When specified, this value takes effect the next time the RMC subsystem is started.

-V

Sets the first command timeout value to the default value of 10 seconds.

When specified, this value takes effect the next time the RMC subsystem is started.

-w *n*

Sets the first command threshold value to *n* client sessions. Once the number of client sessions exceeds this value, the RMC subsystem enables a first command timer on each new, unauthenticated session. If the threshold is set to 0, the first command timeout function is disabled. The maximum value is 150.

When specified, this value takes effect the next time the RMC subsystem is started.

-W

Sets the first command threshold value to the default value of 150 client sessions.

When specified, this value takes effect the next time the RMC subsystem is started.

-x

Enables first command timeouts for non-root authenticated client sessions and for unauthenticated client sessions.

When specified, this value takes effect the next time the RMC subsystem is started.

-X

Disables first command timeouts for non-root authenticated sessions.

When specified, this value takes effect the next time the RMC subsystem is started.

-z

Stops the RMC subsystem and all resource managers, but the command does not return until the RMC subsystem and the resource managers are stopped.

-h

Writes the command's usage statement to standard output.

Security

Privilege control: only the root user must run (x) access to this command.

Exit Status

0

The command is successful.

1

The command was not successful.

Standard Output

When the -h flag is specified, this command's usage statement is written to standard output.

Examples

1. To add the RMC subsystem, enter:

```
rmcctrl -a
```

2. To start the RMC subsystem, enter:

```
rmcctrl -s
```

3. To stop the RMC subsystem, enter:

```
rmcctrl -k
```

4. To delete the RMC subsystem, enter:

```
rmcctrl -d
```

Location

/opt/rsct/bin/rmcctrl

rmcdomainstatus Command

Purpose

Displays the status of the node in management domain and peer domain.

Syntax

```
rmcdomainstatus -s ctrmc [-a ip|IP]
```

Description

When you run the **rmcdomainstatus** command in a node, the output displays the node status in the management domain and peer domain that contains the node. If the output is not displayed, the node is not a member of any peer domain or management domain.

The output of the **rmcdomainstatus** command is displayed in the following format:

```
Domain status
<Token 1 of node status> <Token 2 of node status> <Node ID> <Internal node number> <Node name
| IP address | IP address of the specified MCP> <PD_name>/<PD_status> (n)
```

The following information fields are displayed in the **rmcdomainstatus** command output:

Domain status

Displays the current state of domain. The domain status can be displayed in the following ways:

Peer Domain Status

Displays the current state of peer domain.

Management Domain Status: Managed Nodes

Displays the current state of all the managed nodes that are managed by the node in the management domain.

Management Domain Status: Management Control Points (MCP)

Displays the current state of all the Management Control Points (MCPs) that are managing the node in the management domain.

Note: The output might contain more than one section depending on the current state of the node. That is, if the node is a member of both peer domain and management domain, the output contains two separate sections of information.

Token 1 of node status

Specifies the node status that indicates one of the following conditions:

S

Indicates that text in the output is for current node in the peer domain.

I

In a management domain, this value indicates that the node is in the Up state, which is determined by the Resource Monitoring and Control (RMC) heartbeat mechanism. In a peer domain, this value indicates that the RMC daemon in the specified node is a member of the `rmc_peers` Group Services group and the node is online in the peer domain.

i

In a management domain, this value indicates that the node is in the Pending Up state. Communication is established between two RMC daemons but the initial handshake is not completed.

Note: The `i` token is displayed only for management domains.

O

In a management domain, this value indicates that the node is in the Down state, which is determined by the RMC heartbeat mechanism. In a peer domain, this value indicates that the RMC daemon in the specified node is no longer a member of the `rmc_peers` Group Services group.

X

In a management domain, this value indicates that a communication problem is discovered, and the RMC daemon has suspended communication with the RMC daemon that is in the specified node.

Z

Indicates that the RMC daemon has suspended communication with the RMC daemon that is in the specified node because the Up or Down state of the node is changing quickly.

Token 2 of node status

Specifies the node status that indicates one of the following conditions:

S

Indicates that text in the output is for current node in the peer domain.

A

Indicates that the messages are not queued for the specified node.

a

Indicates the same meaning as the A value, except that the specified node is running a version of the RMC daemon that is at a low level than the local RMC daemon.

R

Indicates that the messages are queued for the specified node.

r

Indicates the same meaning as the R value, except that the specified node is running a version of the RMC daemon that is at a low level than the local RMC daemon.

Node ID

Specifies the 64-bit node ID that is created when RSCT is installed on the node.

Internal node number

Specifies the internal node number that is used by the RMC daemon.

Node name or IP address

Specifies the name of the node that is identified by the RMC subsystem.

Note: This value is displayed only if the node is a member of a peer domain or a management domain.

IP address of the specified MCP

Specifies the first configured IP address of the specified MCP.

Note: This value is displayed only if the node is an MCP.

PD_name/PD_status (n)

Specifies the peer domain name and peer domain status as received from the managed node when the **-a** flag is not used.

Note: This value is displayed only if the node is an MCP.

The *PD_name* attribute is the name of the peer domain of which the managed node is an online member. The *PD_status* attribute is the status of the peer domain.

If the managed node is offline, the *PD_name/PD_status* attributes are set as *!/-*, and the *(n)* attribute is not present. If the peer domain status is received from the managed node, the *PD_name* attribute is set as *+*. The *n* attribute is the number of online nodes in the peer domain of which the specified managed node is a member.

Flags**-s ctrmc**

Specifies the RSCT daemon name. For RMC, the RSCT daemon name is *ctrmc*.

-a IP|ip

Lists the IP addresses that are configured on the node. The valid values that can be specified with the **-a** flag are as follows:

IP

Lists all the configured and harvested IP addresses.

ip

Lists the IP addresses that are configured in the *ctrmc.srctb1* file (for peer domain) and in the *ctrmc.mntb1* or *ctrmc.mcptb1* file (for management domain).

Implementation specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) fileset for AIX, Linux, and other operating systems.

Location

/opt/rsct/bin/rmcdomainstatus

Examples

1. To check the node status in the peer domain and management domain, run the following command:

```
rmcdomainstatus -s ctrmc
```

If the output is not displayed, the node is not a member of a peer domain or a management domain.

If the node is a member of a peer domain, an output that is similar to the following example is displayed:

```
Peer Domain Status
I A 0x09898b3065189db6 0002 test1.ppd.pok.ibm.com
S S 0x07e7287425d0becd 0001 test2.ppd.pok.ibm.com
```

If the node is an MCP, an output that is similar to the following example is displayed:

```
Management Domain Status: Managed Nodes
I a 0xbf1fb04e5b7d0b06 0001 test1 !/+
I a 0x3a75dd6c235c428e 0002 test2 masMMtest/+ (1)
I A 0x07e7287425d0becd 0003 test3 masfive/+ (2)
I A 0x09898b3065189db6 0004 test4 masfive/+(2)
```

If the node is a managed node, an output that is similar to the following example is displayed:

```
Management Domain Status: Management Control Points
I A 0xef889c809d9617c7 0001 9.xx.xx.xxx
```

2. To display the configured and harvested IP addresses in the current node status, run the following command:

```
rmcdomainstatus -s ctrmc -a IP
```

An output that is similar to the following example is displayed:

```
Peer Domain Status
I A 0x4313b01f7aae13d9 0002 myrsct1.in.ibm.com
S S 0xa15313e0cc675d54 0001 myrsct2.in.ibm.com

Management Domain Status: Management Control Points
I A 0x128a32b77a5d91cb 0001 10.xx.xx.xx (C)
```

rmcifscred Command

Purpose

Removes the CIFS credentials stored in the /etc/cifs_fs/cifscred file for the specified server and user entry.

Syntax

```
rmcifscred -h RemoteHost -u user
```

Description

The `rmcifscred` command takes a server and user name as input. If this input has credentials listed in /etc/cifs_fs/cifscred, the credentials are removed. Subsequent mounting to the specified server by the specified user requires manually inputting the password.

Flags

Item	Description
-h <i>RemoteHost</i>	Specifies the name of the remote host (CIFS server). This can be provided as a host name, an IP address, or as a fully qualified domain name.
-u <i>user</i>	Specifies the user name whose credentials for the specified server are to be removed from the <code>cifscred</code> file.

Exit Status

Item	Description
0	The command completed successfully.
>0	An error occurred.

Examples

1. To remove the credentials stored in `/etc/cifs_fs/cifscred` for `user1` to mount on `server1`, enter:

```
rmcifscred -h server1 -u user1
```

Location

`/usr/sbin/rmcifscred`

Files

Item	Description
<code>/etc/cifs_fs/cifscred</code>	Stores the CIFS credentials.

rmcifsmnt Command

Purpose

Removes a CIFS mount from the `/etc/filesystems` file and unmounts the entry if it is mounted.

Syntax

```
rmcifsmnt -f MountPoint [-B | -N]
```

Description

The `rmcifsmnt` command removes a CIFS entry from `/etc/filesystems`. If the entry is mounted, the `rmcifsmnt` command then unmounts it.

Flags

Item	Description
-B	Removes the corresponding entry from the <code>/etc/filesystems</code> file, and unmounts the file system. This is the default.

Item

-f *MountPoint*
-N

Description

Specifies the path name of the CIFS mount.
Unmounts the file system, but does not remove the entry from the `/etc/filesystems` file.

Exit Status**Item**

0
>0

Description

The command completed successfully.
An error occurred.

Examples

1. To remove the CIFS mount that is mounted over `/mnt` and unmount it, enter:

```
rmcifsmt -f /mnt
```

Location

`/usr/sbin/rmcifsmt`

Files**Item**

`/etc/filesystems`

Description

Stores the CIFS entry.

rmclass Command

Purpose

Remove a Workload Management class.

Syntax

```
rmclass [ -d Config_Dir ] [ -S SuperClass ] Name
```

Description

The **rmclass** command removes the superclass or the subclass identified by the *Name* parameter from the class definition file, the class limits file and the class shares file. The class must already exist. The predefined **Default**, **System**, and **Shared** classes cannot be removed.

In addition, when removing a superclass **Super**, the directory `/etc/wlm/Config_Dir/Super` and all the WLM property files it contains (if they exist) are removed. Removing a superclass fails if any user created subclass still exists (subclass other than **Default** and **Shared**).

Note: Only root can remove a superclass. Only root or authorized users whose user ID or group ID matches the user name or group name specified in the attributes **adminuser** and **admingroup** of a superclass can remove a subclass of this superclass.

Normally, **rmclass** deletes the class and its attributes in the relevant WLM property files, and the modifications are applied to the in-core class definitions (active classes) only after an update of WLM using the **wlmcntrl** command.

If an empty string is passed as the configuration name (*Config_dir*) with the **-d** flag, the class is deleted only in the WLM in-core data structures, and no property file is updated. So, if the class is still defined in a WLM configuration, it is recreated after an update or restart of WLM. This flag should be mainly used to remove classes dynamically created in the in-core WLM data structures only by applications using the WLM API, for example, to do some cleanup after application failure.

Note: This command cannot apply to a set of time-based configurations (do not specify a set with the **-d** flag). If the current configuration is a set, the **-d** flag must be given to indicate which regular configuration the command should apply to.

Flags

Item	Description
-d <i>Config_Dir</i>	Uses /etc/wlm/Config_dir as alternate directory for the properties files. If this flag is not used, the configuration files in the directory pointed to by /etc/wlm/current are used. If an empty string is passed as the configuration name (-d "") the class is deleted only in the WLM in-core data structures and no configuration file is modified.
-S <i>SuperClass</i>	Specifies the name of the superclass when removing a subclass. There are two ways of specifying the subclass Sub of superclass Super : 1. Specify the full name of the subclass as Super.Sub and do not use -S. 2. Specify the -S flag to give the superclass name and use the short name for the subclass:

```
rmclass options -S Super Sub
```

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Files

Item	Description
classes	Contains the names and definitions of the classes.
limits	Contains the resource limits.
shares	Contains the resource shares.

rmcluster Command

Purpose

Removes an existing cluster or site configuration.

Syntax

```
rmcluster [-n clustername] [-S sitename][-v]
```

Description

The **rmcluster** command removes the cluster configuration or one of the sites in the cluster. The repository disk and the shared disks of the SAN Volume Controller (SVC) that are associated with the entity that must be removed are released.

When a site is removed from the cluster, the repository and the shared disks that are used by the site are released. Releasing the disks does not cause the site to be removed. When a cluster is removed, all the repository and shared disks are released.

Note: A site cannot remove itself. Sites can only be removed from a node in a different site.

Flags

Item	Description
-n <i>clustername</i>	Specifies the name of the cluster to be removed.
-S <i>sitename</i>	Specifies the name of the site to be removed.
-v	Specifies the verbose mode.

Examples

1. To remove the cluster configuration, enter the following command:

```
rmcluster -n mycluster
```

2. To remove a site named *mysite* from the cluster, enter the following command on a node in a different site:

```
rmcluster -S mysite
```

rmcomg Command

Purpose

Removes a communication group that has already been defined from a peer domain.

Syntax

```
rmcomg [-q] [-h] [-TV] communication_group
```

Description

The **rmcomg** command removes the definition of the existing communication group with the name specified by the *communication_group* parameter for the online peer domain. The communication group is used to define heartbeat rings for use by topology services and to define the tunables for each heartbeat ring. The communication group determines which devices are used for heartbeating in the peer domain.

The **rmcomg** command must be run on a node that is currently online in the peer domain where the communication group is defined. More than half of the nodes must be online to remove a communication group from the domain.

The communication group must not be referred to by an interface resource. Use the **chcomg** command to remove references made by interface resources to a communication group.

Flags

- q**
Specifies quiet mode. The command does not return an error if the communication group does not exist.
- h**
Writes the command's usage statement to standard output.
- T**
Writes the command's trace messages to standard error. For your software service organization's use only.
- V**
Writes the command's verbose messages to standard output.

Parameters

communication_group

Specifies the name of the defined communication group that is to be removed from the peer domain.

Security

The user of the `rmcomg` command needs write permission for the `IBM.CommunicationGroup` resource class. By default, `root` on any node in the peer domain has read and write access to this resource class through the configuration resource manager.

Exit Status

- 0**
The command ran successfully.
- 1**
An error occurred with RMC.
- 2**
An error occurred with a command-line interface script.
- 3**
An incorrect flag was entered on the command line.
- 4**
An incorrect parameter was entered on the command line.
- 5**
An error occurred that was based on incorrect command-line input.
- 6**
The communication group does not exist.

Environment Variables

CT_CONTACT

Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When `CT_CONTACT` is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If `CT_CONTACT` is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

CT_IP_AUTHENT

When the `CT_IP_AUTHENT` environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the `CT_CONTACT` environment variable is set. `CT_IP_AUTHENT` only has meaning if `CT_CONTACT` is set to an IP address; it does not rely on the domain name system (DNS) service.

Restrictions

This command must be run on a node that is defined and online to the peer domain where the communication group is to be removed.

Implementation Specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) fileset for AIX.

Standard Input

When the `-f` "-" or `-F` "-" flag is specified, this command reads one or more node names from standard input.

Standard Output

When the `-h` flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

In this example, nodeA is defined and online to App1Domain. To remove the communication group definition ComGrp1 for the peer domain App1Domain, run this command on nodeA:

```
rmcomg ComGrp1
```

Location

`/opt/rsct/bin/rmcomg`

rmcondition Command

Purpose

Removes a condition.

Syntax

```
rmcondition [-f] [-q] [-h] [-TV] condition[:node_name]
```

Description

The `rmcondition` command removes the condition specified by the *condition* parameter. The condition must already exist to be removed. When the condition must be removed even if it has linked responses, use the `-f` flag to force the condition and the links with the responses to be removed. If the `-f` flag is not specified and links with responses exist, the condition is not removed. This command does not remove responses.

If a particular condition is needed for system software to work properly, it may be locked. A locked condition cannot be modified or removed until it is unlocked. If the condition you specify on the `rmcondition` command is locked, it will not be removed; instead an error will be generated informing you that the condition is locked. To unlock a condition, you can use the `-U` flag of the `chcondition` command. However, since a condition is typically locked because it is essential for system software to work properly, you should exercise caution before unlocking it.

Flags

- f**
Forces the condition to be removed even if it is linked to responses. The links with the responses are removed as well as the condition, but the responses are not removed.
- q**
Does not return an error when *condition* does not exist.
- h**
Writes the command's usage statement to standard output.
- T**
Writes the command's trace messages to standard error. For your software service organization's use only.
- V**
Writes the command's verbose messages to standard output.

Parameters

condition

Specifies the name of a condition to be removed.

node_name

Specifies the node where the condition is defined. If *node_name* is not specified, the local node is used. *node_name* is a node within the scope determined by the CT_MANAGEMENT_SCOPE environment variable.

Security

The user needs write permission for the IBM.Condition resource class to run `rmcondition`. Permissions are specified in the access control list (ACL) file on the contacted system.

Exit Status

- 0**
The command ran successfully.
- 1**
An error occurred with RMC.
- 2**
An error occurred with a command-line interface script.
- 3**
An incorrect flag was entered on the command line.
- 4**
An incorrect parameter was entered on the command line.
- 5**
An error occurred that was based on incorrect command-line input.

Environment Variables

CT_CONTACT

Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When CT_CONTACT is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If CT_CONTACT is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which

the CT_CONTACT environment variable is set. CT_IP_AUTHENT only has meaning if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon in processing the resources of the event-response resource manager (ERRM). The management scope determines the set of possible target nodes where the resources can be processed. The valid values are:

- 0**
Specifies *local* scope.
- 1**
Specifies *local* scope.
- 2**
Specifies *peer domain* scope.
- 3**
Specifies *management domain* scope.

If this environment variable is *not* set, *local* scope is used.

Standard Output

When the -h flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

These examples apply to standalone systems:

1. To remove the condition definition named "FileSystem space used", run this command:

```
rmcondition "FileSystem space used"
```

2. To remove the condition definition named "FileSystem space used" even if the condition is linked with responses, run this command:

```
rmcondition -f "FileSystem space used"
```

This example applies to management domains:

1. In this example, the current node is the management server. To remove the condition definition named "nodeB FileSystem space used" that is defined on managed node nodeB, run this command:

```
rmcondition "FileSystem space used:nodeB"
```

This example applies to peer domains:

1. To remove the condition definition named "nodeA FileSystem space used" that is defined on node nodeA, run this command from any node in the domain:

```
rmcondition "nodeA FileSystem space used:nodeA"
```

Location

/opt/rsct/bin/rmcondition

rmcondresp Command

Purpose

Deletes the link between a condition and one or more responses.

Syntax

To delete the link between a condition and one or more responses:

```
rmcondresp [-q] [-h] [-TV] condition[:node_name] [response [response...]]
```

To delete all of the links to one or more responses:

```
rmcondresp [-q] -r [-h] [-TV] response1 [response2...][:node_name]
```

To lock or unlock the condition/response association:

```
rmcondresp {-U | -L} [-h] [-TV] condition[:node_name] response
```

Description

The `rmcondresp` command deletes the link between a condition and one or more responses. A link between a condition and a response is called a *condition/response association*. The response is no longer run when the condition occurs. Use the `-r` flag to specify that the command parameters consist only of responses. This deletes all links to conditions for these responses. If only a condition is specified, links to all responses for that condition are deleted.

If a particular condition/response association is needed for system software to work properly, it may be locked. A locked condition/response association cannot be removed by the `rmcondresp` command. If the condition/response association you specify on the `rmcondresp` command is locked, it will not be removed; instead an error will be generated informing you that this condition/response association is locked. To unlock a condition/response association, you can use the `-U` flag. However, because a condition/response association is typically locked because it is essential for system software to work properly, you should exercise caution before unlocking it.

Flags

-q

Does not return an error when either *condition* or *response* does not exist.

-r

Indicates that all command parameters are responses. There are no conditions specified. This command removes condition/response associations from all conditions that are linked to the specified responses.

-h

Writes the command's usage statement to standard output.

-T

Writes the command's trace messages to standard error. For your software service organization's use only.

-V

Writes the command's verbose messages to standard output.

-U

Unlocks a condition/response association so it can be started, stopped, or removed. If a condition/response association is locked, this is typically because it is essential for system software to work properly. For this reason, you should exercise caution before unlocking it. When unlocking a condition/response association using the `-U` flag, no other operation can be performed by this command.

-L

Locks a condition/response association so it cannot be started, stopped, or removed. When locking a condition/response association using the **-L** flag, no other operation can be performed by this command.

Parameters

condition

Specifies the name of the condition linked to the response. The condition is always specified first unless the **-r** flag is used.

response

Specifies the name of a response or more than one response. The links from the specified responses to the specified condition are removed.

node_name

Specifies the node where the condition is defined. If the **-r** flag is used, it is the node where the response is defined. *node_name* is a node within the scope determined by the CT_MANAGEMENT_SCOPE environment variable.

Security

The user needs write permission for the IBM.Association resource class to run **rmcondresp**. Permissions are specified in the access control list (ACL) file on the contacted system.

Exit Status

0

The command ran successfully.

1

An error occurred with RMC.

2

An error occurred with a command-line interface script.

3

An incorrect flag was entered on the command line.

4

An incorrect parameter was entered on the command line.

5

An error occurred that was based on incorrect command-line input.

Environment Variables

CT_CONTACT

Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When CT_CONTACT is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If CT_CONTACT is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. CT_IP_AUTHENT only has meaning if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon in processing the resources of the event-response resource manager (ERRM). The management scope determines the set of possible target nodes where the resources can be processed. The valid values are:

- 0** Specifies *local* scope.
- 1** Specifies *local* scope.
- 2** Specifies *peer domain* scope.
- 3** Specifies *management domain* scope.

If this environment variable is *not* set, *local* scope is used.

Standard Output

When the `-h` flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

These examples apply to standalone systems:

1. To delete the link between the condition "FileSystem space used" and the response "Broadcast event on-shift", run this command:

```
rmcondresp "FileSystem space used" "Broadcast event on-
shift"
```

2. To delete the links between the condition "FileSystem space used" and all of its responses, run this command:

```
rmcondresp "FileSystem space used"
```

3. To delete the links between the condition "FileSystem space used" and the responses "Broadcast event on-shift" and "E-mail root anytime", run this command:

```
rmcondresp "FileSystem space used" "Broadcast event on-shift" "E-mail root
anytime"
```

4. To delete the links between the response "Broadcast event on-shift" and all of the conditions that use it, run this command:

```
rmcondresp -r "Broadcast event on-shift"
```

These examples apply to management domains:

1. To delete the link between the condition "FileSystem space used" on the management server and the response "Broadcast event on-shift", run this command on the management server:

```
rmcondresp "FileSystem space used" "Broadcast event on-
shift"
```

2. To delete the links between the condition "FileSystem space used" on the managed node nodeB and the responses "Broadcast event on-shift" and "E-mail root anytime", run this command on the management server:

```
rmcondresp "FileSystem space used":nodeB \
"Broadcast event on-shift" "E-mail root
anytime"
```

These examples apply to peer domains:

1. To delete the links between the condition "FileSystem space used" on nodeA in the domain and the responses "Broadcast event on-shift" and "E-mail root anytime", run this command on any node in the domain:

```
rmcondresp "FileSystem space used":nodeA \  
"Broadcast event on-shift" "E-mail root  
anytime"
```

2. To delete the links between all conditions on nodeA in the domain and the response "Broadcast event on-shift", run this command on any node in the domain:

```
rmcondresp -r "Broadcast event on-  
shift":nodeA
```

Location

/opt/rsct/bin/rmcondresp

rmcosi Command

Purpose

Removes a Common Operating System Image (COSI).

Syntax

rmcosi [-f] [-v] *COSI*

Description

The **rmcosi** command removes a Common Operating System Image (COSI) created with the **mkcosi** command. If the common image to be removed is being used by thin servers, the operation fails unless the force flag (-f) is specified. The -f flag terminates any thin server sessions with the common image so that the COSI can be removed. This command depends on the **bos.sysmgt.nim.master** fileset being present on the system.

Flags

Item	Description
-f	Forces the removal of the common image. If the common image is being used by thin servers, the thin servers will be taken offline so that the common image can be removed.
-v	Enables verbose debug output when the rmcosi command runs.

Exit Status

Item	Description
0	The command completed successfully.
>0	An error occurred.

Security

Access Control: You must have root authority to run the `rmcosi` command.

Examples

1. To common image named `cosi1`, enter:

```
rmcosi cosi1
```

Location

`/usr/sbin/rmcosi`

Files

Item	Description
<code>/etc/niminfo</code>	Contains variables used by NIM.

rmdel Command

Purpose

Removes a delta from a SCCS file.

Syntax

rmdel -r *SID File* ...

Description

The **rmdel** command removes the delta specified by the *SID* variable from each Source Code Control System (SCCS) file indicated in the *File* parameter. You can remove only the most recently created delta in a branch, or the latest trunk delta if it has no branches. In addition, the *SID* you specify must not be a version currently being edited for the purpose of making a delta. To remove a delta, you must either own the SCCS file and the directory, or you must have created the delta you want to remove.

If you specify a directory for the *File* parameter, the **rmdel** command performs the requested actions on all SCCS files (those with the **s.** prefix). If you specify a **-** (dash) for the *File* parameter, the **rmdel** command reads standard input and interprets each line as the name of an SCCS file. The **rmdel** command continues to read input until it reaches an end-of-file character.

After a delta has been removed, it is not included in any **g**-file created by the **get** command. However, the delta table entry remains in the **s.** file with an **R** by the entry to show that the delta has been removed.

Flags

Item	Description
-r <i>SID</i>	Removes the specified delta <i>SID</i> from the SCCS file. This flag is required.

Exit Status

This command returns the following exit values:

Item	Description

0	Successful completion.
>0	An error occurred.

Examples

To remove delta 1.3 from the **s.test.c** SCCS file, type:

```
rmdev -r 1.3 s.test.c
```

Files

Item	Description
<code>/usr/bin/rmdev</code>	Contains the rmdev command.
<code>s.files</code>	Files processed by the rmdev command.

rmdev Command

Purpose

Removes a device from the system.

Syntax

```
rmdev { -l | -p } Name [ -d | -S ] [ -f File ] [ -h ] [ -q ] [ -R ] [ -g ]
```

Description

Note: The **-l** flag cannot be specified if **-p** is specified. If the **-R** flag is specified along with the **-p** flag, it will be ignored.

The **rmdev** command unconfigures or both unconfigures and undefines the device specified with the device logical name using the **-l** *Name* flag. The default action unconfigures the device but retains its device definition in the Customized Devices object class.

If you specify the **-S** flag, the **rmdev** command sets the device to the Stopped state for devices that support the Stopped state. If you specify the **-d** flag, the **rmdev** command deletes the device definition from the Customized Devices object class (undefines). If you do not specify the **-d** flag, the **rmdev** command sets the device to the Defined state (unconfigures). If you specify the **-R** flag, the **rmdev** command acts on any children of the device as well.

Use the **-p** flag along with the parent device's logical name to unconfigure or delete all of the children devices. The children are unconfigured or deleted in the same recursive fashion as described for the **-R** flag, but the specified device itself is not unconfigured or deleted.



Attention: To protect the Configuration database, the **rmdev** command is not interruptible. Stopping this command before it is complete could result in a corrupted database.

You can use the System Management Interface Tool (SMIT) **smit rmdev** fast path to run this command.

Flags

Item	Description
-d	Removes the device definition from the Customized Devices object class. This flag cannot be used with the -S flag.
-f File	Reads the necessary flags from the <i>File</i> parameter.
-g	Forces the remove operation to run on a locked device.
-h	Displays the command usage message.
-l Name	Specifies the logical device, indicated by the <i>Name</i> parameter, in the Customized Devices object class. This flag cannot be used with the -p flag.
-p Name	Specifies the parent logical device (indicated by the <i>Name</i> parameter) in the Customized Devices object class, with children that must be removed. This flag may not be used with the -l flag.
-q	Suppresses the command output messages from standard output and standard error.
-R	Specifies to unconfigure the device and its children. When used with the -d or -S flags, the children are undefined or stopped, respectively.
-S	Makes the device unavailable by calling the Stop method if the device has a Stop method. This flag cannot be used with the -d flag.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Security

Privilege Control: Only the root user and members of the system group should have execute (x) access to this command.

Auditing Events:

Event	Information
DEV_Stop	Device name
DEV_Unconfigure	Device name
DEV_Remove	Device name

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To unconfigure the cd0 CD-ROM device while retaining its device definition in the Customized Devices object class, type the following:

```
imdev -l cd0
```

The system displays a message similar to the following:

```
cd0 defined
```

2. To remove the cd0 CD-ROM device definition from the Customized Devices object class, type the following:

```
rmdev -d -l cd0
```

The system displays a message similar to the following:

```
cd0 deleted
```

3. To unconfigure the scsi1 SCSI adapter and all of its children while retaining their device definitions in the Customized Devices object class, type the following:

```
rmdev -R -l scsi1
```

The system displays a message similar to the following:

```
rmt0 Defined  
hdisk1 Defined  
scsi1 Defined
```

4. To unconfigure the children of the scsi1 SCSI adapter, but not the adapter itself, while retaining their device definitions in the Customized Devices object class, type the following:

```
rmdev -p scsi1
```

The system displays a message similar to the following:

```
rmt0 Defined  
hdisk1 Defined
```

5. To unconfigure the children of the pci1 PCI bus and all other devices under them while retaining their device definitions in the Customized Devices object class, type the following:

```
rmdev -p pci1
```

The system displays a message similar to the following:

```
rmt0 Defined  
hdisk1 Defined  
scsi1 Defined  
ent0 Defined
```

Files

Item	Description
<code>/usr/sbin/rmdev</code>	Contains the rmdev command.

rmmdir Command

Purpose

Removes a directory.

Syntax

rmmdir [**-p**] *Directory ...*

Description

The **rmdir** command removes the directory, specified by the *Directory* parameter, from the system. The directory must be empty before you can remove it, and you must have write permission in its parent directory. Use the **ls -al** command to check whether the directory is empty. The directory must not be exported for use by the NFS version 4 server.

Note: The **rmdir** command supports the **—** (dash, dash) parameter as a delimiter that indicates the end of the flags.

Flags

Item	Description
-p <i>Directory</i>	Removes all directories along the path name specified by the <i>Directory</i> parameter. Parent directories must be empty and the user must have write permission in the parent directories before they can be removed.

Exit Status

This command returns the following exit values:

Item	Description
0	Each directory entry specified by a <i>Directory</i> parameter was removed successfully.
>0	An error occurred.

Examples

1. To empty and remove a directory, type:

```
rm mydir/* mydir/.  
rmdir mydir
```

This command removes the contents of the **mydir** file and then removes the empty directory. The **rm** command displays an error message about trying to remove the directories **.** (dot) and **..** (dot, dot), and then the **rmdir** command removes them.

Note that the **rm mydir/* mydir/.*** command first removes files with names that do not begin with a dot, and then removes those with names that do begin with a dot. You may not realize that the directory contains file names that begin with a dot because the **ls** command does not usually list them unless you use the **-a** flag.

2. To remove the **/home**, **/home/demo**, and **/home/demo/mydir** directories, type:

```
rmdir -p /home/demo/mydir
```

This command removes first the **/mydir** directory and then the **/demo** and **/home** directories, respectively. If a directory is not empty or does not have write permission when it is to be removed, the command terminates.

Files

Item	Description
/usr/bin/rmdir	Contains the rmdir command.

rmdom Command

Purpose

Removes the domains from the domain database.

Syntax

rmdom *Name*

Description

The **rmdom** command removes the domain that is identified by the *Name* parameter. The command only removes the existing domains from the domain database. A domain that is referenced by the domain object database cannot be removed until you remove the references to the domain.

When the system is operating in enhanced role-based access control (RBAC) mode, modifications made to the domains database are not used for security considerations until the database has been sent to the kernel security tables by using the **setkst** command.

Parameters

Item	Description
<i>Name</i>	Specifies the name of the domain to be removed.

Security

The **rmdom** command is a privileged command. You must have the following authorization to run the command:

Item	Description
aix.security.domains.remove	Required to remove the domain from the domain database.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

File Accessed

File	Mode
/etc/security/domains	rw

Examples

To remove the **h1dom** domain, type:

```
rmdom h1dom
```

rmf Command

Purpose

Removes folders and the messages they contain.

Syntax

rmf [**+Folder**] [**-interactive** | **-nointeractive**]

Description

The **rmf** command deletes the messages within the specified folder and then deletes the folder. By default, the **rmf** command confirms your request before deleting a folder. If the folder contains files that are not messages, the **rmf** command does not delete the files and returns an error.



Attention: The **rmf** command irreversibly deletes messages that do not have other links.

By default, the **rmf** command removes the current folder. When the current folder is removed, **inbox** becomes the current folder. If the **+Folder** flag is not specified, and the **rmf** command cannot find the current folder, the command requests confirmation before removing the **+inbox** folder.

The **rmf** command does not delete any folder or any messages in a folder to which you have read-only access. The **rmf** command deletes only your private sequences and your current message information from the profile.

The **rmf** command does not delete folders recursively. You cannot remove subfolders by requesting the removal of a parent folder. If you remove a subfolder, the parent of that folder becomes the current folder.

Flags

Item	Description
+Folder	Specifies the folder to be removed.
-help	Lists the command syntax, available switches (toggles), and version information. Note: For Message Handler (MH), the name of this flag must be fully spelled out.
-interactive	Requests confirmation before removing the folder. If the +Folder flag is not specified, this is the default.
-nointeractive	Removes the folder and its messages without requesting confirmation. This is the default.

Profile Entries

The following entries are entered in the *UserMhDirectory/.mh_profile* file:

Item	Description
Current-Folder:	Sets the default current folder.
Path:	Specifies the user's MH directory.

Security



Attention: This is for RBAC users. The **rmf** command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples



Attention: The **rmf** command irreversibly deletes messages that do not have other links.

1. To remove the current folder called status, enter:

```
rmf
```

The system responds with a message similar to the following:

```
Remove folder "status"?
```

If you do want the folder removed, enter yes. The system responds with a message similar to the following:

```
[+inbox now current]
```

2. To remove the meetings folder noninteractively, enter:

```
rmf +meetings
```

Files

Item	Description
\$HOME/.mh_profile	Defines the MH user profile.
/usr/bin/rmf	Contains the rmf command.

rmfilt Command

Purpose

Removes a filter rule from the filter table.

Syntax

```
rmfilt -v 4|6 -n fid | all [-f]
```

Description

Use the **rmfilt** command to remove filter rules from the filter rule table. Actions by this command will not effect the IP Security subsystem until the **mkfilt** command is executed. IPsec filter rules for this command can be configured by using the **genfilt** command, or IPsec smit (IP version 4 or IP version 6) in the Virtual Private Network submenu.

The **rmfilt** command removes a filter rules from the filter rule table. Only manual filter rules can be removed.

Flags

Item	Description
-f	Force to remove auto-generated filter rules. -f flag works with -n all to remove all the filter rules (user-defined and auto-generated filter rules) except rule number 1 for IP version 4.

Item	Description
-n	The ID of the filter rule you want to remove from the filter rule table. For IP version 4, the value of 1 is invalid for this flag, that is a reserved filter rule. If all is specified, all the user defined filter rules will be removed until the -f flag is specified.
-v	IP version of the filter rule you want to remove. Value 4 specifies IP version 4. Value 6 specifies IP version 6.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

rmfs Command

Purpose

Removes a file system.

Syntax

rmfs [**-r** | **-i**] *FileSystem*

Description

The **rmfs** command removes a file system. If the file system is a journaled file system (JFS or JFS2), the **rmfs** command removes both the logical volume on which the file system resides and the associated stanza in the **/etc/filesystems** file. If the file system is not a JFS or JFS2 file system, the command removes only the associated stanza in the **/etc/filesystems** file. The *FileSystem* parameter specifies the file system to be removed.

You could also use the System Management Interface Tool (SMIT) **smit rmfs** fast path to run this command.

Flags

Item	Description

-r	Removes the mount point of the file system.
-i	Displays warning and prompts the user before removing file system.

Exit Status

This command returns the following exit values:

Item	Description
0	File system is successfully removed.
>0	File system is not successfully removed.

Security

Access Control: Only the root user or a member of the **system** group can run this command.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

To remove the **/test** file system, enter:

```
rmfs /test
```

This removes the **/test** file system, its entry in the **/etc/filesystems** file, and the underlying logical volume.

Files

Item	Description
/etc/rmfs	Contains the rmfs command.
/etc/filesystems	Lists the known file systems and defines their characteristics.

rmgroup Command

Purpose

Removes a group.

Syntax

```
rmgroup [-p] [ -R load_module ] Name
```

Description

The **rmgroup** command removes a group specified by the *Name* parameter. This command deletes all the group attributes as well. To remove a group, the group name must already exist. Users who are group members are not removed from the system.

If the group is the primary group for any user, you cannot remove it unless you redefine the user's primary group with the **chuser** command. The **chuser** command alters the **/etc/passwd** file. Only the root user or a user with GroupAdmin authorization can remove an administrative group or a group with administrative users as members.

For groups that were created with an alternate Identification and Authentication (I&A) mechanism, the **-R** flag can be used to specify the I&A load module used. Load modules are defined in the **/usr/lib/security/methods.cfg** file.

You could also use the System Management Interface Tool (SMIT) **smit rmgroup** fast path to run this command.

Flag

Item	Description
-p	Removes the group keystore.
-R load_module	Specifies the loadable I&A module used to remove a group.

Exit Status

This command returns the following exit values:

Item	Description
0	The command executes successfully and all requested changes are made.
>0	An error occurred. The printed error message gives further details about the type of failure.

Security

Access Control: This command should grant execute (x) access only to the root user and members of the security group. This command should be installed as a program in the trusted computing base (TCB). The command should be owned by the root user with the **setuid** (SUID) bit set.

Files Accessed:

Mode	File
r	/etc/passwd
rw	/etc/group
rw	/etc/security/group

Auditing Events:

Event	Information
GROUP_Remove	group

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Limitations

Removing a group may not be supported by all loadable I&A modules. If the loadable I&A module does not support removing a group, an error is reported.

Examples

1. To remove the finance group, type:

```
imgroup finance
```

2. To remove the LDAP I&A loadable module group monsters, type:

```
imgroup -R LDAP monsters
```

Files

Item	Description
/usr/sbin/rmgroup	Contains the rmgroup command.
/etc/group	Contains the basic attributes of groups.
/etc/security/group	Contains the extended attributes of groups.

rmiscsi Command

Purpose

Removes iSCSI target data.

Syntax

`rmiscsi -l AdapterName [ -g group ] [ -t TargetName ] [ -n PortNumber ] [ -i IPaddress ]`

Description

The `rmiscsi` command removes iSCSI target data to ODM. There are two categories of data stored in ODM. The first is for statically configured iSCSI targets, which require that all the relevant iSCSI target information (such as target name, IP address, and port number) are specified in order for AIX to discover them. The second category of iSCSI target data is for iSCSI target devices that can be configured automatically, but require authentication from the host (such as passwords). These two categories of iSCSI target data are associated with the static and auto groups, respectively, specified by the `-g` flag.

Flags

Item

`-g group`

Description

Specifies which group this iSCSI target is associated with. There two valid groups are static and auto. The static group is for iSCSI targets that cannot be automatically discovered from this host; all relevant iSCSI target information for them (such as target name, IP address, and port number) must be specified. The auto group is for iSCSI targets that are automatically discovered, but require authentication information such as passwords.

`-i IPaddress`

Specifies the IP address of the iSCSI target.

`-l AdapterName`

Specifies the adapter name for the iSCSI TCP/IP Offload Engine (TOE) adapter that is attached to this iSCSI target. It can also specify the iSCSI protocol device for the iSCSI software solution device.

`-n PortNumber`

Specifies the port number on which the iSCSI target is accessed. The default port number is 3260.

`-t TargetName`

Specifies the iSCSI target name (for example, `iqn.sn9216.iscsi-hw1`).

Exit Status

Item

0

Description

The command completed successfully.

>0

An error occurred.

Security

The `rmiscsi` command is executable only by root.

Examples

1. To remove one statically configured iSCSI target, enter:

```
rmiscsi -l ics0 -g static -t iqn.sn1234.iscsi_hw1 -i 10.2.1.4 -n 3260
```

2. To remove all iSCSI targets for the iSCSI TOE adapter `ics0`, enter:

```
rmiscsi -l ics0
```

Location

`/usr/sbin/rmiscsi`

Files

Item	Description
<code>src/bos/usr/sbin/iscsia</code>	Contains the common source files from which the iSCSI commands are built.

rmitab Command

Purpose

Removes records in the `/etc/inittab` record. You can specify a record to remove by using the *Identifier* parameter. The *Identifier* parameter specifies a field of one to fourteen characters used to uniquely identify an object. If the *Identifier* field is not unique, the command is unsuccessful.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the `lssecattr` command or the `getcmdattr` subcommand.

Examples

To remove the `tty` entry for `tty2`, enter:

```
rmitab "tty002"
```

rmkeyserv Command

Purpose

Stops the `keyserv` daemon and comments the entry in the `/etc/rc.nfs` file.

Syntax

`/usr/sbin/rmkeyserv [ -I | -B | -N ]`

Description

The **rmkeyerv** command comments the entry for the **keyerv** daemon in the **/etc/rc.nfs** file. The **rmkeyerv** daemon stops the **keyerv** daemon by using the **stopsrc** command.

You could also use the System Management Interface Tool (SMIT) **smit rmkeyerv** fast path to run this command.

Flags

Item	Description
-I	Comments the entry for the keyerv daemon in the /etc/rc.nfs file.
-B	Comments the entry for the keyerv daemon in the /etc/rc.nfs file and stops the keyerv daemon. This flag is the default.
-N	Stops the keyerv daemon using the stopsrc command. This flag does not change the /etc/rc.nfs file.

Examples

To comment the entry in the **/etc/rc.nfs** file that starts the **keyerv** daemon, enter:

```
rmkeyerv -I
```

This command will not stop the currently executing daemon.

Files

Item	Description
/etc/rc.nfs	Contains the startup script for the NFS and NIS daemons.

rm1pcmd Command

Purpose

Removes one or more least-privilege (LP) resources from the resource monitoring and control (RMC) subsystem.

Syntax

To remove one or more LP resources:

- From the local node:

```
rm1pcmd [-h] [-TV] resource_name1 [ , resource_name2 , ... ]
```

- From all nodes in a domain:

```
rm1pcmd -a [-h] [-TV] resource_name1 [ , resource_name2 , ... ]
```

- From a subset of nodes in a domain:

```
rm1pcmd -n host1 [ , host2 , ... ] [-h] [-TV] resource_name1 [ , resource_name2 , ... ]
```

Description

The **rm1pcmd** command removes one or more LP resources from the RMC subsystem. An LP resource is a root command or script to which users are granted access based on permissions in the LP access control lists (ACLs). You can use the **rm1pcmd** command to remove LP resources from particular nodes or all

nodes in a domain. If you want to remove locked LP resources, you must first use the `ch1pcmd` command to unset the resource's Lock attribute.

This command runs on any node. If you want this command to run on all of the nodes in a domain, use the `-a` flag. If you want this command to run on a subset of nodes in a domain, use the `-n` flag. Otherwise, this command runs on the local node.

Flags

-a

Removes one or more LP resources from all nodes in the domain. The `CT_MANAGEMENT_SCOPE` environment variable's setting determines the cluster scope. If `CT_MANAGEMENT_SCOPE` is not set, the LP resource manager uses scope settings in this order:

1. The management domain, if it exists
2. The peer domain, if it exists
3. Local scope

The `rm1pcmd` command runs once for the first valid scope that the LP resource manager finds. For example, suppose a management domain and a peer domain exist and the `CT_MANAGEMENT_SCOPE` environment variable is not set. In this case, `rm1pcmd -a` runs in the management domain. To run `rm1pcmd -a` in the peer domain, you must set `CT_MANAGEMENT_SCOPE` to 2.

-n *host1[,host2,...]*

Specifies one or more nodes in the domain from which the LP resource is to be removed. By default, the LP resource is removed from the local node. The `-n` flag is valid only in a management or peer domain. If the `CT_MANAGEMENT_SCOPE` variable is not set, the LP resource manager uses scope settings in this order:

1. The management domain, if it exists
2. The peer domain, if it exists
3. Local scope

The `rm1pcmd` command runs once for the first valid scope that the LP resource manager finds.

-h

Writes the command's usage statement to standard output.

-T

Writes the command's trace messages to standard error.

-V

Writes the command's verbose messages to standard output.

Parameters

resource_name1[,resource_name2,...]

Specifies one or more LP resources to be removed.

Security

To run the `rm1pcmd` command, you need read and write permission in the Class ACL of the IBM.LPCommands resource class. Permissions are specified in the LP ACLs on the contacted system. See the `lpac1` file for general information about LP ACLs and the *RSCT Administration Guide* for information about modifying them.

Exit Status

0

The command has run successfully.

- 1**
An error occurred with RMC.
- 2**
An error occurred with the command-line interface (CLI) script.
- 3**
An incorrect flag was specified on the command line.
- 4**
An incorrect parameter was specified on the command line.
- 5**
An error occurred with RMC that was based on incorrect command-line input.
- 6**
The resource was not found.

Environment Variables

CT_CONTACT

Determines the system that is used for the session with the RMC daemon. When CT_CONTACT is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If CT_CONTACT is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the LP resources that are processed.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon to process the LP resource. The management scope determines the set of possible target nodes where the resource can be processed. The valid values are:

- 0**
Specifies *local* scope.
- 1**
Specifies *local* scope.
- 2**
Specifies *peer domain* scope.
- 3**
Specifies *management domain* scope.

If this environment variable is not set, *local* scope is used.

Implementation Specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) fileset for AIX.

Standard Output

When the -h flag is specified, this command's usage statement is written to standard output. When the -V flag is specified, this command's verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

1. To remove an LP resource named LP1, enter:

```
rm1pcmd LP1
```

2. To remove LP resources LP1 and LP2, enter:

```
rmlpcmd LP1 LP2
```

Location

/opt/rsct/bin/rmlpcmd

Contains the `rmlpcmd` command

rmlv Command

Purpose

Removes logical volumes from a volume group.

Syntax

```
rmlv [ -B ] [ -f ] [ -w ] [ -p Physical Volume ] LogicalVolume ...
```

Description



Attention: This command destroys all data in the specified logical volumes.

The **rmlv** command removes a logical volume. The *LogicalVolume* parameter can be a logical volume name or logical volume ID. The logical volume first must be closed. If the *volume group* is varied on in concurrent mode, the logical volume must be closed on all the concurrent nodes on which *volume group* is varied. For example, if the logical volume contains a file system, it must be unmounted. However, removing the logical volume does not notify the operating system that the file system residing on it have been destroyed. The command **rmfs** updates the `/etc/filesystems` file.

Note:

1. To use this command, you must either have root user authority or be a member of the **system** group.
2. You cannot use the **rmlv** command on a snapshot volume group or a volume group that has a snapshot volume group.
3. You cannot use the **rmlv** command on an active firmware that is assisted dump logical volume.
4. In AIX 7.2 Technology Level 1, or later, after the partition is freed by running the **rmlv** command, the space reclamation process runs for the freed partition.
5. **>** Delete the key server and key file authentication methods of all encrypted logical volumes that are configured on the physical volumes before you run the **rmlv** command. The **rmlv** command deletes only the Platform keystore (PKS) authentication method before the encrypted logical volume is deleted.**|<**

You can also use the System Management Interface Tool (SMIT) `smit rmlv` fast path to run this command.

Flags

Table 10. Flags

Item	Description
-B	Issues a <code>chlvcopy -B -s</code> for the parent logical volume if the logical volume was created by using the -l flag. If it is a regular logical volume, then the -B flag is ignored.
-f	Removes the logical volumes without requesting confirmation.

Table 10. Flags (continued)

Item	Description
-p <i>PhysicalVolume</i>	Removes only the logical partition on the <i>PhysicalVolume</i> . The logical volume is not removed unless there are no other physical partitions allocated.  Attention: If the logical volume spans multiple physical volumes, the removal of only logical partitions on the <i>PhysicalVolume</i> can jeopardize the integrity of the entire logical volume.
-w	Waits for the completion of the space reclamation process. Note: The wait time for space reclaim completion depends on the number of physical partitions.

Security

RBAC users

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples



Attention: The command that is used in this example destroys all data in the logical volumes.

To remove logical volume lv05 without requiring user confirmation, enter the following command:

```
rmlv -f lv05
```

The logical volume is removed from the volume group.

Files

Table 11. Files

Item	Description
/usr/sbin/rmlv	Directory where the rmlv command resides.
/tmp	Directory where the temporary files are stored while the command is running.
/etc/filesystems	Lists the known file systems and defines their characteristics.

rmlvcopy Command

Purpose

Removes copies from a logical volume.

Syntax

```
rmlvcopy[ -p mirrorpool] LogicalVolume Copies [ PhysicalVolume ... ]
```

Description

The **rmlvcopy** command removes copies from each logical partition in the *LogicalVolume*. Copies are the physical partitions which, in addition to the original physical partition, make up a logical partition. You can have up to two copies in a logical volume. The *Copies* parameter determines the maximum number of physical partitions that remain. The *LogicalVolume* parameter can be a logical volume name or logical volume ID. The *PhysicalVolume* parameter can be the physical volume name or the physical volume ID. If the *PhysicalVolume* parameter is used, then only copies from that physical volume will be removed.

You could also use the System Management Interface Tool (SMIT) **smit rmlvcopy** fast path to run this command.

Note:

1. To use this command, you must either have `root` user authority or be a member of the **system** group.
2. If LVM has not recognized that a disk has failed it is possible that LVM will remove a different mirror. Therefore if you know that a disk has failed and LVM does not show those disks as missing you should specify the failed disks on the command line or you should use **replacepv** to replace the disk or **reducevg** to remove the disk.
3. The **rmlvcopy** command is not allowed on a snapshot volume group.
4. The **rmlvcopy** command is allowed on a volume group that has a snapshot volume group only if the physical volume names are specified on the command line and the specified physical volumes belong to the snapshot volume group.
5. Running the **rmlvcopy** command on an active firmware-assisted dump logical volume temporarily changes the dump device to the `/dev/sysdumpnull` file. After the successful removal of the logical volume copy, the **rmlvcopy** command calls the **sysdumpdev -P** command to set the firmware-assisted dump logical volume to the original dump logical volume.
6. In AIX 7.2 Technology Level 1, or later, after the partition is freed by running the `rmlvcopy` command, the space reclamation process runs for the freed partition.

Flags

-p mirrorpool

Removes a copy from the specified mirror pool. To remove more than one copy, provide multiple **[-p mirrorpool]** flags.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

To reduce the number of copies of each logical partition belonging to logical volume `lv0112`, enter:

```
rmlvcopy lv0112 2
```

Each logical partition in the logical volume now has at most two physical partitions.

Files

Item	Description
<code>/usr/sbin/rmlvcopy</code>	Contains the rmlvcopy command.
<code>/tmp/*</code>	Directory where the temporary files are stored while the command is running.

rmm Command

Purpose

Removes messages from active status.

Syntax

rmm [**+ Folder**] [*Messages*]

Description

The **rmm** command removes messages from active status by renaming them. To rename a message, the system prefaces the current message number with a , (comma). Inactive files are unavailable to the Message Handler (MH) package. However, system commands can still manipulate inactive files.

Note: The **rmm** command does not change the current message.

Inactive messages should be deleted periodically. An entry can be placed in your **crontab** file to automatically delete all files beginning with a comma.

Flags

Item	Description
+Folder	Specifies the folder containing the messages to rename.
<i>Messages</i>	Specifies the messages to rename. You can specify several messages, a range of messages, or a single message. Use the following references to specify a message: Number Number of the message Sequence A group of messages specified by the user. Recognized values include: all All messages in a folder cur or . (dot) Current message. This is the default. first First message in a folder last Last message in a folder next Message following the current message prev Message preceding the current message
-help	Lists the command syntax, available switches (toggles), and version information. Note: For MH, the name of this flag must be fully spelled out.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command

Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To remove the current message in the current folder, enter:

```
rmm
```

2. To remove messages 2 through 5 from the sales folder, enter:

```
rmm +sales 2-5
```

Profile Entries

The following entries are entered in the *UserMhDirectory/.mh_profile*:

Item	Description
Current-Folder:	Sets the default current folder.
Path:	Specifies the <i>UserMhDirectory</i> .
immproc:	Specifies the program used to remove messages from a folder.

Files

Item	Description
\$HOME/.mh_profile	Contains the MH user profile.
/usr/bin/rmm	Contains the rmm command.

rmnamsv Command

Purpose

Unconfigures TCP/IP-based name service on a host.

Syntax

```
rmnamsv [ -f | -F FileName ]
```

Description

The **rmnamsv** high-level command unconfigures a TCP/IP-based name service on a host. You can unconfigure name service for a host functioning as a client.

To unconfigure name service for a client, the **rmnamsv** command calls the **namerslv** low-level command to unconfigure entries in the **/etc/resolv.conf** file or to rename the **/etc/resolv.conf** file to a default or user-specified file name.

You could also use the System Management Interface Tool (SMIT) **smit rmnamerslv** fast path to run this command.

Flags

Item	Description
-F <i>FileName</i>	Renames the system configuration database to the file name specified by <i>FileName</i> .

Item	Description
-f	Specifies that the default file name (/etc/resolv.conf.sv) should be used to rename the /etc/resolv.conf file.

Files

Item	Description
/etc/resolv.conf	Contains the default system configuration database.

rmnfs Command

Purpose

Changes the configuration of the system to stop running NFS daemons.

Syntax

/usr/sbin/rmnfs [**-I** | **-N** | **-B**]

Description

The **rmnfs** command changes the current configuration of the system so that the **/etc/rc.nfs** file is not executed on system restart. In addition, you can direct the command to stop NFS daemons that are currently running.

Flags

Item	Description
-B	Removes the entry in the inittab file and stops NFS daemons that are currently executing. This flag is the default.
-I	Removes the entry in the inittab file that starts NFS daemons on system restart.
-N	Stops immediately NFS daemons and does not change the inittab file.

Examples

To stop all of the NFS daemons immediately, enter:

```
rmnfs -N
```

This command will not change the **inittab** file.

rmnfsexp Command

Purpose

Unexports a directory from NFS clients.

Syntax

/usr/sbin/rmnfsexp **-d** *Directory* [**-V** *Exported Version*] [**-f** *Exports_file*] [**-I** | **-B** | **-N**] [**-F**]

Description

The **rmnfsexp** command removes an entry from the exports list for NFS clients. This command starts the **exportfs** command to unexport the specified directory. If an entry exists in the **/etc/exports** file, that entry is removed.

Flags

Item	Description
-d <i>Directory</i>	Specifies the directory to be unexported.
-f <i>Exports_File</i>	Specifies the full path name of the exports file to use if other than the /etc/exports file.
-I	Directs the command to remove the entry from the /etc/exports file without executing the exportfs command.
-B	Removes the entry in the /etc/exports file for the directory specified, and executes the exportfs command to remove the export.
-N	Unexports the directory immediately by invoking the exportfs command. The /etc/exports file is not modified with this flag.
-V <i>Exported Version</i>	Specifies the version to be used for unexporting the directory. The valid version numbers are 2, 3 and 4.
-F	Forces to unexport the directory.

Examples

1. To unexport a directory immediately, enter the following command:

```
rmnfsexp -d /usr -N
```

In this example, the **/usr** directory is unexported immediately.

2. To unexport a directory immediately and after every system restart, enter the following command:

```
rmnfsexp -d /home/guest -B
```

3. To unexport a directory immediately from an exports file other than the **/etc/exports** file, enter the following command:

```
rmnfsexp -d /usr -f /etc/exports.other -N
```

4. To unexport the **/common/documents** directory that is exported as version 3, enter the following command:

```
rmnfsexp -d /common/documents -V 3
```

Files

Item	Description
/etc/xtab	Lists the currently exported directories.
html	

rmnfsmnt Command

Purpose

Removes an NFS mount.

Syntax

`/usr/sbin/rmnfsmnt -f PathName [ -I | -B | -N ]`

Description

The **rmnfsmnt** command removes the appropriate entry from the **/etc/filesystems** file and unmounts the file system specified. When used with the **-N** flag, the **rmnfsmnt** command unmounts the file system and does not modify the **/etc/filesystems** file.

Flags

Item	Description
-B	Removes the entry in the /etc/filesystems file and unmounts the directory. If no entry exists in the /etc/filesystems file, the flag makes no changes to the file. If the file system is not currently mounted, the flag does not attempt to unmount it. This flag is the default.
-f PathName	Specifies the path name of the NFS-mounted file system.
-I	Removes the entry specified by the path name from the /etc/filesystems file.
-N	Unmounts the specified directory and does not modify the /etc/filesystems file.

Examples

1. To unmount a file system, enter:

```
rmnfsmnt -f /usr/man -N
```

In this example, the **/usr/man** file system is unmounted.

2. To remove a mount for a file, enter:

```
rmnfsmnt -f /usr/local/man -B
```

In this example, the mount for the **/usr/local/man** file is removed.

File

Item	Description
/etc/filesystems	Lists the remote file systems to mount during the system restart.

rmnfsproxy Command

Purpose

Removes a previously configured and mounted instance of a proxy-enabled Caches.

Syntax

`/usr/sbin/rmnfsproxy Cachefs_mount_point`

Description

The specified Cachefs mount is unmounted. The corresponding NFS client mount is also unmounted. Finally, all cached information created in the local file system is removed.

Note: If the Cachefs instance is NFS-exported, the instance must first be unexported before running `rmnfsproxy`.

Parameters

Item	Description
<i>Cachefs_mount_point</i>	Specifies where the proxy-enabled Cachefs instance to be removed was mounted.

Exit Status

Item	Description
0	The command completed successfully.
>0	An error occurred.

Examples

1. To remove a previously configured `/proj1_cached` Cachefs instance, enter:

```
rmnfsproxy /proj1_cached
```

Location

`/usr/sbin/rmnfsproxy`

rmnotify Command

Purpose

Removes a notify method definition from the Notify object class.

Syntax

`rmnotify -n NotifyName`

Description

The **rmnotify** command removes a notify method definition from the notify object class.

Flags

Item	Description
-n <i>NotifyName</i>	Specifies the notify method definition to be removed. The rmnotify command is unsuccessful if the <i>NotifyName</i> name does not already exist in the Notify object class.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Files

Item	Description
/etc/objrepos/SRCnotify	Specifies the SRC Notify Method object class.

rmpath Command

Purpose

Removes from the system a path to an MPIO capable device.

Syntax

rmpath [**-l** *Name*] [**-p** *Parent*] [**-w** *Connection*] [**-i** *PathID*]

rmpath [**-l** *Name*] [**-p** *Parent*] [**-w** *Connection*] [**-d**] [**-g**]

rmpath -h

Description

The **rmpath** command unconfigures, and possibly undefines, one or more paths associated with the specified target device (**-l** *Name*). The set of paths that are removed are determined by the combination of the **-l** *Name*, **-p** *Parent*, and **-w** *Connection* flags. If the command will result in all paths associated with the device being unconfigured or undefined, the command will exit with an error and without unconfiguring or undefining any path. In this situation, **rmdev** command must be used instead to unconfigure or undefine the target device itself.

The default action unconfigures each specified path, but does not completely remove it from the system. If the **-d** flag is specified, the **rmpath** command unconfigures (if necessary) and removes, or deletes, the path definition(s) from the system.

When the **rmpath** command finishes, it displays a status message. When unconfiguring paths, it is possible for this command to be able to unconfigure some paths and not others (e.g., paths that are in the process of doing I/O cannot be unconfigured).

The **rmpath** command provides status messages about the results of operation. Messages in one of the following formats will be generated:

path [defined | deleted]

This message is displayed when a single path was successfully unconfigures or undefined. If the path is successfully configured the message `path available` displays. If the path is not successfully configured and there is no explicit error code returned by the method, the message `path defined` displays.

paths [defined | deleted]

This message is displayed if multiple paths were identified and all paths were successfully unconfigured or undefined. If the **-d** flag is not specified, the message would be `paths defined`. If the **-d** flag is specified, the message would be `paths deleted`.

some paths [defined | deleted]

This message is display if multiple paths were identified, but only some of them were successfully unconfigured or undefined. If the **-d** flag is not specified, the message would be `some paths defined`. If the **-d** flag is specified, the message would be `some paths deleted`.

no paths processed

This message is generated if no paths were found matching the selection criteria.

Flags

Item	Description
-d	Indicates that the specified paths are to be deleted from the system.
-g	Forces the remove path operation to run on a locked device.
-h	Displays the command usage message.
-i PathID	Indicates the path ID associated with the path to be removed and is used to uniquely identify a path.
-l Name	Specifies the logical device name of the target device whose path is to be removed. The paths to be removed are qualified via the -p and -w flags.
-p Parent	Indicates the logical device name of the parent device to use in qualifying the paths to be removed. Since all paths to a device cannot be removed by this command, either this flag, the -w flag, or both must be specified.
-w Connection	Indicates the connection information to use in qualifying the paths to be removed. Since all paths to a device cannot be removed by this command, either this flag, the -p flag, or both must be specified.

Security

Privilege Control: Only the root user and members of the system group have execute access to this command.

Auditing Events:

Event	Information
DEV_Change	rmpath,Unconfigure,<unconfigure method arguments>
DEV_Change	rmpath,Undefine,<undefine method arguments>

Examples

1. To unconfigure the path from **scsi0** to **hdisk1** at connection **5,0**, type:

```
rmpath -l hdisk1 -p scsi0 -w "5,0"
```

The message generated would be similar to:

```
path defined
```

2. To unconfigure all paths from **scsi0** to **hdisk1**, type:

```
rmpath -l hdisk1 -p scsi0
```

If all paths were successfully unconfigured, the message generated would be similar to:

```
paths defined
```

However, if only some of the paths were successfully unconfigured, the message would be similar to:

```
some paths defined
```

3. To undefine the path definition between **scsi0** and **hdisk1** at connection **5,0**, type:

```
rmpath -d -l hdisk1 -p scsi0 -w "5,0"
```

The message generated would be similar to the following:

```
path deleted
```

4. To unconfigure all paths from **scsi0** to **hdisk1**, type:

```
rmpath -d -l hdisk1 -p scsi0
```

The message generated would be similar to:

```
paths deleted
```

Files

Item	Description
/usr/sbin/rmpath	Contains the rmpath command.

rmprtsv Command

Purpose

Unconfigures a print service on a client or server machine.

Syntax

```
rmprtsv { -c | -s } [ -T | -U | -A ] [ -h "HostName ..." | -H FileName ] [ -q "QEntry ..." ] [ -q QEntry -v "DeviceName ..." ]
```

Description

The **rmprtsv** high-level command unconfigures a print service on a client or server machine.

To unconfigure print service for a client, the **rmprtsv** command calls the **rmque** and **rmqueuedev** commands to disable the client spool queue and to remove the appropriate entries in the **/etc/qconfig** file.

To unconfigure print service for a server, the **rmprtsv** command performs the following procedure:

1. Calls the **stopsrc** command to deactivate the **lpd** and **qdaemon** servers.
2. Calls the **ruser** low-level command to unconfigure remote users on the print server.
3. Calls the **rmque** and **rmqueuedev** commands to unconfigure the spooler and its device queues, and delete the appropriate entries in the server's **/usr/lib/lpd/qconfig** file.

Flags

Item	Description
-A	Removes specified entries from the /etc/qconfig file but does not fully unconfigure print service.

Item	Description
-c	Unconfigures print service for a client machine. Use the -q flag with the -c flag.
-H FileName	Specifies the name of a file containing a list of host names to be configured for print service.
-h "HostName..."	Specifies a list of remote host names not allowed to use the print server. Note that the queuing system does not support multibyte host names.
-q "QEntry..."	Specifies a list of entries to remove from the /etc/qconfig file.
-s	Unconfigures print service for a server machine. The -h , -H , and -q flags should be used with the -s flag.
-T	Stops print service but does not fully unconfigure print service.
-U	Removes specified remote users on the print server but does not fully unconfigure print service.
-v "DeviceName..."	Specifies a list of the names of the device stanzas in the qconfig file. Must be used with the -q QEntry flag.

Files

Item	Description
/etc/qconfig	Contains configuration information for the printer queueing system.

rmpps Command

Purpose

Removes an inactive paging space.

Syntax

rmpps [**-t ps_helper**] *PagingSpace*

Description

The **rmpps** command removes an inactive paging space. The *PagingSpace* parameter specifies the name of the paging space that must be removed. This paging space is the name of the logical volume on which the paging space is present.

For an NFS paging space, the *PagingSpace* parameter specifies the name of the paging space to be removed. The device and its definition, which corresponds to this paging space, is removed from the system. Nothing is changed on the NFS server where the file that is used for paging is present.

If the **-t** flag is specified, the argument is assumed to be a third-party helper executable. If the helper executable is present in the **/sbin/helpers/pagespace** path, the executable is created by passing the **-r** flag to specify the **rmpps** command. The **/etc/swapspaces** directory is modified so that the helper executable returns zero.

The helper executable is used to remove the paging space. If the named helper does not exist in the **/sbin/helpers/pagespace** directory, the **rmpps** command displays a usage error. The helper executable exits with a value 0 when successful and a non-zero value when it fails.

Active pages can be removed by first deactivating them with the **swapoff** command.

Flags

Item	Description

-t	Specifies to use the helper program under /sbin/helpers/pagespace directory.

ps_helper	

	Name of the helper program for a third-party device.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To remove PS01 paging space, run the following command:

```
rmpps PS01
```

This removes the PS01 paging space.

2. To remove PS01 paging space by using the helper program foo, run the following command:

```
rmpps -t foo PS01
```

This removes the PS01 paging space.

Files

Item	Description
/etc/swapspace	Specifies the paging space devices and their attributes.

rmqos Command

Purpose

Changes the configuration of the system to remove QoS support.

Syntax

```
/usr/sbin/rmqos [ -I | -N | -B ]
```

Description

The **rmqos** command changes the current configuration of the system to remove Quality of Service (QoS) support.

Flags

Item	Description
-B	Removes the entry in the inittab file that enables QoS at system startup and stops the QoS daemons. This flag is the default.

Item	Description
-I	Removes the entry in the inittab file that enables QoS at system startup but does not affect the currently running QoS subsystem.
-N	Disables QoS support immediately but does not change the inittab file.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Files

Item	Description
inittab	Controls the initialization process of the system.
/etc/rc.qos	Contains the startup script for the QoS daemons.

rmque Command

Purpose

Removes a printer queue from the system.

Syntax

rmque **-q** *Name*

Description

The **rmque** command removes a queue from the system configuration by deleting the queue stanza named by the **-q** flag from the **/etc/qconfig** file. All queue devices must be deleted using the **rmquedev** command before entering this command.

You could also use the System Management Interface Tool (SMIT) **smit rmque** fast path to run this command.

Recommendation: To edit the **/etc/qconfig** file, use the **chque**, **mkque**, **rmque**, **chquedev**, **mkquedev**, and **rmquedev** commands or SMIT. Further, it is recommended to run these commands during slow or off-peak time.

If manual editing of the **/etc/qconfig** file is necessary, you can first issue the **enq -G** command to bring the queuing system and the **qdaemon** to a halt after all jobs are processed. Then you can edit the **/etc/qconfig** file and restart the **qdaemon** with the new configuration.

Flags

Item	Description
-q <i>Name</i>	Specifies the name of the queue to be removed.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command

Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

To remove printer queue lp0, enter:

```
rmque -q lp0
```

Files

Item	Description
/usr/bin/rmque	Contains the rmque command.
/etc/qconfig	Contains the configuration file.

rmqueuedev Command

Purpose

Removes a printer or plotter queue device from the system.

Syntax

rmqueuedev **-d** *Name* **-q** *Name*

Description

The **rmqueuedev** command removes a printer or plotter queue device from the system configuration by deleting the device stanza named by the **-d** flag from the **/etc/qconfig** file. It also modifies the **Device=DeviceName1, DeviceName2, DeviceName3** line of the queue stanza, deleting the entry for the device *Name*.

You could also use the System Management Interface Tool (SMIT) **smit rmqueuedev** fast path to run this command.

Recommendation: To edit the **/etc/qconfig** file, use the **chque**, **mkque**, **rmque**, **chqueuedev**, **mkqueuedev**, and **rmqueuedev** commands or SMIT. Further, it is recommended to run these commands during slow or off-peak time.

If manual editing of the **/etc/qconfig** file is necessary, you can first issue the **enq -G** command to bring the queuing system and the **qdaemon** to a halt after all jobs are processed. Then you can edit the **/etc/qconfig** file and restart the **qdaemon** with the new configuration.

Flags

Item	Description
-d <i>Name</i>	Specifies the <i>Name</i> of the device stanza to be deleted from the qconfig file.
-q <i>Name</i>	Specifies the <i>Name</i> of the device to be modified in the preceding queue stanza.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

To delete the `loc` device stanza from the `/etc/qconfig` file and modify the "DEVICE =" stanza in the preceding queue stanza `lpq`, enter:

```
rmqueuedev -q lpq -d loc
```

Files

Item	Description
<code>/usr/bin/rmqueuedev</code>	Contains the rmqueuedev command.
<code>/etc/qconfig</code>	Configuration file.

rmramdisk Command

Purpose

Removes RAM disks created by the **mkramdisk** command.

Syntax

```
rmramdisk ram_disk_name
```

Description

The **rmramdisk** command removes the specified RAM disk and the device special files that were created for that RAM disk. RAM disks are also removed when the system is rebooted. Device special files can only be removed via the **rmramdisk** command.

Parameters

Item	Description
<u>ram_disk_name</u>	Name of the specific RAM disk to be removed from memory. If not specified, an error is returned. The names of the RAM disks are in the form of rramdiskx where x is the logical RAM disk number (0 through 63).

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

```
# ls -l /dev/*ramdisk2
brw----- 1 root    system 38,  0 Aug 01 05:52 /dev/ramdisk2
crw----- 1 root    system 38,  0 Aug 01 05:52 /dev/irramdisk2
```

To remove the ramdisk2, enter:

```
# rmramdisk ramdisk2

# ls -l /dev/*ramdisk2
ls: 0653-341 The file /dev/*ramdisk2 does not exist.
```

Files

Item	Description
<code>/usr/sbin/rmramdisk</code>	Contains the rmramdisk command.

rmresponse Command

Purpose

Removes a response.

Syntax

```
rmresponse [-f] [-q] [-h] [-TV] response[:node_name]
```

Description

The **rmresponse** command removes the response specified by the *response* parameter. The response must already exist in order to be removed. When the response must be removed even if it is linked with conditions, specify the **-f** flag. This forces the response and the links with the conditions to be removed. If the **-f** flag is not specified and links with conditions exist, the response is not removed. This command does not remove conditions.

If a particular response is needed for system software to work properly, it may be locked. A locked response cannot be modified or removed until it is unlocked. If the response you specify on the **rmresponse** command is locked, it will not be removed; instead an error will be generated informing you that the response is locked. To unlock a response, you can use the **-U** flag of the **chresponse** command. However, since a response is typically locked because it is essential for system software to work properly, you should exercise caution before unlocking it.

Flags

- f**
Forces the response to be removed even if it is linked with conditions. The links with the conditions are removed as well as the response, but the conditions are not removed.
- q**
Does not return an error when *response* does not exist.
- h**
Writes the command's usage statement to standard output.
- T**
Writes the command's trace messages to standard error. For your software service organization's use only.
- v**
Writes the command's verbose messages to standard output.

Parameters

response

Specifies the name of a defined response to be removed.

node_name

Specifies the node in a cluster where the response is defined. If *node_name* is not specified, the local node is used. *node_name* is a node within the scope determined by the CT_MANAGEMENT_SCOPE environment variable.

Security

The user needs write permission for the IBM.EventResponse resource class to run `rmresponse`. Permissions are specified in the access control list (ACL) file on the contacted system.

Exit Status

0

The command ran successfully.

1

An error occurred with RMC.

2

An error occurred with a command-line interface script.

3

An incorrect flag was entered on the command line.

4

An incorrect parameter was entered on the command line.

5

An error occurred that was based on incorrect command-line input.

Environment Variables

CT_CONTACT

Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When CT_CONTACT is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If CT_CONTACT is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. CT_IP_AUTHENT only has meaning if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon in processing the resources of the event-response resource manager (ERRM). The management scope determines the set of possible target nodes where the resources can be processed. The valid values are:

0

Specifies *local* scope.

1

Specifies *local* scope.

2

Specifies *peer domain* scope.

3

Specifies *management domain* scope.

If this environment variable is *not* set, *local* scope is used.

Standard Output

When the `-h` flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

These examples apply to standalone systems:

1. To remove the response definition named "Broadcast event on-shift", run this command:

```
rmresponse "Broadcast event on-shift"
```

2. To remove the response definition named "Broadcast event on-shift" even if the response is linked with conditions, run this command:

```
rmresponse -f "Broadcast event on-shift"
```

This example applies to management domains:

1. In this example, the current node is the management server. To remove the response definition named "Broadcast event on-shift" on managed node nodeB, run this command:

```
rmresponse "Broadcast event on-shift":nodeB
```

This example applies to peer domains:

1. To remove the response definition named "Broadcast event on-shift" defined on node nodeA, run this command from any node in the domain:

```
rmresponse "Broadcast event on-shift":nodeA
```

Location

/opt/rsct/bin/rmresponse

rmrole Command

Purpose

Removes a role.

Syntax

rmrole [**-R** *load_module*] *Name*

Description

The **rmrole** command removes the role identified by the *Name* parameter from the **/etc/security/roles** file. The role name must already exist.

You can use the System Management Interface Tool (SMIT) to run the **rmrole** command.

If the system is configured to use databases from multiple domains, the **rmrole** command finds the first match from the database domains in the order that it was specified by the **secorder** attribute of the roles

stanza in the **/etc/nscontrol.conf** file. Meanwhile, the **rmrole** command removes the role entry from the domain. If any matching roles from the rest of the domains exist, they are not affected. Use the **-R** flag to remove a role from a specific domain.

When the system is operating in enhanced role based access control (RBAC) mode, roles removed from the role database still exist in the kernel security tables (KST) until the KST is updated with the **setkst** command.

Flags

Item	Description
-R <i>load_module</i>	Specifies the loadable module to use for role deletion.

Security

The **rmrole** command is a privileged command. You must have the **aix.security.role.remove** authorization to run the command:

Item	Description
aix.security.role.remove	Required to run the command.

Files Accessed:

Mode	File
rw	/etc/security/roles
r	/etc/security/user.roles

Auditing Events:

Event	Information
ROLE_Remove	role

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To remove the ManageObjects role, use the following command:

```
rmrole ManageObjects
```

2. To remove the ManageRoles role from LDAP, use the following command:

```
rmrole -R LDAP ManageRoles
```

Files

Item	Description
/etc/security/roles	Contains the attributes of roles.
/etc/security/user.roles	Contains the role attribute of users.

rmrpdomain Command

Purpose

Removes a peer domain that has already been defined.

Syntax

```
rmrpdomain [-f] [-q] [-h] [-TV] peer_domain
```

Description

The `rmrpdomain` command removes the peer domain definition that is specified by the *peer_domain* parameter. The peer domain that is to be removed must already be defined. This command must be run on a node that is defined in the peer domain. When `rmrpdomain` is run on a node that is online to the peer domain, it removes the peer domain definition on all nodes defined to the peer domain that are reachable from that node. If a node defined to the peer domain is not reachable, that node's local peer domain definition is not removed. To remove the local peer domain definition when the peer domain is not online or when the node is not online to the peer domain, run the `rmrpdomain` command on that node and specify the `-f` flag.

The most efficient way to remove a peer domain definition is to make sure the peer domain is online. Then, from a node that is online to the peer domain, run the `rmrpdomain` command. If there are nodes that are not reachable from the node on which the `rmrpdomain` command was run, on each of those nodes, run the `rmrpdomain` command using the `-f` flag. This can be done at a later time if the node itself is not operational.

The `-f` flag must also be used to override a subsystem's rejection of the peer domain removal. A subsystem may reject the request if a peer domain resource is busy, for example. Specifying the `-f` flag in this situation indicates to the subsystems that the peer domain definition must be removed.

The `rmrpdomain` command does not require configuration quorum. Therefore, this command is still successful if it is issued to a minority subcluster. Later, the majority subcluster may become active. If so, the domain is still removed.

If a Cluster-Aware AIX (CAA) cluster is configured and this peer domain is representing it, the **rmrpdomain** command removes the underlying CAA cluster as well.

Flags

-f

Forces the peer domain to be removed. The force flag is required to remove a peer domain definition:

- from the local node when the node is not online to the peer domain.
- when a subsystem may reject the request, as when resources are allocated, for example.

-q

Specifies quiet mode. The command does not return an error if the peer domain does not exist.

-h

Writes the command's usage statement to standard output.

-T

Writes the command's trace messages to standard error. For your software service organization's use only.

-V

Writes the command's verbose messages to standard output.

Parameters

peer_domain

Specifies the name of the defined peer domain that is to be removed.

Security

The user of the `rmrpdomain` command needs write permission to the `IBM.PeerDomain` resource class on each node that is to be defined to the peer domain. By default, `root` on any node in the peer domain has read and write access to this resource class through the configuration resource manager.

Exit Status

0

The command ran successfully.

1

An error occurred with RMC.

2

An error occurred with a command-line interface script.

3

An incorrect flag was entered on the command line.

4

An incorrect parameter was entered on the command line.

5

An error occurred that was based on incorrect command-line input.

6

The peer domain definition does not exist.

Environment Variables

CT_CONTACT

Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When `CT_CONTACT` is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If `CT_CONTACT` is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

CT_IP_AUTHENT

When the `CT_IP_AUTHENT` environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the `CT_CONTACT` environment variable is set. `CT_IP_AUTHENT` only has meaning if `CT_CONTACT` is set to an IP address; it does not rely on the domain name system (DNS) service.

Restrictions

The node on which this command is run must be defined to the peer domain and should be able to reach all of the nodes that are defined to the peer domain. The node's local peer domain definition will not be removed if the node is not reachable.

Implementation Specifics

This command is part of the **rsct.basic.rte** fileset for AIX®.

Standard Input

When the `-f` " " or `-F` " " flag is specified, this command reads one or more node names from standard input.

Standard Output

When the `-h` flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

1. To remove the peer domain definition of `ApplDomain` where `nodeA`, `nodeB`, and `nodeC` are defined and online to *ApplDomain*, and all are reachable to each other, run this command on `nodeA`, `nodeB`, or `nodeC`:

```
rmrpdomain ApplDomain
```

2. To remove the local peer domain definition of `ApplDomain` on `nodeD` when `nodeD` is not online to the peer domain, the peer domain is offline, or the peer domain does not exist, run this command on `nodeD`:

```
rmrpdomain -f ApplDomain
```

3. To remove the peer domain definition of `ApplDomain` where `nodeA`, `nodeB`, and `nodeC` are defined and online to *ApplDomain*, all are reachable to each other, and to prevent a subsystem from rejecting the request, run this command on `nodeA`, `nodeB`, or `nodeC`:

```
rmrpdomain -f ApplDomain
```

Location

`/opt/rsct/bin/rmrpdomain`

Files

The `/etc/services` file is modified.

rmrpnode Command

Purpose

Removes one or more nodes from a peer domain definition.

Syntax

```
rmrpnode [-f] [-q] [-h] [-TV] node_name1 [node_name2 ...]
```

```
rmrpnode -F { file_name | "-" } [-f] [-q] [-h] [-TV]
```

Description

The `rmrpnode` command removes one or more nodes from the online peer domain where the command is run. The command must be run on a node that is online to the peer domain in which the nodes are to be removed. The nodes that are to be removed must be offline to the peer domain and must be reachable from the node where the command is run. To take nodes offline, use the `stoprpnode` command.

If a Cluster-Aware AIX (CAA) cluster is configured and this peer domain is representing it, the **rmrpnode** command removes the nodes from the underlying CAA cluster as well.

Specifying the **-f** flag forces the specified nodes to be removed from the peer domain. When the last tiebreaker node is removed using **rmrpnode -f**, only the remaining quorum nodes (as opposed to all nodes) are converted to being tiebreaker nodes.

If the **-f** flag is not specified when this command is run:

- more than half of the quorum nodes must be online to remove one or more nodes from the domain
- an error is returned if the peer domain has no remaining tiebreaker nodes as a result

See the *Administering RSCT* for more information about quorum nodes and tiebreaker nodes.

Flags

-f

Forces the specified nodes to be removed from the peer domain.

When the last tiebreaker node is removed using this flag, only the remaining quorum nodes (as opposed to all nodes) are converted to being tiebreaker nodes.

See the *Administering RSCT* for more information about quorum nodes and tiebreaker nodes.

-q

Specifies quiet mode. The command does not return an error if the specified nodes are not in the peer domain.

-F { file_name | "-" }

Reads a list of node names from *file_name*. Each line of the file is scanned for one node name. The pound sign (#) indicates that the remainder of the line (or the entire line if the # is in column 1) is a comment.

Use **-F " - "** to specify STDIN as the input file.

-h

Writes the command's usage statement to standard output.

-T

Writes the command's trace messages to standard error. For your software service organization's use only.

-V

Writes the command's verbose messages to standard output.

Parameters

node_name1 [node_name2 ...]

Specifies the peer domain node names of the nodes to be removed from the peer domain definition. You can remove one or more nodes using the **rmrpnode** command. You must specify the node names in exactly the same format as they were specified with the **addrpnode** command or the **mkrpdomain** command. To list the peer domain node names, run the **lsrpnode** command.

Security

The user of the **rmrpnode** command needs write permission for the **IBM.PeerNode** resource class on each node that is to be removed from the peer domain. By default, **root** on any node in the peer domain has read and write access to this resource class through the configuration resource manager.

Exit Status

0

The command ran successfully.

1

An error occurred with RMC.

- 2 An error occurred with a command-line interface script.
- 3 An incorrect flag was entered on the command line.
- 4 An incorrect parameter was entered on the command line.
- 5 An error occurred that was based on incorrect command-line input.
- 6 The node does not exist in the peer domain.

Environment Variables

CT_CONTACT

Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When CT_CONTACT is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If CT_CONTACT is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. CT_IP_AUTHENT only has meaning if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

Restrictions

This command must be run on a node that is online in the peer domain in which the nodes are to be removed. The nodes to be removed must also be offline to the peer domain.

Implementation Specifics

This command is part of the **rsct.basic.rte** fileset for AIX®.

Standard Input

When the -F "-" flag is specified, this command reads one or more node names from standard input.

Standard Output

When the -h flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

To remove the peer domain definitions of nodes nodeB and nodeC from the peer domain App1Domain, when nodeA is defined and online to App1Domain, and nodeB and nodeC are reachable from nodeA, run this command from nodeA:

```
rmnode nodeB nodeC
```

Location

/opt/rsct/bin/rmrpnode

rmrset Command

Purpose

Remove an rset from the system registry.

Syntax

```
rmrset rsetname
```

Description

The **rmrset** command removes an rset or exclusive rset (xrset) from the system registry. When used to delete an xrset, the **rmrset** command changes the state of the corresponding CPUs on the system to general use mode. Deleting an xrset requires root privilege.

Parameters

Item	Description
<i>rsetname</i>	The name of the rset to be removed from the system registry. The name consists of a <i>namespace</i> and an <i>rname</i> separated by a "/" (slash). Both the <i>namespace</i> and <i>rname</i> may contain up to 255 characters. See the rs_registername() service for additional information about character set limits of rset names.

Security

The user must have **root** authority, or **CAP_NUMA_ATTACH** capability and write access permission to the specified rset.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To remove **test/cpus0to7** from system registry, type:

```
rmrset test/cpus0to7
```

Files

Item	Description
/usr/bin/rmrset	Contains the rmrset command.

rmrsrc Command

Purpose

Removes a defined resource.

Syntax

To remove one or more resource.

- entered on the command line:

```
rmsrc -s "selection_string" [ -a | -N { node_file | " - " } ] [-h] [-TV] resource_class
```

```
rmsrc -r "resource_handle" [-h] [-TV]
```

- predefined in an input file:

```
rmsrc -f resource_data_input_file -s "selection_string" [ -a | -N { node_file | " - " } ] [-h] [-TV]  
resource_class
```

```
rmsrc -f resource_data_input_file -r "resource_handle" [-h] [-TV]
```

To display the names and datatypes of the command arguments:

```
rmsrc -l [-h] resource_class
```

Description

The **rmsrc** command removes — or "undefines" — the specified resource instance (or instances). The **rmsrc** command makes a request to the resource monitoring and control (RMC) subsystem to undefine a specific resource instance. The resource manager of the resource removes the resource.

The first format of this command requires a resource class name parameter and a selection string specified using the **-s** flag. All resources in the specified resource class that match the specified selection string are removed. If the selection string identifies more than one resource to be removed, it is the same as running this command once for each resource that matches the selection string.

The second format of this command allows the actual resource handle linked with a specific resource to be specified as the parameter. It is expected that this form of the command would be more likely used from within a script.

Instead of specifying multiple node names in *selection_string*, you can use the **-N *node_file*** flag to indicate that the node names are in a file. Use **-N " - "** to read the node names from standard input.

If Cluster Systems Management (CSM) is installed on your system, you can use CSM defined node groups as node name values to refer to more than one node. For information about working with CSM node groups and using the CSM **nodegrp** command, see the *CSM: Administration Guide* and the *CSM: Command and Technical Reference*.

Flags

-a

Specifies that this command applies to all nodes in the cluster. The cluster scope is determined by the **CT_MANAGEMENT_SCOPE** environment variable. If it is not set, first the management domain scope is chosen if it exists, then the peer domain scope is chosen if it exists, and then local scope is chosen, until the scope is valid for the command. The command will run once for the first valid scope found. For example, if both a management and peer domain exist, **rmsrc -a** with **CT_MANAGEMENT_SCOPE** not set will apply to the management domain. In this case, to apply to the peer domain, set **CT_MANAGEMENT_SCOPE** to 2.

-f *resource_data_input_file*

Specifies the name of the file that contains resource argument information.

-l

Lists the command arguments and datatypes. Some resource managers accept additional arguments that are passed to the remove request. Use this flag to list any defined command arguments and the datatypes of the command argument values.

-N { *node_file* | " - " }

Specifies that node names are read from a file or from standard input. Use **-N *node_file*** to indicate that the node names are in a file.

- There is one node name per line in *node_file*
- A number sign (#) in column 1 indicates that the line is a comment
- Any blank characters before a node name are ignored
- Any characters after a node name are ignored

Use -N "-" to read the node names from standard input.

The CT_MANAGEMENT_SCOPE environment variable determines the scope of the cluster. If CT_MANAGEMENT_SCOPE is not set, management domain scope is chosen first (if a management domain exists), peer domain scope is chosen next (if a peer domain exists), and then local scope is chosen, until the scope is valid for the command. The command runs once for the first valid scope it finds. For example, if a management domain and a peer domain both exist and CT_MANAGEMENT_SCOPE is not set, this command applies to the management domain. If you want this command to apply to the peer domain, set CT_MANAGEMENT_SCOPE to 2.

-r "resource_handle"

Specifies a resource handle. The resource handle must be specified using the format: "0xnnnn 0xnnnn 0xnnnnnnnn 0xnnnnnnnn 0xnnnnnnnn 0xnnnnnnnn", where *n* is any valid hexadecimal digit. The resource handle uniquely identifies a particular resource instance that should be removed.

-s "selection_string"

Specifies a selection string. All selection strings must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
```

```
-s 'Name ?= "test"'
```

Only persistent attributes can be listed in a selection string. For information on how to specify selection strings, see the *RSCT: Administration Guide*.

-h

Writes the command's usage statement to standard output.

-T

Writes the command's trace messages to standard error. For your software service organization's use only.

-V

Writes the command's verbose messages to standard output.

Parameters

resource_class

Specifies the resource class name. The resource instances for this resource class that match the selection string criteria are removed.

Security

The user needs write permission for the *resource_class* specified in `rmisc` to run `rmisc`. Permissions are specified in the access control list (ACL) file on the contacted system. See the *RSCT: Administration Guide* for information about the ACL file and how to modify it.

Exit Status

0

The command has run successfully.

1

An error occurred with RMC.

- 2** An error occurred with the command-line interface (CLI) script.
- 3** An incorrect flag was specified on the command line.
- 4** An incorrect parameter was specified on the command line.
- 5** An error occurred with RMC that was based on incorrect command-line input.
- 6** No resources were found that match the selection string.

Environment Variables

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. CT_IP_AUTHENT only has meaning if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon to monitor and control the resources and resource classes. The management scope determines the set of possible target nodes where the resources and resource classes can be monitored and controlled. The valid values are:

- 0** Specifies *local* scope.
- 1** Specifies *local* scope.
- 2** Specifies *peer domain* scope.
- 3** Specifies *management domain* scope.

If this environment variable is *not* set, *local* scope is used.

Implementation Specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) fileset for AIX.

Standard Output

When the -h flag is specified, this command's usage statement is written to standard output.

The command output and all verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

1. To remove the resource with the Name c175n05 from resource class IBM.Host, enter:

```
rmrsrc -s 'Name == "c175n05"' IBM.Host
```

2. To remove the resource linked with resource handle: 0x4017 0x0001 0x00000000 0x0069684c 0x0d52332b3 0xf3f54b45, enter:

```
rmrsrc -r "0x4017 0x0001 0x00000000 0x0069684c 0x0d52332b3 0xf3f54b45"
```

3. To remove the resources named Test1 from IBM.Foo for certain nodes in the cluster, using the /tmp/common/node_file file:

```
# common node file
#
node1.ibm.com      main node
node2.ibm.com      main node
node4.ibm.com      backup node
node6.ibm.com      backup node
#
```

as input, enter:

```
rmrsrc -s 'Name == "Test1"' -N /tmp/common/node_file IBM.Foo
```

Location

/opt/rsct/bin/rmrsrc

rmsecattr Command

Purpose

Removes the definition of the security attributes for a command, a device, a privileged file, or a domain-assigned object in the database.

Syntax

```
rmsecattr [-R load_module] { -c | -d | -f | -o } Name
```

Description

The **rmsecattr** command removes the security attributes for a command, a device, a file entry, or a domain-assigned object that is identified by the *Name* parameter from the appropriate database. The command interprets the *Name* parameter as a command, device, file entry, or domain-assigned object based on whether the **-c** (command), **-d** (device), **-f** (privileged file), or **-o** (domain-assigned object) flag is specified. If the **-c** flag is specified, the *Name* parameter must include the full path to the command and the command must at that time have an entry in the **/etc/security/privcmds** privileged command database.

If you specify the **-d** flag, the *Name* parameter must include the full path to the device and the device must at that time have an entry in the **/etc/security/privdevs** privileged device database.

If you specify the **-f** flag, the *Name* parameter must include the full path to the file and the file must have an entry in the **/etc/security/privfiles** privileged file database.

If you specify the **-o** flag, the *Name* parameter must include the full path if the object type is file or device and it must have an entry in the **/etc/security/domobjs** domain-assigned object database.

Important: The **rmsecattr** command removes only the definition of its security attributes; it does not remove the actual command, device, or file.

If the system is configured to use databases from multiple domains, the **rmsecattr** command finds the first match from the database domains in the order that was specified by the **secorder** attribute of the corresponding database stanza in the **/etc/nscontrol.conf** file. Meanwhile, the **rmsecattr** command removes that command or device entry from the domain. If any matching entries from the rest of the domains exist, they are not affected. Use the **-R** flag to remove an entry from a specific domain.

Modifications made by this command are not used for the security considerations until the databases are sent to the kernel security tables using the **setkst** command.

Flags

Item	Description
-c	Specifies, when used with the <i>Name</i> parameter, the full paths to one or more commands on the system that have entries in the privileged command database.
-d	Specifies, when used with the <i>Name</i> parameter, the full paths to one or more devices on the system that have entries in the privileged device database.
-f	Specifies, when used with the <i>Name</i> parameter, the full path to a privileged file on the system.
-o	Specifies, when used with the <i>Name</i> parameter, an object as specified in the domain-assigned object database.
-R load_module	Specifies the loadable module to use for the deletion of the <i>Name</i> entry.

Parameters

Item	Description
<i>Name</i>	The object to modify. The <i>Name</i> parameter is interpreted according to the -c , -d , -f , or -o flags that you specified.

Security

The **rmsecattr** command is a privileged command. It is owned by the root user and the security group, with mode set to 755. You must have at least one of the following authorizations to run the command:

Item	Description
aix.security.cmd.remove	Required to remove the security attributes of a command with the -c flag.
aix.security.device.remove	Required to remove the security attributes of a device with the -d flag.
aix.security.dobject.remove	Required to remove the security attributes of a domain-assigned object with the -o flag.
aix.security.file.remove	Required to remove the security attributes of a file with the -f flag.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

File Accessed

File	Mode
/etc/security/domobjs	rw
/etc/security/privcmds	rw
/etc/security/privdevs	rw
/etc/security/privfiles	rw

Examples

1. To remove the `/usr/sbin/mytest` command from the privileged command database, type:

```
rmsecattr -c /usr/sbin/mytest
```

2. To remove the `/dev/mydev` device from the privileged device database, type:

```
rmsecattr -d /dev/mydev
```

3. To remove the `/dev/mydev` device from the privileged device database in LDAP, type:

```
rmsecattr -R LDAP -d /dev/mydev
```

4. To remove the **/etc/testconf** file from the privileged file database, type:

```
rmsecattr -f /etc/testconf
```

5. To remove the network interface `en0` from the domained object database, type:

```
rmsecattr -o objecttype=netint en0
```

rm sensor Command

Purpose

Removes a sensor or a microsensor from the resource monitoring and control (RMC) subsystem.

Syntax

```
rm sensor [ -m ] [ -a | -n host1[, host2...] ] -N { node_file | "-" } [ -h ] [ -v | -V ] sensor_name1  
[sensor_name2...]
```

Description

The **rm sensor** command removes one or more sensors from the **IBM.Sensor** resource class or one or more microsensors from the **IBM.MicroSensor** resource class in the RMC subsystem. Use the **-m** flag to remove a microsensor.

If the sensor or microsensor is being monitored, monitoring will be stopped, but the event response resource manager (ERRM) resources defined for monitoring are not removed. To remove the ERRM resources, use the **rm condition**, **rm response**, or **rm condresp** command against the monitoring resources that were used for this sensor or microsensor.

The **rm sensor** command runs on any node. If you want **rm sensor** to run on all of the nodes in a domain, use the **-a** flag. If you want **rm sensor** to run on a subset of nodes in a domain, use the **-n** flag. Instead of specifying multiple node names using the **-n** flag, you can use the **-N node_file** flag to indicate that the node names are in a file. Use **-N "-"** to read the node names from standard input.

If Cluster Systems Management (CSM) is installed on your system, you can use CSM defined node groups as node name values to refer to more than one node. For information about working with CSM node groups and using the CSM **nodegrp** command, see the *CSM: Administration Guide* and the *CSM: Command and Technical Reference*.

Flags

-a

Removes sensors that match the specified name on all nodes in the domain.

The CT_MANAGEMENT_SCOPE environment variable determines the cluster scope. If CT_MANAGEMENT_SCOPE is not set, first the management domain scope is chosen if it exists, then the peer domain scope is chosen if it exists, and then local scope is chosen, until the scope is valid for the command. The command will run once for the first valid scope found. For example, if both a management domain and a peer domain exist, `rmsensor -a` with CT_MANAGEMENT_SCOPE not set will run in the management domain. In this case, to run in the peer domain, set CT_MANAGEMENT_SCOPE to 2.

-m

Specifies that the resources to be removed are microsensor resources.

-h

Writes the command's usage statement to standard output.

-n host1[,host2...]

Specifies the node from which the sensor should be removed. By default, the sensor is removed from the local node. This flag is only appropriate in a management domain or a peer domain.

-N {node_file | "-"}]

Specifies a file or standard input listing the nodes on which the sensor must be removed. This flag is only appropriate in a Cluster Systems Management (CSM) or a peer domain cluster.

-v | -V

Writes the command's verbose messages to standard output.

Parameters

sensor_name1 [sensor_name2...]

Specifies one or more names of sensors to remove.

Security

To remove sensors using this command, you need write permission for the **IBM.Sensor** resource class. To remove microsensors using this command, you need write permission for the **IBM.MicroSensor** resource class. Permissions are specified in the access control list (ACL) file on the contacted system. See the *RSCT: Administration Guide* for details on the ACL file and how to modify it.

Exit Status

0

The command has run successfully.

1

An incorrect combination of flags and parameters has been entered.

6

No sensor resources were found.

n

Based on other errors that can be returned by the RMC subsystem.

Environment Variables

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the resource monitoring and control (RMC) daemon on the specified host. If this environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. CT_IP_AUTHENT only has meaning if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the the session with the RMC daemon to monitor and control the resources and resource classes. The management scope determines the set of possible target nodes where the resources and resource classes can be monitored and controlled.

The valid values are:

0

Specifies *local* scope.

1

Specifies *local* scope.

2

Specifies *peer domain* scope.

3

Specifies *management domain* scope.

If this environment variable is *not* set, *local* scope is used.

Implementation Specifics

This command is part of the rsct.core fileset for AIX®.

Examples

1. To remove the sensor **sensor1**, enter:

```
rm sensor sensor1
```

2. To remove the sensor called **sensor1** from the nodes that are listed in the **/u/joe/common_nodes** file, enter:

```
rm sensor -N /u/joe/common_nodes sensor1
```

where **/u/joe/common_nodes** contains:

```
# common node file
#
node1.myhost.com    main node
node2.myhost.com    backup node
```

3. To remove the microsensor called **IBM.usensor1**, enter:

```
rm sensor -m IBM.usensor1
```

Location

/opt/rsct/bin/rmsensor

rmserver Command

Purpose

Removes a subserver definition from the Subserver Type object class.

Syntax

rmserver **-t** *Type*

Description

The **rmserver** command removes an existing subserver definition from the Subserver Type object class.

Flags

Item	Description
-t <i>Type</i>	Specifies the subserver name that uniquely identifies the existing subserver to be removed. The rmserver command is unsuccessful if the <i>Type</i> name is not known in the Subserver Type object class.

Security

Auditing Events: If the auditing subsystem has been properly configured and is enabled, the **rmserver** command will generate the following audit record (event) every time the command is executed:

Event	Information
SRC_Delserver	Lists in an audit log the name of the subserver definition that was deleted.

See html

>lrmsmbcmnt Command

Purpose

Removes a Server Message Block (SMB) client file system entry from the `/etc/filesystems` file and unmounts the SMB client file system if the SMB client file system is mounted.

Syntax

```
lrmsmbcmnt -f MountPoint [-B | -N | -I]
```

Description

The **lrmsmbcmnt** command removes an SMB client file system entry from the `/etc/filesystems` file. If the SMB client file system is mounted, the **lrmsmbcmnt** command unmounts it.

Flags

- B**
Removes the corresponding SMB client file system entry from the `/etc/filesystems` file, and unmounts the SMB client file system. This is the default setting.
- f** *MountPoint*
Specifies the path name of the SMB client mount point.

-I

Does not unmount the file system, but removes the entry from `/etc/filesystems` file, only if the filesystem is already unmounted.

-N

Unmounts the SMB client file system, but does not remove the SMB entry from the `/etc/filesystems` file.

Exit status**0**

The command completed successfully.

>0

An error occurred.

Example

To remove the SMB entry that is mounted over the `/mnt` mount point, and to unmount the SMB client file system, enter the following command:

```
rmsmbcmnt -f /mnt
```

Location

`/usr/sbin/rmsmbcmnt`

Files**`/etc/filesystems`**

Stores the SMB client file system entry that contains the SMB client mount points.

[<](#)

rmsmbcred Command

Purpose

Removes the Server Message Block (SMB) client file system credentials that are stored in the `/etc/smbcred` file for the specified server and user entry.

Syntax

```
rmsmbcred -s server_name -u user_name
```

Description

You must specify the server name and the username in the **rmsmbcred** command. If the specified server-user entry is found in the credentials that are listed in the `/etc/smbcred` file, the corresponding credential entry is removed. After the credential entry is removed, if you want to mount the SMB client file system on the specified server for the specified user, you must manually specify the password in the specified server.

Flags**-s *server_name***

Specifies the name of the remote host, which is an SMB server. The *server_name* parameter can be provided as a hostname, an IP address, or a fully qualified domain name.

-u *user_name*

Specifies the username for which credentials must be removed from the `/etc/smbcred` file.

Exit status

0

The command completed successfully.

>0

An error occurred.

Example

To remove the credential that is stored in `/etc/smbcred` file for `user1` to mount the SMB client file system on the `xxx.in.ibm.com` server, enter the following command:

```
rmsmbcred -s xxx.in.ibm.com -u user1
```

Location

`/usr/sbin/rmsmbcred`

Files

`/etc/smbcred`

Stores the credentials of the SMB client file system.

rmsock Command

Purpose

Removes a socket that does not have a file descriptor.

Syntax

rmsock *Address TypeofAddress*

Description

The **rmsock** command removes a socket that does not have a file descriptor. It accepts a `socket`, `tcpcb`, `inpcb`, `ripcb`, or `rawcb` address and converts it to a socket address. All opened files in every process are then checked to find a match to the socket. If a match is not found, an abort action is performed on that socket regardless of the existence of the socket **linger** option. The port number held by the socket is released. If a match is found, its file descriptor and status of the owner process are displayed to the user. The results are passed to **syslogd** and recorded in the `/var/adm/ras/rmsock.log` file.

If the socket to be removed is not held by any active processes, but there are processes in the exiting state, **rmsock** will not remove the socket specified because the socket could be held by the processes in the exiting state. Any socket that is held by the exiting processes will be cleaned up when those processes exit completely.

Examples

1. To remove a socket from its socket address, type:

```
rmsock 70054edc socket
```

You do not need to specify the type of the socket. It can be a `tcpcb`, `udp`, `raw`, or routing socket.

2. To remove a socket from its inpcb address, type:

```
rmsock 70054edc inpcb
```

3. To remove a socket from its tcpcb address, type:

```
rmsock 70054ecc tcpcb
```

Files

Item	Description
<code>/usr/sbin</code>	Directory where the rmsock command resides.
<code>/var/adm/ras/rmsock.log</code>	Contains the rmsock.log file.

rmss Command

Purpose

Simulates a system with various sizes of memory for performance testing of applications.

Syntax

rmss -c *MemSize*

rmss -r

rmss -p

rmss [-d *MemSize*] [-f *MemSize*] [-n *NumIterations*] [-o *OutputFile*] [-s *MemSize*] *Command*

Description

The **rmss** command simulates a system with various sizes of real memory, without having to extract and replace memory boards. By running an application at several memory sizes and collecting performance statistics, you can determine the memory needed to run an application with acceptable performance. The **rmss** command can be invoked for either of two purposes:

- To change the memory size and then exit, using the -c , -p , and -r flags. This lets you experiment freely with a given memory size.
- To function as a driver program, using the -s , -f , -d , -n , and -o flags. In this mode, the **rmss** command executes a specified command multiple times over a range of memory sizes, and displays important statistics describing command performance at each memory size. The command can be an executable or shell script file, with or without command line arguments.

The **rmss** command is incompatible with the DR subsystem. If a DR event occurs during the **rmss** command execution, the system can hang. Since the memory removal function of the **rmss** command can be replaced by DR memory removal with the **drmgr** command, the information text of the **rmss** command must be amended with this warning:



Attention: The **rmss** command is incompatible with the AIX DLPAR component, and its usage may result in a hung system. The **drmgr** command provides a safe memory removal function in a DLPAR environment.



Attention: When **rmss** is used on a multiple memory pool system, it may fail with:

```
Failure: VMM unable to free enough frames for stealing.  
Choose a larger memory size or retry with less system activity.
```

Or a similar message. This failure can occur when **rmss** has stolen all the frames from a memory pool, and is unable to steal frames from other pools. A workaround is to decrease memory by increments.

The number and size of memory pools on a system can be retrieved with the command:

```
echo "mempool *" | kdb
```

The **-c**, **-p**, and **-r** flags are mutually exclusive. The **-c** flag changes the memory size; the **-p** flag displays the current memory size; and the **-r** flag resets the memory size to the real memory size of the machine.

The **-s**, **-f**, **-d**, **-n**, and **-o** flags are used in combination when the **rmss** command is invoked as a driver program to execute and measure the performance of a command (where a command is an executable or a shell script file) over a range of memory sizes. When invoked this way, the **rmss** command displays performance statistics, such as the response time of the command and the number of page-ins that occurred while the command ran, for each memory size. These statistics, which are also written to a file, are described in this [example](#).

The **-s** and **-f** flags specify the starting and ending points of the range, while the **-d** flag specifies the increment or decrement between memory sizes within the range. The **-n** flag is used to specify the number of times to run the command at each memory size, and the **-o** flag is used to specify the name of an output file into which to write the **rmss** report. The *Command* parameter specifies the command to be run and measured at each memory size.

Note:

1. The **rmss** command reports “usable” real memory. On machines where there is bad memory or where the system is using the memory, **rmss** reports the amount of real memory as the amount of physical real memory minus the memory that is bad or in use by the system. For example, using the **rmss -r** flag might report:

```
Simulated Memory Size changed to 79.9062MB
```

This could be a result of some pages being marked bad or a result of a device that is reserving some pages for its own use (and thus not available to the user).

2. The **rmss** command may underestimate the number of page-ins that are required to run an application if the application, combined with background processes such as daemons, accesses a lot of different files (including directory files). The number of different files that must be accessed to cause such results is approximately 250 files per 8MB of simulated memory size. The following table gives the approximate number of different files that, when accessed at the given simulated memory size, may result in the **rmss** command underestimating page-in requirements.

Simulated Memory Size (MB)	Access to Different Files
8	250
16	500
24	750
32	1000
48	1500
64	2000
128	4000
256	8000

You can use the **filemon** command to determine the number of files accessed while your command runs, if you suspect that it may be accessing many different files.

Flags

Item	Description
-c <i>MemSize</i>	Changes the simulated memory size to the <i>MemSize</i> value, which is an integer or decimal fraction in units of megabytes. The <i>MemSize</i> variable must be between 8MB and the real memory size of the machine. There is no default for the -c flag. Note: It is difficult to change the simulated memory size to less than 8MB, because of the size of inherent system structures such as the kernel.
-d <i>MemSize</i>	Specifies the increment or decrement between memory sizes to be simulated. The <i>MemSize</i> value is an integer or decimal fraction in units of megabytes. If the -d flag is omitted, the increment or decrement will be 8MB.
-f <i>MemSize</i>	Specifies the final memory size. You should finish testing the simulated system by executing the command being tested at a simulated memory size given by the <i>MemSize</i> variable, which is an integer or decimal fraction in units of megabytes. The <i>MemSize</i> variable must be between 4MB and the real memory size of the machine. If the -f flag is omitted, the final memory size will be 8MB. Note: It is difficult to finish at a simulated memory size of less than 8MB because of the size of inherent system structures such as the kernel.
-n <i>NumIterations</i>	Specifies the number of times to run and measure the command, at each memory size. There is no default for the -n flag. If the -n flag is omitted, during rmss command initialization, the rmss command will determine how many iterations of the command being tested are necessary to accumulate a total run time of 10 seconds, and then run the command that many times at each memory size. Note: The rmss command always executes the command once at each memory size prior to the executions that are measured. This prepares the simulation for the actual test.
-o <i>OutputFile</i>	Specifies the file into which to write the rmss report. If the -o flag is omitted, then the rmss report is written to the file rmss.out. In addition, the rmss report is always written to standard output.
-p	Displays the current simulated memory size.
-r	Resets the simulated memory size to the real memory size of the machine.
-s <i>MemSize</i>	Specifies the starting memory size. Start by executing the command at a simulated memory size specified by the <i>MemSize</i> variable, which is an integer or decimal fraction in units of megabytes. The <i>MemSize</i> variable must be between 4MB and the real memory size of the machine. If the -s flag is omitted, the starting memory size will be the real memory size of the machine. Note: It is difficult to start at a simulated memory size of less than 8MB, because of the size of inherent system structures such as the kernel.
<i>Command</i>	Specifies the command to be run and measured at each memory size. The <i>Command</i> parameter may be an executable or shell script file, with or without command line arguments. There is no default command.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Security

Access Control: You must have root authority to run this command.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To change the memory size to 13.5MB, enter:

```
rmss -c 13.5
```

2. To print the current memory size, enter:

```
rmss -p
```

3. To reset the memory size to the real memory size of the machine, enter:

```
rmss -r
```

4. To investigate the performance of the command `cc -O foo.c` on memory sizes 32, 24, 16, and 8MB; run and measure the command once at each memory size; and then write the report to the `cc.rmss.out` file, enter:

```
rmss -s 32 -f 8 -d 8 -n 1 -o cc.rmss.out cc -O foo.c
```

5. To investigate the performance of the sequence of commands in the `foo.sh` shell script file on memory sizes starting at the real memory size of the machine and ending at 8MB, by increments of 8MB; let the **rmss** command determine the number of iterations to run and measure the `foo.sh` at file each memory size; and then write the **rmss** report to the `rmss.out` file (with all defaults used in this invocation of the **rmss** command), enter the following:

```
rmss foo.sh
```

6. To investigate the performance of the executable `bar` on memory sizes from 8MB to 16MB, by increments of 0.5MB; run and measure `bar` twice at each memory size; and write the report to the `bar.rmss.out` file, enter:

```
rmss -s 8 -f 16 -d .5 -n 2 -o bar.rmss.out bar
```

7. When any combination of the **-s**, **-f**, **-d**, **-n**, and **-o** flags is used, the **rmss** command runs as a driver program, which executes a command multiple times over a range of memory sizes, and displays statistics describing the command's performance at each memory size.

An example of the report printed out by the **rmss** command follows:

```

Hostname: xray.austin.ibm.com
Real memory size: 48.00 Mb
Time of day: Wed Aug 8 13:07:33 1990
Command: cc -O foo.c
Simulated memory size initialized to 24.00 Mb.
Number of iterations per memory size = 1 warmup + 1 measured = 2.
Memory size Avg. Pageins Avg. Response Time Avg. Pagein Rate
(megabytes) (sec.) (pageins/sec.)
-----
24.00 0.0 113.7 0.0

```

22.00	5.0	114.8	0.0
20.00	0.0	113.7	0.0
18.00	3.0	114.3	0.0
16.00	0.0	114.6	0.0
14.00	139.0	116.1	1.2
12.00	816.0	126.9	6.4
10.00	1246.0	135.7	9.2
8.00	2218.0	162.9	13.6

This report was generated by the following command:

```
rmss -s 24 -f 8 -d 2 -n 1 cc -O foo.c
```

The first part of the report gives general information, including the machine that the **rmss** command was running on, the real memory size of that machine, the time and date, and the command that was being measured. The next two lines give informational messages that describe the initialization of the **rmss** command. Here, the **rmss** command displays that it has initialized the simulated memory size to 24MB, which was the starting memory size given with the **-s** flag. Also, the **rmss** command prints out the number of iterations that the command will be run at each memory size. The command is to be run twice at each memory size: once to warmup, and once when its performance is measured. The number of iterations was specified by the **-n** flag.

The next part of the report provides the following for each memory size the command was run at:

- The memory size, along with the average number of page-ins that occurred while the command was run
- The average response time of the command
- The average page-in rate that occurred when the command was run.

Note: The average page-ins and average page-in rate values include all page-ins that occurred while the command was run, not just those initiated by the command.

Files

Item	Description
<code>/usr/bin/rmss</code>	Contains the rmss command.

rmssys Command

Purpose

Removes a subsystem definition from the subsystem object class.

Syntax

rmssys **-s** *Subsystem*

Description

The **rmssys** command removes an existing subsystem definition from the subsystem object class. It also removes any subservers and notify method definitions that exist for the subsystem being removed.

Flags

Item	Description
-s <i>Subsystem</i>	Specifies the name that uniquely identifies the subsystem to be removed. The rmssys command is unsuccessful if the subsystem name is not known in the subsystem object class. The rmssys command removes any subserver definitions from the Subserver Type object class that are defined for this subsystem, as well as any notify method definitions from the Notify object class that are defined for this subsystem.

Security

For details about selecting and grouping audit events, and configuring audit event data collection, see [Setting Up Auditing in Security](#) topic.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Files

Item	Description
/etc/objrepos/SRCsubsys	Specifies the SRC Subsystem Configuration object class.
/etc/objrepos/SRCsubsvr	Specifies the SRC Subserver Configuration object class.
/etc/objrepos/SRCnotify	Specifies the SRC Notify Method object class.
/dev/SRC	Specifies the AF_UNIX socket file.
/dev/.SRC-unix	Specifies the location for temporary socket files.

rmt Command

Purpose

Allows remote access to magnetic tape devices.

Syntax

rmt

Description

The **rmt** command allows remote access to magnetic tape devices. The remote dump and restore programs use the **rmt** command as a remote magnetic tape protocol module. The **rmt** command is normally started with a **rexec** or **rcmd** subroutine.

The **rmt** command accepts requests specific to the manipulation of magnetic tapes, performs the commands, and then responds with a status indication. All responses are in ASCII and in one of two forms. Successful commands receive responses of Axxx, where xxx is an ASCII representation of a decimal number. Unsuccessful commands receive responses of Eyyy error-message, where yyy is one of the possible error numbers described in the **errno.h** file and error-message is the corresponding error string as printed from a call to the **perrot** subroutine. The protocol is comprised of the following subcommands.

Subcommands

Item	Description
O <i>DeviceMode</i>	Opens the device specified by the <i>Device</i> parameter using the mode indicated by the <i>Mode</i> parameter. The value of the <i>Device</i> parameter is a full path name, and that of the <i>Mode</i> parameter is an ASCII representation of a decimal number suitable for passing to the open subroutine. An open device is closed before a new open operation is performed.
C <i>Device</i>	Closes the open device. The device specified with the <i>Device</i> parameter is ignored.
L <i>WhenceOffset</i>	Performs an lseek operation using the specified parameters. The lseek subroutine returns the response value.
W <i>Count</i>	Writes data onto the open device. From the connection, the rmt command reads the number of bytes specified by the <i>Count</i> parameter, ending if a premature end-of-file is encountered. The write subroutine returns the response value.
R <i>Count</i>	Reads, from the open device, the number of bytes of data specified by the <i>Count</i> parameter. The rmt command then performs the requested read operation and responds with Azzz, where zzz is the number of bytes read if the operation was successful. The data read is then sent. Otherwise, an error in the standard format is returned.
I <i>OperationCount</i>	Performs an STIOCTOP ioctl subroutine using the specified parameters. The parameters are interpreted as the ASCII representations of the decimal values to place in the <i>mt_op</i> and <i>mt_count</i> fields of the structure used in the ioctl subroutine. The return value is the value of the <i>Count</i> parameter when the operation is successful.

Any other subcommand causes the **rmt** command to exit.

Note: For the **R** and **W** subcommands, if the *Count* parameter specifies more bytes than the connection can handle, the data will be truncated to a size that can be handled.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Files

Item	Description
/usr/sbin/rmt	Contains the rmt command.
/usr/include/sys/errno.h	Describes the possible error numbers.

rmtcpip Command

Purpose

Removes the TCP/IP configuration for a host machine.

Syntax

rmtcpip

Description

The **rmtcpip** command removes TCP/IP configuration on a host machine. The basic functions of this command is:

- Removes the network interface configurations
- Restores **/etc/rc.tcpip** to the initial installed state
- Restores **/etc/hosts** to the initial installed state
- Removes the **/etc/resolv.conf** file
- Removes the default and static routes
- Sets the hostname to localhost
- Sets the hostid to 127.0.0.1
- Resets configuration database to the initial installed state

Note:

1. Any daemon which is commented out by default in **/etc/rc.tcpip**, but running at the time this command is issued, is stopped.
2. Your version of the **/etc/hosts** file is saved as **/etc/hosts.save** prior to the **/etc/hosts** file being restored to the originally installed state.
3. Your version of the **/etc/resolv.conf** file is saved as **/etc/resolv.conf.save** prior to the removal of the **/etc/resolv.conf** file.

Security

This command can only be run by root.

rmts Command

Purpose

Removes a thin server.

Syntax

rmts [-f] [-v] *ThinServer*

Description

The **rmts** command removes a thin server specified by *ThinServer* and created with the **mkts** command. If the thin server is running, the **rmts** command does not remove the thin server. Instead, it prints a message indicating that the thin server could not be removed. In this case, use the **-f** flag to terminate the thin server's session with a common image.

Flags

Item	Description
-f	Forces the removal of the thin server if the thin server is up and running.
-v	Enables verbose debug output when the rmts command runs.

Exit Status

Item	Description
0	The command completed successfully.
>0	An error occurred.

Security

Access Control: You must have root authority to run the `rmts` command.

Examples

1. To remove a thin server named `lobo`, enter:

```
rmts lobo
```

Location

`/usr/sbin/rmts`

Files

Item	Description
<code>/etc/niminfo</code>	Contains variables used by NIM.

rmtun Command

Purpose

Deactivates operational tunnel(s) and optionally removes tunnel definition(s).

Syntax

```
rmtun -v 4|6 -t tid_list | all [-d]
```

Description

Use the **rmtun** command to deactivate an active tunnel(s) and optionally remove tunnel definition(s). It also will remove the auto-generated filter rules created for the tunnel by the **gentun** command when the tunnel definition is removed from the tunnel database.

Flags

Item	Description
all	Deactivates and optionally removes all the tunnel(s).
tid_list	The list of the tunnel(s) you want to deactivate. The tunnel IDs can be separated by ";" or "-". You can use "-" to specify a range of IDs. For example, 1,3,5-7 specified there are five tunnel IDs in the list, 1, 3, 5, 6 and 7.
-d	Specifies that the tunnels are to be removed from the tunnel database. This is an optional flag.

Item	Description
-t	The list of the tunnel(s) you want to deactivate. If -d is specified, all the tunnel definitions in the list will also be removed from the tunnel database.
-v	The IP version of the tunnel. For the IP version 4 tunnel, use the value of 4 . For the IP version 6 tunnel, use the value of 6 .

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

rmusil Command

Purpose

Removes an existing user-specified installation location (USIL) instance.

Syntax

rmusil **-R** *RelocatePath* **-r**

Description

The **rmusil** command removes an existing USIL instance.

Flags

Item	Description
-r	Removes the Software Vital Product Data (SWVPD) of an USIL instance.
-R <i>RelocatePath</i>	The path to an existing USIL location.

Note: The **rmusil** command only removes the USIL reference in the SWVPD. No files are removed in the USIL installation path.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Files

Item	Description
/usr/sbin/rmusil	Contains the rmusil command.

rmuser Command

Purpose

Removes a user account.

Syntax

rmuser [**-R** *load_module*] [**-c**] [**-p**] *Name*

Description

The **rmuser** command removes the user account that is identified by the *Name* parameter. This command removes a user account's attributes without removing the user's home directory and files. The user name must exist. If you specify the **-c** flag, the **rmuser** command checks whether the user is logged in or has running processes before removing the user account. If the user is logged in or has running processes, the **rmuser** command fails. If you specify the **-p** flag, the **rmuser** command also removes passwords and other user authentication information from the `/etc/security/passwd` file.

For user accounts that are created with an alternate Identification and Authentication (I&A) mechanism, use the **-R** flag with the appropriate load module to remove that user. The load modules are defined in the `/usr/lib/security/methods.cfg` file.

Only the root user or users with UserAdmin authorization can remove administrative users. Administrative users are those users with **admin=true** set in the `/etc/security/user` file.

You can also use the System Management Interface Tool (SMIT) **smit rmuser** fast path to run this command.

Flags

Item	Description
-c	Verifies that the user is not logged in and does not have running processes before removing the user account.
-p	Removes user password information from the <code>/etc/security/passwd</code> file and removes the user keystore.
-R <i>load_module</i>	Specifies the loadable I&A module that is used to remove the user account.

Parameter

Item	Description
<i>Name</i>	Specifies a user account.

Exit Status

This command returns the following exit values:

Item	Description
0	The command ran successfully and all requested changes are made.
>0	An error occurred. The printed error message gives further details about the type of failure.

Security

Access Control: This command should grant execute (x) access only to the root user and members of the security group. This command should be installed as a program in the trusted computing base (TCB). The command should be owned by the root user with the **setuid** (SUID) bit set.

Files Accessed:

Mode	File
rw	/etc/passwd
rw	/etc/security/passwd
rw	/etc/security/user
rw	/etc/security/user.roles
rw	/etc/security/limits
rw	/etc/security/envIRON
rw	/etc/security/audit/config
rw	/etc/group
rw	/etc/security/group
> > rw	/etc/security/uattr < <

Auditing Events:

Event	Information
USER_Remove	user

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To remove the user account `davis` and its attributes from the local system, enter:

```
rmuser davis
```

2. To remove the user account `davis` and all its attributes, including passwords and other user authentication information in the `/etc/security/passwd` file, type:

```
rmuser -p davis
```

3. To remove the user account `davis`, who was created with the LDAP load module, type:

```
rmuser -R LDAP davis
```

Files

Item	Description
<code>/usr/sbin/rmuser</code>	Contains the rmuser command.
<code>/etc/security/passwd</code>	Contains password information.
<code>/etc/security/user</code>	Contains the extended attributes of user accounts.

Item	Description
<code>/etc/security/enviro</code>	Contains environment attributes of user accounts.
<code>/etc/group</code>	Contains the basic attributes of groups.
<code>> > /etc/security/uattr</code>	Contains further user attributes. < <

rmvfs Command

Purpose

Removes entries in the `/etc/vfs` file. The *VfsName* parameter is the name of a virtual file system. The **rmvfs** command takes one argument, the name of the virtual file system type to be removed from the file. If this *VfsName* entry exists, it is removed from the file.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

To remove the `newvfs` entry, enter:

```
rmvfs newvfs
```

Files

Item	Description
<code>/etc/vfs</code>	Contains descriptions of virtual file system types.

rmvirprt Command

Purpose

Removes a virtual printer.

Syntax

```
rmvirprt -q PrinterQueueName -d QueueDeviceName
```

Description

The **rmvirprt** command removes the virtual printer assigned to the *PrinterQueueName* and *QueueDeviceName* variable value. The **rmvirprt** command also removes the System Management Interface Tool (SMIT) Object Database Manager (ODM) objects associated with the specified queue and queue device.

You can also use the System Management Interface Tool (SMIT) **smit rmvirprt** fast path to run this command.

Note: When the command **rmvirprt** is run from the command line, it does not remove the queue or queue device, nor does it check for any jobs running or queued on the specified queue and queue device. However, if SMIT is used to run this command interactively, the corresponding queue, queue device,

and, optionally, printer device, are removed along with the virtual printer, if there are no jobs running or queued.

Flags

Item	Description
-d <i>QueueDeviceName</i>	Specifies the name of the queue device to which the virtual printer is assigned.
-q <i>PrinterQueueName</i>	Specifies the name of the print queue to which the virtual printer is assigned.

Examples

To remove the attribute values for the mypro virtual printer associated with the proq print queue, type:

```
rmvirprt -d mypro -q proq
```

Files

Item	Description
<u>/etc/qconfig</u>	Contains the configuration file.
<u>/usr/sbin/rmvirprt</u>	Contains the rmvirprt command.
<u>/var/spool/lpd/pio/@local/custom/*</u>	Contains the customized virtual printer attribute files.
<u>/var/spool/lpd/pio/@local/ddi/*</u>	Contains the digested virtual printer attribute files.

rmwpar Command

Purpose

Removes a workload partition.

Syntax

```
/usr/sbin/rmwpar [ -F ] [ -p ] [ -s ] [ -v ] WparName
```

Description

The **rmwpar** command deletes the specified workload partition from the system that includes the following tasks:

- Removing the workload partition's configuration data from the system's workload partition database
- Deleting the workload partition's file systems (if you do not specify the **-p** flag)
- Removing the workload partition's Workload Manager (WLM) profile

Without the **-F** flag, the **rmwpar** command stops the first time any part of the operation fails. If you specify the **-F** flag, the **rmwpar** command removes as much as possible. If the specified workload partition is active, the **rmwpar** command fails unless you specify the **-s** flag or the **-F** flag.

Flags

Item	Description
-F	Specifies that the rmwpar command must override or ignore most failures. It can be used to force the removal of broken workload partitions. This flag implies the -s flag.
-p	Removes a preservation removal that is assigned for the workload partition. The configured local file systems that are the logical volumes or subdirectories within the pre-existing logical volumes are not emptied or removed. This flag is for system workload partitions only. This flag cannot be used with rootvg workload partitions. File systems that are preserved by using this flag can be used with the following command to create a new workload partition that is attached to the m: <pre>mkwpar -p</pre>
-S	Stops the workload partition. This flag is equivalent to calling the stopwpar command before the rmwpar command. Use this flag to shut down and delete a workload partition in 1 step. If the rmwpar command was run with the -F flag specified, the stopwpar command can be run with the -F flag specified. If the rmwpar command is run on an active workload partition without the -s flag or the -F flag that is specified, the rmwpar command fails.
-v	Verbose mode.

Security

Access Control: Only the root user can run this command.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To remove the workload partition called "roy", enter:

```
rmwpar roy
```

2. To stop and remove the workload partition called "roy", preserving data on its file system, enter:

```
rmwpar -p -s roy
```

rmyp Command

Purpose

Removes the configuration for NIS.

Syntax

`/usr/sbin/rmyp { -s | -c }`

Description

The **rmyp** command removes everything from the system that is used to make NIS work. For example, the **rmyp** command removes all of the NIS maps and all of the entries in the `/etc/rc.nfs` file for the NIS daemons.

You could also use the System Management Interface Tool (SMIT) **smit rmyp** fast path to run this command. You can use the **System management interface tool (SMIT)** to run this command. To use SMIT, enter:

```
smit rmyp
```

Flags

Item	Description

-s	Removes the server configuration from the system.
-c	Removes the client configuration from the system.

rndc Command

Purpose

Name server control utility.

Syntax

rndc [**-b** *source-address*] [**-c** *config-file*] [**-k** *key-file*] [**-s** *server*] [**-p** *port*] [**-q**] [**-r**] [**-V**] [**-y** *key_id*] [**-4**] [**-6**] { *command* }

Description

The **rndc** command controls the operation of a name server. If you run the **rndc** command with no command-line options or arguments, it prints a short summary of the supported commands and the available options and their arguments.

The **rndc** command communicates with the name server over a TCP connection, sending commands authenticated with digital signatures. In the current versions of the **rndc** command and the **named** daemon, following algorithms are the only supported authentication algorithm:

- HMAC-MD5 (for compatibility)
- HMAC-SHA1
- HMAC-SHA224
- HMAC-SHA256 (default)
- HMAC-SHA384
- HMAC-SHA512

The supported authentication algorithms use a shared secret on each end of the connection, which provides TSIG-style authentication for the command request and the response of the name server. A *key_id* that is known to the server must sign all the commands that are sent over the channel.

The **rndc** command reads a configuration file to determine how to contact the name server and decide what algorithm and key it must use.

Flags

Table 12. Flags

Item	Description
-4	Indicates use of IPv4 only.
-6	Indicates use of IPv6 only.
-b source-address	Uses the <i>source-address</i> value as the source address for the connection to the server. Multiple instances are permitted to allow setting of both the IPv4 and IPv6 source addresses.
-c config-file	Uses the <i>config-file</i> value as the configuration file instead of the default <code>/etc/rndc.conf</code> configuration file.
-k key-file	Uses the <i>key-file</i> value as the key file instead of the default <code>/etc/rndc.key</code> file. The key in the <code>/etc/rndc.key</code> file is used to authenticate commands that are sent to the server if the <i>config-file</i> argument does not exist.
-s server	Specifies the name or address of the server that matches a server statement in the configuration file for the rndc command. If you do not specify the <i>server</i> value, the host that is named by the default-server clause in the option statement of the configuration file is used.
-p port	Sends commands to TCP port instead of the default control channel port, 953.
-q	Sets quiet mode to avoid printing the message text that is returned by the server unless an error occurs.
-r	Prints the result code that is returned by the named utility after the named utility runs the requested command (for example, <code>ISC_R_SUCCESS</code> , <code>ISC_R_FAILURE</code>).
-v	Enables verbose logging.
-y key_id	Uses the <i>key_id</i> key from the configuration file. The <i>key_id</i> value must be known by the named daemon with the same algorithm and secret string for control message validation to succeed. If you do not specify the <i>key_id</i> value, the rndc command first looks for a key clause in the server statement of the server that is in use. If no server statement is present for that host, then the default-key clause of the options statement is used.

Note: The configuration file contains shared secrets that are used to send authenticated control commands to name servers. It cannot have general read or write access.

For the complete set of commands supported by the **rndc** command, see the [BIND 9 Administrator Reference Manual](#) or run the **rndc** command without arguments to see its help message.

Limitations

The **rndc** command works only with the **named9** daemon. The shared-secret for a *key_id* cannot be provided without using the configuration file.

rndc-confgen Command

Purpose

Generates configuration files for the **rndc** command.

Syntax

rndc-confgen [**-a**] [**-A** *algorithm*] [**-b** *keysize*] [**-c** *keyfile*] [**-h**] [**-k** *keyname*] [**-p** *port*] [**-q**] [**-s** *address*] [**-t** *chrootdir*] [**-u** *user*]

Description

The **rndc-confgen** command generates configuration files for the **rndc** command. You can use this command as an alternative to manually writing the `rndc.conf` file, the corresponding controls, and key statements in the **named.conf** command. You can run the **rndc-confgen** command with the **-a** flag to set up a `rndc.key` file to avoid the need for a `rndc.conf` file and a control statement.

Flags

Item	Description
-a	Performs automatic rndc command configuration. This creates a file <code>rndc.key</code> in <code>/etc</code> directory (or whatever <code>sysconfdir</code> directory was specified as when BIND was built) read by both the rndc command and the named daemon on startup. The <code>rndc.key</code> file defines a default command channel and authentication key that allows the rndc command to communicate with the named daemon on the local host with no further configuration.
-A <i>algorithm</i>	Specifies an algorithm that must be used for the transaction signatures (TSIG) key. You can specify any of the following values: • <code>hmac-md5</code> • <code>hmac-sha1</code> • <code>hmac-sha224</code> • <code>hmac-sha256</code> • <code>hmac-sha384</code> • <code>hmac-sha512</code> The default value is <code>hmac-sha256</code> .
-b <i>keysize</i>	Specifies the size of the authentication key in bits. The range for the value of the <i>keysize</i> is 1-512.
-c <i>keyfile</i>	Used with the -a flag to specify an alternative location for the <code>rndc.key</code> file name.
-h	Prints a short summary of the options and arguments of the rndc-confgen command.
-k <i>keyname</i>	Specifies the key name of the rndc command authentication key. The name must be a valid domain name. The default is the <code>rndc-key</code> file.
-p <i>port</i>	Specifies the command channel port where the named daemon listens for connections from rndc command authentication key. The default is 953 .
-q	Prevents printing the written path in automatic configuration mode.
-s <i>address</i>	Specifies the IP address where the named daemon listens for command channel connections from rndc command authentication key. The default is the loopback address 127.0.0.1 .
-t <i>chrootdir</i>	Used with the -a flag to specify a directory where the named daemon runs chrooted. An extra copy of the <code>rndc.key</code> file is written relative to this directory that is found by the chrooted named daemon.
-u <i>user</i>	Used with the -a flag to set the owner of the <code>rndc.key</code> file generated. If the -t flag is also specified, only the file in the chroot area has its owner changed.

Examples

1. To use the **rndc** command with no manual configuration, enter the following command:

```
rndc-confgen -a
```

2. To print a sample **rndc** .conf file and have corresponding controls and key statements to be manually inserted into the named .conf file, enter the following command:

```
rndc-confgen
```

roffbib Command

Purpose

Prints a bibliographic database.

Syntax

roffbib [**-m** *Macro*] [**-x**] [*FormatFlags*] [*Database...*]

Description

The **roffbib** command prints out all records that are in a bibliographic database format rather than in a format for footnotes or endnotes. Generally, the command is used as a filter for the **troff** command, in particular, the **-e**, **-h**, **-n**, **-o**, **-r**, **-s**, and **-T** flags.

If abstracts or comments are entered following the **%X** key field, they are formatted into paragraphs for an annotated bibliography. Several **%X** fields can be given if several annotation paragraphs are desired.

Parameters

Item	Description
<i>FormatFlags</i>	Accepts most of the nroff command flags, especially the -e , -h , -n , -o , -r , -s , and -T flags.
<i>Database</i>	Stores a bibliographic database of all records.

Flags

Item	Description
-m <i>Macro</i>	Specifies a file that contains a user-defined set of macros. There should be a space between the -m flag and the macro. This set of macros replaces the ones defined in the /usr/share/lib/tmac/tmac.bib file. Users can rewrite macros to create customized formats.
-x	Suppresses the printing of abstracts or comments that are entered following the %X field key.

Examples

Following is an example of the **roffbib** command used in conjunction with the **html**

rolelist Command

Purpose

Displays role information for a user or process.

Syntax

```
rolelist [-a] [-e | -u username | -p PID]
```

Description

The **rolelist** command provides role and authorization information to the invoker about their current roles or the roles assigned to them. If no flags or arguments are specified, the **rolelist** command displays the list of roles assigned to the invoker on the real user ID with the text description of each role if one is provided in the roles database. Specifying the **-e** flag outputs information about the current effective active role set for the session. If the invoker is not currently in a role session and specifies the **-e** flag, no output is displayed. Specifying the **-a** flag displays the authorizations associated with the roles instead of the text description.

The **rolelist** command also allows a privileged user to list the role information for another user or for a process. Specifying a user name with the **-u** flag allows a privileged user to list the roles assigned to another user. The active role set of a given user cannot be determined because the user can have multiple active role sessions. Therefore, if the **-u** flag is specified, the **-e** flag is not allowed. Specifying a process ID with the **-p** flag allows a privileged user to display the roles associated with a process. The command fails immediately if invoked by a non-privileged user when the **-u** or **-p** flag is specified.

The authorization information displayed by the **rolelist** command is retrieved from the kernel security tables. The information can differ with the current state of the roles database if it is modified after the kernel security tables are updated.

Flags

Item	Description
-a	Displays the authorizations assigned to each role instead of the role description.
-e	Displays information about the effective active role set of the session.
-u <i>username</i>	Displays role information for the specified user.
-p <i>PID</i>	Displays role information of the specified process.

Security

All users can run the **rolelist** command. To query the role information of another user or a process, the following authorizations are required.

Item	Description
aix.security.role.list	Required to invoke the command on another user.
aix.security.proc.role.list	Required to list the roles associated with a process.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Files Accessed

Files	Mode
/etc/security/user.roles	r
/etc/security/roles	r

Examples

1. To display the list of roles that assigned to you and their text descriptions, use the following command:

```
rolelist
```

Information similar to the following example is displayed:

UserAdmin	User Administrator
RoleAdmin	Role Administrator
FSAdmin	File System Administrator

2. To display the authorizations associated with the assigned roles, use the following command:

```
rolelist -a
```

Information similar to the following example is displayed:

UserAdmin	aix.security.user
RoleAdmin	aix.security.role
FSAdmin	aix.security.fs

3. As a privileged user, use the following command to display the roles assigned to a specific user :

```
rolelist -u user1
```

Information similar to the following example is displayed:

SysInfo	System Information Retrieval

roleqry Command

Purpose

Queries the usage of roles over a time period.

Syntax

```
roleqry {-c [-s] | -q [-F <trailListfile> ] [-t <time_period_in_days> ] } user
```

Description

The **roleqry** command queries information about the roles used by a user over a specified time frame.

When the **-c** flag is specified, the user is configured for the auditing of role information and authorization information. A **rbacqry** class is added to the /etc/security/audit/config file with events for auditing authorizations and roles. If the user is already being audited, a user entry present in the configuration file, then the **rbacqry** class is added to the user. Otherwise the username is added to the /etc/security/audit/config with the **rbacqry** class parameter. If the **-s** flag is specified, the user is enabled for audit. If the audit subsystem is already turned on, then it is restarted. If the audit system is already turned off, then the audit subsystem is started.

When the **-q** flag is specified, the audit data is queried for role information. When the **-t** flag is specified, the usage of roles from the date to the current system date is queried and obtained. Without **-t** flag, role

usage over the period from which auditing was enabled for that user is obtained. The command displays the entire set of roles used during the time frame.

Note: The **roleqry** commands make use of the auditing feature in AIX. Auditing has to be turned on, audit configuration for the user setup and the audit data collected during the specified time frame for the **roleqry** command to work as expected.

Flags

Item	Description
-c	Use this flag to configure the user for auditing of role usage.
-s	Use this flag to start auditing subsystem if it is turned off. Shutdown and restart auditing subsystem if it is already turned on.
-q	Use this flag to query audit data for role usage over a time period.
-F	Use this flag to read the names of the audit trails to obtain audit information from the <i>trailListFile</i> . The names of audit trail files should be one name per line of text. If the -F flag is not specified, the system “audit/trail file is taken by default as the file to obtain audit information from.
-t	Use this flag to specify the number of days from the current date to get the authorization usage.

Exit Status

Item	Description
0	Successful completion.
>0	An error occurred.

Security

Access Control: This command should grant execute (x) access to the root user.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Files:

- /etc/security/roles
- audit/trail

Examples

1. To query roles used by Bob run the following command:

```
roleqry -q Bob
```

2. To query roles used by Simon for the past 20 days run the following command:

```
roleqry -q -t 20 Simon
```


rolerpt Command

Purpose

Reports the security capabilities of roles.

Syntax

rolerpt [-R <load_module>] [-C] [-c | -f] { "ALL" | role1, role2, ... | -a }

rolerpt [-R <load_module>] [-C] [-u] { "ALL" | role1, role2, ... }

Description

The **rolerpt** command reports capability information of roles such as privileged commands, privileged files, and user information.

Either of **-c**, **-f**, or **-u** flags can be specified. When the **-c** flag is specified, the privileged commands present in the `/etc/security/privcmds` database that can be run by virtue of the roles is listed. When the **-f** flag is specified, the list of privileged files present in the `/etc/security/privfiles` database that can be accessed by users that are assigned to the roles is displayed.

When the **-u** flag is specified, the list of users with roles are displayed based on the Loadable Authentication Model (LAM) 's that is configured in the `/etc/nscontrol.conf` database. The **-u** flag can be used only by a root user or a privileged user that is authorized for the **rolerpt** command. Only root user or the authorized user with **aix.security.role.list** authorization can view reports that display capabilities for roles that are not held by them.

When no flag is specified, all the capability information such as commands, privileged files, and user information for the role is displayed.

The **-a** flag specifies the capabilities of the active roles. The **-u** flag cannot be used with the **-a** flag. The root user or the authorized user can specify the **ALL** keyword to display capabilities for all the roles on the system.

The **rolerpt** command accepts inputs such as **-a** flag to specify the active roles, the **ALL** keyword, or a comma-separated list of role names. When no role name is specified, all the capability information such as commands, privileged files, and user information that is associated with the roles of the invoker is displayed.

Flags

Item	Description
-a	Specifies that report on only capabilities of active roles is to be obtained.
-c	Specifies that a report of privileged commands executable by the roles is to be obtained.
-C	Displays the role attributes in colon-separated records, as displayed in the following example: <pre>#role:attribute1:attribute2: ... role1:value1:value2: ... role2:value1:value2: ...</pre>
-f	Specifies that a report of privileged file information accessible to the roles is to be obtained.
-R	Specifies the loadable module to obtain the report of roles capabilities from.
-u	Specifies that a report of authorized user information that is assigned to the roles is to be obtained.

Exit status

Item	Description
0	Successful completion.
>0	An error occurred.

Security

Access Control: This command must grant execute (x) access to all users. The **-u** flag can be used by the root user or authorized users with **aix.security.role.list** authorization or **aix.security.user.list** authorization. Only root or the authorized user with **aix.security.role.list** authorization can specify the **ALL** keyword and view reports of capabilities of roles that are not held by them.

Attention RBAC users: This command does privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, review the Privileged Command Database topic. For a list of privileges and the authorizations that are associated with this command, review the **lssecattr** command or the **getcmdattr** subcommand.

Files

- /etc/security/roles
- /etc/security/authorizations
- /etc/security/privcmds
- /etc/security/privfiles

Examples

1. To report the commands that are associated with the role ManageAllUsers, run the following command:

```
rolerpt -c ManageAllUsers
```

2. To report capabilities of active roles that are, the authorization, command, and privileged file information run the following command:

```
rolerpt -a
```

3. To report all capabilities of role ManageAllUsers in a colon separated format, run the following command:

```
rolerpt -C ManageAllUsers
Information similar to the following appears:
#role:commands:privfiles:users
ManageAllUsers:/usr/bin/lsuser,/usr/bin/mkuser:/var/adm/sulog:Bob,Simon
```

rollback Command

Purpose

Reverts a JFS2 file system to a point-in-time snapshot.

Syntax

To rollback to an external snapshot

```
rollback [-s ] [ -v ] [-c] snappedFS snapshotObject
```

To rollback to an internal snapshot

```
rollback [ -v ] -n snapshotName snappedFS
```

Description

The `rollback` command is an interface to revert a JFS2 file system to a point-in-time snapshot. The *snappedFS* parameter must be unmounted before the `rollback` command is run and remains inaccessible for the duration of the command. Any snapshots that are taken after the specified snapshot (*snapshotObject* for external or *snapshotName* for internal) are removed. The associated logical volumes are also removed for external snapshots.

If the `rollback` command is interrupted for any reason, the *snappedFS* parameter remains inaccessible until the command is restarted and completes. A restarted `rollback` must target the same *snapshotObject* or *snapshotName* as the initial command.

Flags

Item	Description
-c	If specified, <code>rollback</code> continues even if read or write errors are observed when restoring the <i>snappedFS</i> from the snapshot. If you do not specify the -c flag, an error message is issued and the rollback stops. Run the fsck command in this case.
-n <i>snapshotName</i>	Specifies the name of the internal snapshot to use for the rollback.
-s	If specified, any logical volumes associated with snapshots removed by <code>rollback</code> will be preserved. The snapshots are still deleted.
-v	This is the verbose option and causes a count of restored blocks to be printed as the rollback progresses.

Parameters

Item	Description
<i>snappedFS</i>	The JFS2 system to roll back.
<i>snapshotObject</i>	The logical volume of the external snapshot to revert to.

Examples

To roll back the `/home/janet/sb` file system to the external snapshot on logical volume `/dev/snapsb`, enter:

```
rollback /home/janet/sb /dev/snapsb
```

Location

Item	Description
<code>/usr/sbin/rollback</code>	Contains the <code>rollback</code> command.

route Command

Purpose

Manually manipulates the routing tables.

Syntax

route [**-f**] [**-n**] [**-q**] [**-C**] [**-v**] *Command* [*Family*] [[**-net** | **-host**] *Destination* [**-prefixlen** *n*] [**-netmask** [*Address*]] *Gateway* [*Arguments*] [**-i**] [**-@** *WparName*]

Description

The **route** command allows you to make manual entries into the network routing tables. The **route** command distinguishes between routes to hosts and routes to networks by interpreting the network address of the *Destination* variable, which can be specified either by symbolic name or numeric address. The **route** command resolves all symbolic names into addresses, using either the /etc/hosts file or the network name server.

Routes to a particular host are distinguished from those to a network by interpreting the Internet address associated with the destination. The optional phs **-net** and **-host** force the destination to be interpreted as a network or a host, respectively. If the destination has a local address part of INADDR_ANY or if the destination is the symbolic name of a network, then the route is assumed to be to a network; otherwise, it is presumed to be a route to a host.

For example, 128.32 is interpreted as **-host** 128.0.0.32; 128.32.130 is interpreted as **-host** 128.32.0.130; **-net** 128.32 is interpreted as 128.32.0.0; and **-net** 128.32.130 is interpreted as 128.32.130.0.

If the route is by way of an interface rather than through a gateway, the **-interface** argument should be specified. The specified gateway is the address of the host on the common network, indicating the interface to be used for transmission.

The **-netmask** argument must be followed by an address parameter (to be interpreted as a network mask). One can override the implicit network mask generated in the **-inet** case by making sure this option follows the *Destination* parameter.

All symbolic names specified for a destination or gateway are looked up first as a host name, using the **gethostbyname** subroutine. If this fails, the **getnetbyname** subroutine is then used to interpret the name as a network name.

Note: Route uses a routing socket and the new message types RTM_ADD, RTM_DELETE, and RTM_CHANGE. As such, only the root user may modify the routing tables.

If the **flush** or **-f** command is specified, route will "flush," or clear, the routing tables of all gateway entries. One can choose to flush only those routes whose destinations are of a given address family, by specifying an optional ph describing which address family.

The **netstat -r** command displays the current routing information contained in the routing tables.

Note: You must use the same set of commands (**route**, **smitty**, **ifconfig** and **chdev** commands) for creating or deleting the routing table. If you create the routing table by using the **smitty** or **chdev** command, and delete it by using the **route** command, the route entry is not deleted from Object Data Manager (ODM), and if the system gets restarted, the routing table gets builds from ODM, due to which you can see the same route entry again.

Flags

Item	Description
-f	Purges all entries in the routing table that are not associated with network interfaces.

Item	Description
-i	Enables workload-partition-specific routing for the workload partition (WPAR). By default, outgoing network traffic from a WPAR is routed as if it were being sent from the global environment: • Traffic between addresses that are hosted on the same global system is sent through the loopback interface. • Routing table entries that are configured in the global system, including the default route, are used to transmit workload partition traffic. If you enable WPAR-specific routing by specifying the -i flag, the WPAR creates and uses its own routing table for the outgoing traffic. Routing entries are created automatically for each of the network addresses of the WPAR to accommodate broadcast, loopback, and subnet routes.
-n	Displays host and network names numerically, rather than symbolically, when reporting results of a flush or of any action in verbose mode.
-q	Specifies quiet mode and suppresses all output.
-C	Specifies preference for ioctl calls over routing messages for adding and removing routes.
-v	Specifies verbose mode and prints additional details.
-net	Indicates that the <i>Destination</i> parameter should be interpreted as a network.
-netmask	Specifies the network mask to the destination address. Make sure this option follows the <i>Destination</i> parameter.
-host	Indicates that the <i>Destination</i> parameter should be interpreted as a host.
-prefixlen <i>n</i>	Specifies the length of a destination prefix (the number of bits in the netmask).
-@WparName	Displays the network statistics that are associated with the WPAR that is, (<i>@WparName</i> flag). If the <i>@WparName</i> flag is not specified, the network statistics for all the WPARs are displayed.

The route default is a host (a single computer on the network). When neither the **-net** parameter nor the **-host** parameter is specified, but the network portion of the address is specified, the route is assumed to be to a network. The host portion of the address is 0 (zero).

Parameters

Item	Description
<i>Arguments</i>	Specifies one or more of the following arguments. Where <i>n</i> is specified as a variable to an argument, the value of the <i>n</i> variable is a positive integer. -active_dgd Enables Active Dead Gateway Detection on the route. -cloning Clones a new route. -genmask Extracts the length of TSEL, which is used for the generation of cloned routes. -interface Manipulates interface routing entries. -rtt <i>n</i> Specifies round-trip time. -rttvar <i>n</i> Specifies round-trip time variance.

Item	Description
-sendpipe <i>n</i>	Specifies send-window size.
-recvpipe <i>n</i>	Specifies receive-window size.
-allowgroup <i>gid</i>	Specifies a group ID that is allowed to use the route. The group ID will be added to a list of allowed groups or deleted from a list of denied groups.
-denygroup <i>gid</i>	Specifies a group ID that is not allowed to use the route. The group ID will be added to a list of denied groups or deleted from a list of allowed groups.
-stopsearch	Stops searching if a routing table lookup matches the route, but it is not allowed to use the route due to group routing restrictions.
-mtu <i>n</i>	Specifies maximum transmission unit for this route. Will override interface mtu for TCP applications as long as it does not exceed maximum mtu for the interface. This flag has no affect on mtu for applications using UDP.
-hopcount <i>n</i>	Specifies maximum number of gateways in the route.
-policy <i>n</i>	Specifies the policy to be used for Multipath Routing. <i>n</i> is number between 1 and 5 where these numbers mean the following: 1. Weighted Round-Robin 2. Random 3. Weighted Random 4. Lowest Utilization 5. Hash-based If the policy is not explicitly set and multipath routing is used, then the global no command option called mpr_policy determines the policy that will be used. The default policy is Weighted Round Robin which behaves just like Round-Robin when the weights are all 1. Although the Default policy is Weighted Round-Robin, when the policy is not set, then the network option mpr_policy takes precedence. On the other hand, if the policy is explicitly set to WRR then this setting overrides the mpr_policy setting. For more information about these policies, see the no command.
-weight <i>n</i>	Specifies the weight of the route that will be used for the Weighted policies with the Multipath Routing feature.

Item	Description
	-expire <i>n</i> Specifies expiration metrics used by routing protocol -sssthresh <i>n</i> Specifies outbound gateway buffer limit. -lock Specifies a meta-modifier that can individually lock a metric modifier. The -lock meta-modifier must precede each modifier to be locked. -lockrest Specifies a meta-modifier that can lock all subsequent metrics. -if <i>ifname</i> Specifies the interface (en0, tr0 ...) to associate with this route so that packets will be sent using this interface when this route is chosen. -xresolve Emits a message on use (for external lookup). -iface Specifies that the destination is directly reachable. -static Specifies the manually added route. -nostatic Specifies the pretend route that is added by the kernel or daemon. -reject Emits an ICMP unreachable when matched. -blackhole Silently discards packets during updates. -proto1 Sets protocol specific routing flag number 1. -proto2 Sets protocol specific routing flag number 2.
Command	Specifies one of six possibilities: add Adds a route. flush or -f Removes all routes. delete Deletes a specific route. change Changes aspects of a route (such as its gateway). monitor Reports any changes to the routing information base, routing lookup misses, or suspected network partitionings. get Lookup and display the route for a destination. set Set the policy and weight attributes of a route.
Family	Specifies the address family. The -inet address family is the default. The -inet6 family specifies that all subsequent addresses are in the inet6 family.

Item	Description
<i>Destination</i>	Identifies the host or network to which you are directing the route. The <i>Destination</i> parameter can be specified either by symbolic name or numeric address.
<i>Gateway</i>	Identifies the gateway to which packets are addressed. The <i>Gateway</i> parameter can be specified either by symbolic name or numeric address.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To establish a route so that a computer on one network can send a message to a computer on a different network, type:

```
route add 192.100.201.7 192.100.13.7
```

The 192.100.201.7 address is that of the receiving computer (the *Destination* parameter). The 192.100.13.7 address is that of the routing computer (the *Gateway* parameter).

2. To establish a route so you can send a message to any user on a specific network, type:

```
route add -net 192.100.201.0 192.100.13.7
```

The 192.100.201.0 address is that of the receiving network (the *Destination* parameter). The 192.100.13.7 address is that of the routing network (the *Gateway* parameter).

3. To establish a default gateway, type:

```
route add 0 192.100.13.7
```

The value 0 or the default *ph* for the *Destination* parameter means that any packets sent to destinations not previously defined and not on a directly connected network go through the default gateway. The 192.100.13.7 address is that of the gateway chosen to be the default.

4. To clear the host gateway table, type:

```
route -f
```

5. To add a route specifying weight and policy information, type:

```
route add 192.158.2.2 192.158.2.5 -weight 5 -policy 4
```

6. To set the weight and policy attributes of a preexisting route, type:

```
route set 192.158.2.2 192.158.2.5 -weight 3 -policy
```

routed Daemon

Purpose

Manages network routing tables.

Syntax

Note: Use SRC commands to control the **routed** daemon from the command line. Use the **gated** daemon, which supports all TCP/IP gateway protocols, the **routed** daemon only implements the Routing

Information Protocol (RIP). Do not use the **routed** daemon when Exterior Gateway Protocol (EGP), Simple Network Management Protocol (SNMP), or Distributed Computer Network Local-Network Protocol routing is needed. Use the `/etc/gateways` file for information about these distant and external gateways.

The `/etc/gateways` file contains information about routes through distant and external gateways to hosts and networks that should be advertised through RIP. These routes can be either static routes to specific destinations or default routes for use when a static route to a destination is unknown. The format of the `/etc/gateways` file is:

```
{ net | host } name1 gateway name2 metric { passive | active | external }
```

When a gateway specified in the `/etc/gateways` file supplies RIP routing information, it should be marked as active. Active gateways are treated like network interfaces. That is, RIP routing information is distributed to the active gateway. If no RIP routing information is received from the gateway for a period of time, the **routed** daemon deletes the associated route from the routing tables.

A gateway that does not exchange RIP routing information should be marked as passive. Passive gateways are maintained in the routing tables indefinitely. Information about passive gateways is included in any RIP routing information transmitted.

An external gateway is identified to inform the **routed** daemon that another routing process will install such a route and that the **routed** daemon should not install alternative routes to that destination. External gateways are not maintained in the routing tables and information about them is not included in any RIP routing information transmitted.

Note: Routes through external gateways must be to networks only.

The **routed** daemon can also perform name resolution when routing to different networks. For example, the following command adds a route to the network called `netname` through the gateway called `host1`. The `host1` gateway is one hop count away.

```
route add net netname host1 1
```

To perform network name resolution, the **routed** daemon uses the `/etc/networks` file to get information on the network addresses and their corresponding names. To perform host name resolution, the **routed** daemon must take additional steps before the routing is complete. First the daemon checks for the existence of the `/etc/resolv.conf` file. This file indicates whether the host is running under a domain name server, and if so, gives the IP address of the host machine running the **named** daemon.

If the `/etc/resolv.conf` file does not exist, the **routed** daemon uses the `/etc/hosts` file to find the host for which it is routing.

The **routed** daemon should be controlled using the System Resource Controller (SRC) or the System Management Interface Tool (SMIT). Entering the **routed** daemon at the command line is not recommended.

Manipulating the routed Daemon with the System Resource Controller

The **routed** daemon is a subsystem controlled by the System Resource Controller (SRC). The **routed** daemon is a member of the SRC **tcpip** system group. This daemon is disabled by default and can be manipulated by the following SRC commands:

Item	Description
startsrc	Starts a subsystem, group of subsystems, or subserver.
stopsrc	Stops a subsystem, group of subsystems, or subserver.
tracesoff	Disables tracing of a subsystem, group of subsystems, or subserver.
lssrc	Gets the status of a subsystem, group of subsystems, or subserver.

Signals

The following signals have the specified effect when sent to the **routed** process using the **kill** command:

Item	Description
SIGINT	Restarts the routed daemon and flushes the routing table.
SIGHUP, SIGTERM, or SIGQUIT	Broadcasts RIP packets with hop counts set to infinity. These signals disable the local host as a router. After a second SIGHUP, SIGTERM, or SIGQUIT signal, the routed daemon terminates.
SIGUSR1	Turns packet tracing on or, if packet tracing is already on, steps up the tracing one level. The first level traces transactions only. The second level traces transactions plus packets. The third level traces the packet history, reporting packet changes. The fourth level traces packet contents. This command increments the level of tracing through four levels.
SIGUSR2	Turns packet tracing off.

Flags

Item	Description
-d	Enables additional debugging information, such as bad packets received, to be logged.
-g	Runs the routing daemon on a gateway host. The -g flag is used on internetwork routers to offer a route to the default destination.
-q	Prevents the routed daemon from supplying routing information regardless of whether it is functioning as an internetwork router. The -q flag indicates "quiet". Do not use the -q flag and the -s flag together.
-s	Supplies routing information regardless of whether it is functioning as an internetwork router. The -s flag indicates "supply". Do not use the -q flag and the -s flag together.
-t	Writes all packets sent or received to standard output or to the file specified in the <i>LogFile</i> parameter. The routed daemon remains under control of the controlling terminal that started it. Therefore, an interrupt from the controlling terminal keyboard stops the routed process.

Examples

1. To start the **routed** daemon manually, type:

```
startsrc -s routed -a "-s"
```

Note: The **routed** daemon is not started by default at each system startup. Use the **rc.tcpip** file format and a System Resource Controller (SRC) command to start the **routed** daemon. You can also start the **routed** daemon using the System Management Interface Tool (SMIT).

The **-s** flag causes the **routed** daemon to return routing information regardless of whether the **routed** daemon is an internetwork router.

2. To stop the **routed** daemon, type the following:

```
stopsrc -s routed
```

3. To get a short-status report from the **routed** daemon, type the following:

```
lssrc -s routed
```

This command returns the name of the daemon, the process ID of the daemon, and the state of the daemon (active or inactive).

4. To enable tracing for **routed** daemon, type the following:

```
traceson -s routed
```

This command enables socket-level debugging. Use the `html`

rpc.pcnfsd Daemon

Purpose

Handles service requests from PC-NFS (Personal Computers Network File System) clients.

Syntax

`/usr/sbin/rpc.pcnfsd`

Description

The **rpc.pcnfsd** daemon handles requests from PC-NFS clients for authentication services on remote machines. These services include authentication for mounting and for print spooling. The PC-NFS program allows personal computers running DOS to be networked with machines running NFS. The **rpc.pcnfsd** daemon supports Versions 1 and 2 of the **pcnfsd** protocol.

When a PC-NFS client makes a request, the **inetd** daemon starts the **rpc.pcnfsd** daemon (if the **inetd.conf** file contains the appropriate entry). The **rpc.pcnfsd** daemon reads the **umask** specifications. A record of logins is appended to the **exportfs** command and the **enq** command. The daemon adopts the identity of the personal computer user to execute the print request command. Because constructing and executing the command involves user ID privileges, the **rpc.pcnfsd** daemon must be run as a root process.

All print requests from clients include the name of the printer to be used. The printer name is represented by queue and device definitions in the **/etc/qconfig** file. Additionally, the **rpc.pcnfsd** daemon provides a method for defining PC-NFS virtual printers recognized only by **rpc.pcnfsd** clients. Each PC-NFS virtual printer is defined in the **/etc/pcnfsd.conf** file with a line similar to the following:

```
printer Name AliasFor Command
```

In this format, **Name** specifies the name of the printer to be defined, and **AliasFor** is the name of the existing printer that will do the work. For example, a request to show the queue for **Name** translates into a queue command on the **AliasFor** printer. To define a printer **Name** with no existing printer, use a single **-** (minus sign) in place of the **AliasFor** parameter. The **Command** parameter specifies a command run when a file is printed on the **Name** printer. This command is executed by the Bourne shell, using the **-c** option. For complex operations, replace the **Command** parameter with an executable shell script.

The following list of tokens and substitution values can be used in the *Command* parameter:

Token	Substitution Value

\$FILE	The full path name of the print data file. After the command has executed, the file is unlinked.
\$USER	The user name of the user logged-in to the client.
\$HOST	The host name of the client system.

Examples

The following example **/etc/pcnfsd.conf** file configures a virtual printer on the first line and a null device for testing on the second line:

```
printer rotated lw /bin/enscript -2r $FILE
printer test - /usr/bin/cp $FILE /usr/tmp/$HOST-$USER
```

The first line stipulates that if a client system prints a job on the `rotated` printer, the `enscript` utility is called to preprocess the `$FILE` file. The `-2r` option causes the file to be printed in two-column, rotated format on the default PostScript printer. If a client requests a list of the print queue for the `rotated` printer, the **rpc.pcnfsd** daemon translates this request into a request for a similar listing for the `lw` printer.

The second line establishes a printer test. Files sent to the test printer are copied into the `/usr/tmp` directory. Requests to the test printer to list the queue, check the status, or perform similar printer operations, are rejected because `-` (minus sign) is specified in place of the *AliasFor* parameter.

Files

Item	Description
<code>/etc/pcnfsd.conf</code>	Contains the rpc.pcnfsd daemon configuration file.
<code>/var/spool/pcnfs</code>	Contains the default print-spooling directory.

rpcgen Command

Purpose

Generates C code to implement an RPC protocol.

Syntax

To Generate Four Types of Output Files for a File

```
/usr/bin/rpcgen InputFile
```

To Generate a Specific Output File for a File

```
rpcgen { -c | -h | -l | -m } [ -o OutputFile ] [ InputFile ]
```

To Generate a Server-Side File for TCP or UDP

```
rpcgen { -s Transport ... } [ -o OutputFile ] [ InputFile ]
```

Description

The **rpcgen** command generates C code to implement a Remote Procedure Call (RPC) protocol. The input to the **rpcgen** command is a language similar to C language known as RPC Language.

The first syntax structure is the most commonly used form for the **rpcgen** command where it takes an input file and generates four output files. For example, if the *InputFile* parameter is named **proto.x**, then the **rpcgen** command generates the following:

Item	Description
proto.h	Header file
proto_xdr.c	XDR routines
proto_svc.c	Server-side stubs
proto_clnt.c	Client-side stubs

Use the other syntax structures when you want to generate a particular output file rather than all four output files.

The **cpp** command, a C preprocessor, is run on all input files before they are actually interpreted by the **rpcgen** command. Therefore, all the **cpp** directives are legal within an **rpcgen** input file. For each type of output file, the **rpcgen** command defines a special **cpp** symbol for use by the **rpcgen** programmer:

Item	Description
RPC_HDR	Defined when compiling into header files
RPC_XDR	Defined when compiling into XDR routines
RPC_SVC	Defined when compiling into server-side stubs
RPC_CLNT	Defined when compiling into client-side stubs

In addition, the **rpcgen** command does some preprocessing of its own. Any line beginning with a **%** (percent sign) passes directly into the output file, uninterpreted by the **rpcgen** command.

To create your own XDR routines, leave the data types undefined. For every data type that is undefined, the **rpcgen** command assumes that a routine exists by prepending **xdr_** to the name of the undefined type.

Note:

1. Nesting is not supported. As a work-around, structures can be declared earlier with their names used inside other structures in order to achieve the same effect.
2. Name clashes can occur when using program definitions since the apparent scoping does not really apply. Most of these can be avoided by giving unique names for programs, versions, procedures, and types.
3. To program to the TIRPC interfaces, and allow the use of multi-threaded RPC applications use the **tirpcgen** command. It will also be necessary to define the preprocessor variable **_AIX_TIRPC** in the Makefile as well as the **libtli.a (-ltli)** specification. **tirpcgen** is a temporary name for a new **rpcgen** command that will replace **rpcgen** in a future version the operating system.

Flags

Item	Description
-c	Compiles into XDR routines.
-h	Compiles into C-data definitions (a header file).
-l	Compiles into client-side stubs.
-m	Compiles into server-side stubs, but does not generate a main routine. This option is useful for doing call-back routines and for writing a main routine to do initialization.
-o <i>OutputFile</i>	Specifies the name of the output file. If none is specified, standard output is used.
-s <i>Transport</i>	Compiles into server-side stubs, using given transport. The supported transports are udp and tcp . This flag can be run more than once to compile a server that serves multiple transports.

rpcinfo Command

Purpose

Reports the status of Remote Procedure Call (RPC) servers.

Syntax

To Display a List of Statistics

```
/usr/bin/rpcinfo [ -m | -s ] [ Host ]
```

To Display a List of Registered RPC Programs

```
/usr/bin/rpcinfo -p [ Host ]
```

To Report Transport

```
/usr/bin/rpcinfo -T transport Host Prognum [ Versnum ]
```

To Display a List of Entries

```
/usr/bin/rpcinfo -l [ -T transport ] Host Prognum Versnum
```

To Report Program Status using UDP

```
/usr/bin/rpcinfo [ -n PortNum ] -u Host Prognum [ Versnum ]
```

To Report Program Status using TCP

```
/usr/bin/rpcinfo [ -n PortNum ] -t Host Prognum [ Versnum ]
```

To Report Program Status

```
/usr/bin/rpcinfo -a ServAddress -T transport Host Prognum [ Versnum ]
```

To Display All Hosts Running a Specified Program Version

```
/usr/bin/rpcinfo [ -b ] [ -T transport ] Prognum Versnum
```

To Delete Registration of a Service

```
/usr/bin/rpcinfo [ -a -d ] [ -T transport ] Prognum Versnum
```

Description

The **rpcinfo** command makes an RPC call to an RPC server and reports the status of the server. For instance, this command reports whether the server is ready and waiting or not available.

The program parameter can be either a name or a number. If you specify a version, the **rpcinfo** command attempts to call that version of the specified program. Otherwise, the **rpcinfo** command attempts to find all the registered version numbers for the program you specify by calling version 0 (zero) and then attempts to call each registered version. (Version 0 is presumed not to exist. If it does exist, the **rpcinfo** command attempts to obtain this information by calling an extremely high version number instead.)

Flags

Item	Description
-a	Specifies the complete IP address and port number of the host.
-b	Makes an RPC broadcast to procedure 0 of the specified prognum and versnum and reports all hosts that respond. If <i>transport</i> is specified, it broadcasts its request only on the specified <i>transport</i> . If broadcasting is not supported by any <i>transport</i> , an error message is printed. Using broadcasting (-b flag) should be limited because of the possible adverse effect on other systems.
-d	Deletes registration for the RPC service of the specified prognum and versnum. If <i>transport</i> is used, unregister the service only on that <i>transport</i> , otherwise unregister the service on all the transports where it was registered. This option can be exercised only by the root user.

Item	Description
-l	Displays a list of entries with the specified prognum and versnum on the specified host. Entries are returned for all transports in the same protocol family as those used to contact the remote portmap daemon.
-m	Displays a table of portmap operations statistics on the specified host. The table contains statistics for each version of portmap (Versions 2, 3, and 4), the number of times each procedure was requested and successfully serviced, the number and type of remote call requests that were made, and information about RPC address lookups that were handled. This information is used for monitoring RPC activities on the host.
-n <i>Portnum</i>	Use the <i>Portnum</i> parameter as the port number for the -t and -u options instead of the port number given by the portmap. Using the -n options avoids a call to the remote portmap to find out the address of the service. This option is made obsolete by the -a option.
-p	Probes the portmap service on the host using Version 2 of the portmap protocol and displays a list of all registered RPC programs. If a host is not specified, it defaults to the local host.
-s	Displays a concise list of all registered RPC programs on the host. If host is not specified, the default is the local host.
-t	Makes an RPC call to procedure 0 of prognum on the specified host using TCP, and reports whether a response was received. This option is made obsolete when using the -T option as shown in the third syntax.
-T	Specifies the transport where the service is required.
-u	Makes an RPC call to procedure 0 of prognum on the specified host using UDP, and reports whether a response was received. This option is made obsolete when using the -T option as shown in the third syntax.

Examples

1. To show all of the RPC services registered on a local machine, enter:

```
rpcinfo -p
```

2. To show all of the RPC services registered on a specific machine, enter:

```
rpcinfo -p zelda
```

In this example, the **rpcinfo** command shows all RPC services registered on a machine named *zelda*.

3. To show all machines on the local network that are running a certain version of a specific server, enter:

```
rpcinfo -b ypserv 2
```

In this example, the **rpcinfo** command shows a list of all machines that are running version 2 of the **ypserv** daemon.

4. To delete the registration of a service, enter:

```
rpcinfo -d sprayd 1
```

In this example, the **rpcinfo** command deletes version 1 of the **sprayd** daemon.

5. To check whether the host with IP address 127.0.0.1, program 100003, and version 3 is listening on port 2049 over the TCP, enter:

```
rpcinfo -a 127.0.0.1.8.1 -T tcp 100003 3
```

Files

Item	Description
/etc/services	Contains an entry for each service available through the Internet network.

rpvstat Command

Purpose

Displays RPV client device statistics.

Syntax

```
rpvstat -h
rpvstat [-d] [-t] [-i Interval] [-c Count] [-d]] [rpvclient_name...]
rpvstat -N [-t] [-i Interval] [-c Count] [-d]]
rpvstat -m [-d] [-t] [rpvclient_name...]
rpvstat -R [-r] [rpvclient_name...]
rpvstat -r [-R] [rpvclient_name...]
rpvstat -A [-t] [-i Interval] [-d] [-c Count] ] [rpvclient_name...] |
rpvstat -C [-t] [-i Interval] [-d] [-c Count] ] [rpvclient_name...]
>|rpvstat -G [-t] [-i Interval] [-d] [-c Count] ] [rpvclient_name...]
rpvstat -g [-t] [-i Interval] [-d] [-c Count] ] [rpvclient_name...] |<
```

Description

The **rpvstat** command displays statistical information available from the RPV client device that includes the following details:

- RPV client name
- Connection status
- Total number of completed reads
- Total number of KBs read
- Total number of read errors
- Total number of pending reads
- Total number of pending KBs to read
- Total number of completed writes
- Total number of KBs written
- Total number of write errors
- Total number of pending writes
- Total number of pending KBs to write
- Statistics for asynchronous I/O
- Statistics for asynchronous I/O cache

The read and write errors are displayed together. These counters indicate the number of I/O errors returned to the application.

The **rpvstat** command can optionally display its I/O-related statistics on a per-network basis. A network summary option of the command displays the following additional information:

- Network throughput in kilobytes per second. The throughput is calculated per interval time specified by the user while in monitoring mode.
- The highest recorded values for the pending statistics. These historical high water mark numbers are:
 - Maximum number of pending reads per network
 - Maximum number of pending kilobytes to read per network
 - Maximum number of pending writes per network
 - Maximum number of pending kilobytes to write per network
- Number of retried I/O operations (both read and write operations). This count records the number of I/O retries that have occurred on this network or device. This can be used as an indicator for a marginal or failing network. These statistics are reported on a separate display.

You can also display the statistics for asynchronous mirroring. The **rpvstat** command prints overall asynchronous statistics using the **-A** option. To display statistics per device, you need to specify the list of devices. You can display the asynchronous I/O cache information using **-C** option.

Flags

Table 13. Flags	
Flag	Description
-A	Displays the following statistical information for one or more asynchronous I/O operations: • Asynchronous device name • Asynchronous status: The status is printed as a single character. – A - Device is fully configured for asynchronous I/O and can accept asynchronous I/O requests. – I - Asynchronous configuration is incomplete. – U - The device is not configured with asynchronous configuration. Hence it is acting as a synchronous device. All statistics will be printed as zero. – X - Device status cannot be retrieved. All the remaining statistics will be printed as zero. • Total number of asynchronous remote writes completed. The writes are mirrored and complete. • Total asynchronous remote writes completed in kilobyte. The writes are mirrored and complete. • Total number of asynchronous writes pending to mirror. The writes are in the cache. These writes are complete as per LVM is concerned but not yet mirrored. • Total asynchronous writes pending to mirror in kilobyte. The writes are in the cache. These writes are complete as per LVM is concerned but not yet mirrored. • Total number of writes whose response pending. These writes are in the pending queue and not yet written to cache. • Total asynchronous writes response pending in kilobyte. These writes are in the pending queue and not yet written to cache.

Table 13. Flags (continued)

Flag	Description
-C	Displays the following statistical information for asynchronous I/O cache. The VG name is extracted from the ODM. • Volume group name • Asynchronous status: The status is printed as a single character. – A - Device is fully configured for asynchronous I/O and can accept asynchronous I/O requests. – I - Asynchronous configuration is incomplete. – U - The device is not configured with asynchronous configuration. Hence it is acting as a synchronous device. All statistics will be printed as zero. – X - Device status can't be retrieved. All the remaining statistics will be printed as zero • Total asynchronous write operations • Maximum cache utilization in percent • Number of pending asynchronous write requests waiting for the cache flush after cache hits high water mark. • Percentage of writes waiting for the cache flush after cache hits high water mark limit. • Maximum time waited after cache hits high water mark in seconds. • Current free space that is available in cache in kilobytes.
-c <i>Count</i>	Displays information at the indicated interval for <i>Count</i> times. The value of the <i>Count</i> parameter must be an integer greater than zero and less than or equal to 999999. If the <i>Interval</i> parameter is specified, but the <i>Count</i> parameter is not, then it re-displays indefinitely.
-d	Displays applicable monitored statistics as delta amounts from prior value.
> > -G	Displays the following statistical information about the asynchronous I/O group: • Volume group name, which is obtained from the Object Data Manager (ODM) • Number of committed volume groups • Average volume group commit time • Total committed asynchronous I/O data in KB • Asynchronous I/O data that is committed per second in KB • Number of completed volume groups • Average volume group completion time • Total completed asynchronous I/O data in KB • Asynchronous I/O data that is completed per second in KB • Number of volume groups read from cache disk • Total asynchronous I/O data that is read from cache disk in KB • Average volume group read time • Asynchronous I/O data that is read per second in KB • Number of times cache disk is detected as full • Total waiting time for cache memory to be available • Total asynchronous I/O write data in transit in KB

Table 13. Flags (continued)

Flag	Description
> > -g < <	Displays a summary of statistical information about the asynchronous I/O group that includes the following data: • Volume group name, which is obtained from the ODM • Average volume group formation time • Average volume group commit time • Average volume group completion time • Average volume group read time • Number of times cache disk is detected as full
-h	Displays command syntax and usage.
-i <i>Interval</i>	Automatically displays status in every <i>Interval</i> seconds. The value of the <i>Interval</i> parameter must be an integer greater than zero and less than or equal to 3600. If the <i>Interval</i> parameter is not specified, then the status information is displayed once.
-m	Displays historical maximum pending values (high water mark values) and accumulated retry count.
-N	Displays summary statistics by mirroring network, including throughput rate for each network.
-n	Displays statistics for individual mirroring networks.
-R	Resets counters in the RPV clients (requires root privilege).
-r	Resets counters for the asynchronous I/O cache information. You can specify the -R and -r options together to reset all counters. Requires root access.
-t	Includes date and time in display.

Table 14. Operands

Field	Value
<i>rpvclient_name</i>	Name of one or more RPV clients for which to display information. If no RPV client names are specified, then information for all RPV clients is displayed.

Note:

- In monitor mode (**-i**) if the **-d** option is also specified, then some statistics (completed reads, completed writes, completed kilobyte read, completed kilobyte written, and errors) are represented as delta amounts from their previously displayed values. These statistics are prefixed with a plus sign (+) on the second and succeeding displays. A delta value is not displayed under certain circumstances, such as when an error is detected in the previous iteration, or a configuration change is made between iterations.
- When a list of RPV client devices is not explicitly listed on the command line, the list of all available RPV Clients is generated at command initiation. In monitor mode, this list of RPV clients to display is not refreshed on each display loop. This means any additional RPV clients added or deleted are not recognized until the command is started again.
- The **-i** interval is the time, in seconds, between each successive gathering and display of RPV statistics in monitor mode. This interval is not a precise measure of the elapsed time between each successive updated display. The **rpvstat** command obtains some of the information it displays by calling system services and has no control over the amount of time these services take to complete their processing. Larger numbers of RPVs will result in the **rpvstat** command taking longer to gather information and

will elongate the time between successive displays in monitor mode, sometimes taking much longer than the **-i** interval between displays.

- The count of reads and writes is accumulated on a per buffer basis. This means that if an application I/O passes a vector of buffers in a single read or write call, then instead of counting that read or write as a single I/O, it is counted as the number of buffers in the vector.
- The count of completed and pending I/O kilobytes is truncated. Any fractional amount of a KB is dropped in the output display.
- The **cx** field in the output displays one of the following connection status:

Table 15. cx output	
Field	Description
A number	This number is the count of active network connections between the RPV Client and its RPV Server.
Y	Indicates the connection represented by the IP address is available and functioning.
N	Indicates the connection represented by the IP address is not available.
X	Indicates the required information could not be retrieved from the device driver. Reasons for this status can be that the device driver is not loaded, the device is not in the available state, or the device has been deleted.

Exit Status

This command returns the following exit values:

Table 16. Exist status	
Field	Description
0	No errors.
>0	An error occurred.

Examples

1. To display statistical information for all RPV clients, enter the following command:

```
rpvstat
```

2. To display statistical information for RPV client **hdisk14**, enter the following command:

```
rpvstat hdisk14
```

3. To reset the statistical counters in RPV client **hdisk23**, enter the following command:

```
rpvstat -R hdisk23
```

4. To display statistical information for RPV client **hdisk14** and repeat the display every 30 seconds for 12 times, enter the following command:

```
rpvstat hdisk14 -i 30 -c 12
```

5. To display statistical information for all RPV clients and include detailed information by mirroring network, enter the following command:

```
rpvstat -n
```

6. To display statistical information for all mirroring networks, enter the following command:

```
rpvstat -N
```

7. To display statistical information on maximum pending values for all RPV clients, enter the following command:

```
rpvstat -m
```

8. **>|**To display statistical information about asynchronous I/O groups, enter the following command:

```
rpvstat -G
```

|<

9. **>|**To display summary of statistical information about asynchronous I/O groups, enter the following command:

```
rpvstat -g
```

|<

Files

The `/usr/sbin/rpvstat` path contains the **rpvstat** command.

>|rpvutil Command

Purpose

Configures a mirror pool in the remote physical volume (RPV) client of Geographic Logical Volume Manager (GLVM).

Syntax

```
rpvutil -h [tunable_name]  
rpvutil [-v vg_name] {-a | -o tunable_name[=value] >| -c}  
rpvutil -d tunable_name >|  
rpvutil -s rpvs_name  
rpvutil -r rpvs_name |<
```

Description

Starting with IBM AIX 7.2 with Technology Level 5, you can use the **rpvutil** command to perform the following operations:

- Set the maximum expected delay before the application receives the I/O acknowledgments for a mirror pool in a volume group.
- Compress the I/O data packet before it is sent from the RPV client to the RPV server or for monitoring the RPV network.
- Monitor the RPV data network.
- **>|**Change the cache size of the asynchronous I/O operations. You must increase the cache size by using the **extendlv** command before you run the **rpvutil** command. **|<**
- **>|**Suspend and resume operations on the RPV server. **|<**
- **>|**Enable or disable the cache full timer feature. However, if you disable the cache full timer feature, the buffered I/O operations are invalidated. **|<**
- **>|**Set the cache full timer value. **|<**

>| The data in a remote site with GLVM configuration might become inconsistent in the following scenarios:

- When the remote disk is not accessible during a write operation on the RPV server, but the remote disk is accessible during a site failover.
- When a site failover occurs after the **syncvg** command fails on remote disk.
- When the local site or the network fails during the execution of the **syncvg** command.

|<

>| To handle these data inconsistency issues at the remote site, you can use the **rpvutil** command to perform the following tasks:

1. Suspend incoming I/O requests at the remote site.
2. Periodically save disk data snapshots on remote site's storage disks or on cloud by using external utilities.
3. Resume I/O operations at the remote site.

|<

>| During an unplanned site failover, if data in the remote site becomes inconsistent, restore snapshot data on remote disks by using external utilities. |<

Flags

-a

Displays the current values of all tunable parameters of the GLVM RPV client.

>|-c

Changes the cache size of the asynchronous I/O operations. |<

-d **tunable_name**

Resets the specified tunable parameter to its default value.

-h [**tunable_name**]

Displays help information for the **rpvutil** command. If you specify a tunable parameter with this flag, the command displays help information for the specified tunable parameter.

>|

nw_sessions=number_of_sessions

Specifies the number of parallel RPV sessions (sender and receiver threads) per network. This tunable parameter is used to increase the number of parallel RPV sessions per GLVM network, which helps in sending the data in parallel and to improve the data transfer rate and use full bandwidth between the sites. A single RPV session on an RPV client consists of a sender thread for transfer of the data, and a receiver thread to receive the acknowledgment. Range of the **nw_sessions** is 1 - 99. You can change RPV value only when all the RPV devices on the AIX operating system are in defined state.

Note: When you change the number of sessions value to a higher value, you might create many threads. You must choose number of sessions value based on the workload.

cf_tmr_feature={1/0}

Enables or Disables the cache full timer feature. If this feature is enabled, the **rpvutil** command starts a timer when the I/O buffer cache is full. When the I/O buffer cache is full, all the subsequent I/O requests are buffered internally. When the timer expires, all the I/O requests that are buffered internally are invalidated and the threads of the application are released from the queue of threads that are waiting for the response from the I/O operations. Also, the physical volumes are moved to the stale state. You can set the timer value by using the **cf_tmr_value** tunable parameter.

cf_tmr_value=value_in_seconds

Sets the timeout value for the cache full timer feature. The timeout value can be in the range 2 - 30 seconds. The default timeout value is 10 seconds.



-o tunable_name[=value]

If you do not specify any value for the tunable parameter, the **-o tunable_name** flag displays the current values of the specified tunable parameter. You can specify the following tunable parameters for the **rpvutil -o** command:

compression=1|0

To use the compression tunable parameter, consider the following prerequisites:

- Ensure that the RPV client and the RPV server are running AIX version 7.2.5, or later, with all the latest RPV device drivers.
- Ensure that both the RPV server and the RPV client are IBM Power® Systems servers with NX842 acceleration units.
- Ensure that the compression tunable parameter is enabled on both the RPV server and RPV client so that the I/O data packets are compressed when the workload is failed over between the RPV client and the RPV server.

When you set the **compression** tunable parameter to 1, the **rpvutil** command compresses the I/O data packet before it is sent from the RPV client to the RPV server by using the special acceleration units for cryptography and compression (NX842) that are available in IBM Power Systems servers. If the I/O data packet is compressed successfully, a flag is set in the data packet that is sent to indicate that the data packet is compressed. If the I/O data packet that is received at the RPV server has the compression flag enabled, the RPV server decompresses the I/O data packet. If the NX842 acceleration unit is not available in the RPV server, the RPV server attempts a software decompression operation instead of the hardware decompression operation. By default, this option is set to 0 (disabled).

io_grp_latency=timeout_value

Indicates the maximum expected delay, in milliseconds, before receiving the I/O acknowledgment for a mirror pool that is configured in asynchronous mode. You must specify the volume group that is associated to the mirror pool by using the **-v** flag. The default delay value is 10 ms. You can specify lesser values to improve I/O performance but CPU consumption might increase.

>| -r rpv_name

Resume I/O operations on the volume group to which the `rpverver` device belongs.

>| -s rpv_name

Suspend I/O operations on the volume group to which the `rpverver` device belongs.

-v vg_name

Specifies the name of the volume group name for which the tunable parameters are added, modified, or displayed.

Examples

- To set a timeout value of 5 ms for the `gmvg1` volume group, enter the following command:

```
rpvutil -v gmvg1 -o io_grp_latency=5
```

An output that is similar to the following example is displayed:

```
RPVC timeout: 10. ODM Value: 10
Updated timeout[5] to ODM VG: gmvg1
Updated timeout[5] to rpvc. VG: gmvg1
```

- To check the value of the `compression` option, enter the following command:

```
rpvutil -o compression
```

An output that is similar to the following example is displayed:

```
compression[odm,driver]=ENABLED,ENABLED
```

- To compress the I/O data packets before the I/O data packets are sent from the RPV client to the RPV server, enter the following command:

```
rpvutil -o compression=1
```

The command returns 0 after successful completion.

- To set the network sessions, enter the following command:

```
rpvutil -o nw_sessions=2
```

An output that is similar to the following example is displayed:

```
Setting nw_sessions to 2
```

- To display the current value of tunable parameter, **nw_sessions**, enter the following command:

```
rpvutil -o nw_sessions
```

An output that is similar to the following example is displayed:

```
nw_sessions = 1
```

- [|<](#) To apply cache size of the asynchronous I/O operation on a volume group, enter the following command:

```
rpvutil -c -v vg_name
```

[|<](#)

- [|<](#) To suspend I/O operations on a volume group that contains the `rpvserver0` device, enter the following command:

```
rpvutil -s rpvserver0
```

[|<](#)

- [|<](#) To resume I/O operations on a volume group that contains the `rpvserver0` device, enter the following command:

```
rpvutil -r rpvserver0
```

[|<](#)

- [|<](#) To enable the cache full timer feature, enter the following command:

```
rpvutil -v agmvg -o cf_tmtr_feature=1
```

[|<](#)

- [|<](#) To set the cache full timer value to 20 seconds, enter the following command:

```
rpvutil -v agmvg -o cf_tmtr_value=20
```

[|<](#)

Files

The `/usr/lib/methods` directory contains the **rpvutil** command.

[|<](#)

rrestore Command

Purpose

Copies previously backed up file systems from a remote machine's device to the local machine.

Syntax

```
rrestore [ -bNumber ] [ -h ] [ -i ] [ -m ] [ -sNumber ] [ -t ] [ -v ] [ -y ] [ -x ] [ -r ] [ -R ] -fMachine:Device  
[ FileSystem ... ] [ File ... ]
```

Description

The **rrestore** command restores Version 3 by i-node backups from a remote machine's device to a file system on the local machine. The **rrestore** command creates a server on the remote machine to the backup medium.

The **rrestore** command only accepts backup formats created when a file system is backed up by i-node.

Note: A user must have root authority to execute this command.

Flags

Item	Description
-b <i>Number</i>	Specifies the number of blocks to read in a single input operation. If you do not specify this flag, the rrestore command selects a default value appropriate for the physical device you have selected. Larger values of the <i>Number</i> variable result in larger physical transfers from tape devices.
-f <i>Machine:Device</i>	Specifies the input device on the remote machine. Specify the <i>Device</i> variable as a file name (such as the /dev/rmt0 file) to get input from the named device. For more information on using tape devices see the rmtspecial file.
-h	Restores only the actual directory named by the <i>File</i> parameter, not the files contained in that directory. This option is ignored when either the -r or -R flag is specified.

Item	Description
-i	Starts the interactive mode. This flag allows you to restore selected files from the directory represented by the <i>File</i> parameter. The subcommands for the -i flag are: ls [<i>Directory</i>] Displays directory names within the specified <i>Directory</i> parameter with a / (slash) after the name, and displays files to be restored with an * (asterisk) before the name. If the -v flag is used, the i-node number of each file and directory is also displayed. If the <i>Directory1</i> parameter is not specified, the current directory is used. cd <i>Directory</i> Changes the current directory to the <i>Directory</i> parameter. pwd Displays the full path name of the current directory. add [<i>File</i>] Specifies the <i>File</i> parameter to restore. If the <i>File</i> parameter is a directory, that directory and all its files are restored (unless the -h flag is used). Files to be restored are displayed with an * (asterisk) before the name by the ls subcommand. If the <i>File</i> parameter is not specified, the current directory is used. delete [<i>File</i>] Specifies the <i>File</i> parameter to ignore in restore. If the <i>File</i> parameter is a directory, the directory and all its files are not restored (unless the -h flag is used). If the <i>File</i> parameter is not specified, the current directory is used. extract Restores all files displayed with an * (asterisk) before the name by the ls subcommand. setmodes Sets owner, modes, and times for the files being restored rather than using this information as it resides on the backup medium. verbose Displays the i-node numbers of all restored files with the ls subcommand. Information about each file is also displayed as it is restored. The next invocation of the verbose subcommand turns verbose off. help Displays a summary of the subcommands. quit Stops execution of the rrestore command immediately, even if all files requested have not been restored.
-m	Restores files by i-node number rather than by path name.
-r	Restores an entire file system.



Attention: If you do not follow this procedure carefully, you can ruin an entire file system. If you are restoring a full (level 0) backup, run the **mkfs** command to create an empty file system before doing the restore. To restore an incremental backup at level 2, for example, run the **mkfs** command, restore the appropriate level 0 backup, restore the level 1 backup, and finally restore the level 2 backup. As an added safety precaution, run the **fsck** command after you restore each backup level.

Item	Description
-R	Causes the rrestore command to request a specific volume in a multivolume set of backup medium when restoring an entire file system. The -R flag provides the ability to interrupt and resume the rrestore command.
-sNumber	Specifies which backup to restore from a multibackup medium. Numbering starts with 1.
-t	Displays the table of contents for the backed up files. The rrestore command displays the file name. The names are relative to the root (/) directory of the file system backed up. The only exception is the root (/) directory itself.
-v	Reports the progress of the restoration as it proceeds.
-x	Restores individually named files. If no names are given, all files on that medium are restored. The names must be in the same form as the names shown by the -t flag.
-y	Prevents the rrestore command from asking whether it should stop the restore if a tape error is encountered. The rrestore command attempts to skip over bad blocks.
-?	Displays the usage message.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To list files from a remote tape device, enter:

```
rrestore -fmachine1:/dev/rmt0 -t
```

This command reads information from the /dev/rmt0 device on remote machine1. The file names are shown.

2. To restore files, enter:

```
rrestore -x -fmachine1:/dev/rmt0 /home/mike/file1
```

This command extracts the /home/mike/file1 file from the backup medium on the /dev/rmt0 device on remote machine1.

3. To restore all the files in a directory, enter:

```
rrestore -fhost:/dev/rmt0 -x /home/mike
```

This command restores the directory /home/mike and all the files it contains.

4. To restore a directory, but not the files in the directory, enter:

```
rrestore -fhost:/dev/rmt0 -x -h /home/mike
```

5. To restore all the files in a directory from a specific backup on a multibackup medium, enter:

```
rrestore -s3 -fhost:/dev/rmt0.1 -x /home/mike
```

This command restores the /home/mike directory and all the files it contains from the third backup on the backup medium.

Files

Item	Description
/dev/rfd0	Specifies the default restore device.
/usr/sbin/rrestore	Contains the rrestore command.

Rsh command

Purpose

Starts the restricted version of the Bourne shell.

Syntax

Rsh [-i] [{ + | - } { [-a] [-e] [-f] [-h] [-k] [-n] [-t *timeout*] [-u] [-v] [-x] }] [-c *String* | -s | *File* [*Parameter*]

Note: Preceding a flag with a + (plus sign) rather than a - (minus sign) turns it off.

Description

The **Rsh** command starts a restricted version of the Bourne shell, which is useful for installations that require a more controlled shell environment. You can create user environments with a limited set of privileges and capabilities.

Flags

The Bourne shell interprets the following flags only when the shell is started at the command line.

Note: Unless you specify either the **-c** or **-s** flag, the shell assumes that the next parameter is a command file (shell script). It passes anything else on the command line to that command file.

Item	Description
-a	Marks for export all variables to which an assignment is performed. If the assignment precedes a command name, the export attribute is effective only for that command's execution environment, except when the assignment precedes one of the special built-in commands. In this case, the export attribute persists after the built-in command is completed. If the assignment does not precede a command name, or if the assignment is a result of the operation of the getopts or read command, the export attribute persists until the variable is unset.
-c <i>String</i>	Runs commands that are read from the <i>String</i> variable. Sets the value of special parameter 0 from the value of the <i>String</i> variable and the positional parameters (\$1, \$2, and so on) in sequence from the remaining parameter operands. The shell does not read additional commands from standard input when you specify this flag.

Item	Description
-e	Exits immediately if all of the following conditions exist for a command: • It exits with a return value greater than 0. • It is not part of the compound list of a while, until, or if command. • It is not being tested by using AND or OR lists. • It is not a pipeline that is preceded by the ! (exclamation point) reserved word.
-f	Disables file name substitution.
-h	Locates and remembers the commands that are called within functions as the functions are defined. (Usually these commands are located when the function is run; see the hash command.)
-i	Makes the shell interactive, even if input and output are not from a workstation. In this case, the shell ignores the TERMINATE signal, so that the kill 0 command does not stop an interactive shell, and traps an INTERRUPT signal, so you can interrupt the function of the wait command. In all cases, the shell ignores the QUIT signal.
-k	Places all keyword parameters in the environment for a command, not just those preceding the command name.
-n	Reads commands but does not run them. The -n flag can be used to check for shell-script syntax errors. An interactive shell might ignore this option.
-s	Reads commands from standard input. Any remaining parameters that are specified are passed as positional parameters to the new shell. Shell output is written to standard error, except for the output of built-in commands.
-t timeout	Exits after the timeout seconds if there is no reply from the server.
-u	Treats an unset variable as an error and immediately exits when it performs variable substitution. An interactive shell does not exit.
-v	Displays shell input lines as they are read.
-x	Displays commands and their arguments before they are run.

Note: Using a + (plus sign) rather than a - (minus sign) unsets flags. The \$- special variable contains the current set of flags.

Files

Item	Description
/usr/bin/bsh	Specifies the path name to the Bourne shell.
/usr/bin/Rsh	Specifies the path name to the restricted Bourne shell, a subset of the Bourne shell.
/tmp/sh*	Contains temporary files that are created when a shell is opened.

rsh or remsh Command

Purpose

Executes the specified command at the remote host or logs in to the remote host.

Syntax

```
{ rsh | remsh } RemoteHost [ -n ] [ -l User ] [ -f | -F ] [ -k realm ] [ -S ] [ -u ] [ Command ]
```

Description

The **/usr/bin/rsh** command executes the command specified by the *Command* parameter at the remote host specified by the *RemoteHost* parameter; if the *Command* parameter is not specified, the **rsh** command logs into the remote host specified by the *RemoteHost* parameter. The **rsh** command sends standard input from the local command line to the remote command and receives standard output and standard error from the remote command.

Note: Because any input to the remote command must be specified on the local command line, you cannot use the **rsh** command to execute an interactive command on a remote host. If you need to execute an interactive command on a remote host, use either the **rlogin** command or the **rsh** command without specifying the *Command* parameter. If you do not specify the *Command* parameter, the **rsh** command executes the **rlogin** command instead.

Access Files

If you do not specify the **-l** flag, the local user name is used at the remote host. If **-l User** is entered, the specified user name is used at the remote host.

Using Standard Authentication

The remote host allows access only if at least one of the following conditions is satisfied:

- The local user ID is not the root user, and the name of the local host is listed as an equivalent host in the remote **/etc/hosts.equiv** file.
- If either the local user ID is the root user or the check of **/etc/hosts.equiv** is unsuccessful, the remote user's home directory must contain a **\$HOME/.rhosts** file that lists the local host and user name.

Although you can set any permissions for the **\$HOME/.rhosts** file, it is recommended that the permissions of the **.rhosts** file be set to 600 (read and write by owner only).

In addition to the preceding conditions, the **rsh** command also allows access to the remote host if the remote user account does not have a password defined. However, for security reasons, use of a password on all user accounts is recommended.

For Kerberos 5 Authentication

The remote host allows access only if all of the following conditions are satisfied:

- The local user has current DCE credentials.
- The local and remote systems are configured for Kerberos 5 authentication (On some remote systems, this method is not necessary. It is necessary that a daemon is listening to the klogin port).
- The remote system accepts the DCE credentials as sufficient for access to the remote account. See the **kvalid_user** function for more information.

Remote Command Execution

When the remote command is run, pressing the Interrupt, Terminate, or Quit key sequences sends the corresponding signal to the remote process. However, pressing the Stop key sequence stops only the local process. Usually, when the remote command terminates, the local **rsh** process terminates.

To have shell metacharacters interpreted on the remote host, place the metacharacters inside " " (double quotation marks). Otherwise, the metacharacters are interpreted by the local shell.

When using the **rsh** command, you can create a link to a path (to which you have permission to write), by using a host name that is specified by the *HostName* parameter as the link name. For example:

```
ln -s /usr/bin/rsh HostName
```

After the link is established, you can specify the *HostName* parameter and a command that is specified by the *Command* parameter from the command line. The **rsh** command remotely runs the command on the remote host. The syntax is:

For example, if you are linked to remote host *opus* and want to run the **date** command, enter:

```
opus date
```

Because you can not specify the **-l User** flag, the remote command is successful only if the local user has a user account on the remote host. Otherwise, the **rsh** command returns a `Login incorrect` error message. When you specify the *HostName* parameter without a command, the **rsh** command calls the **rlogin** command, which logs you into the remote host. Again, for successful login, the local user must have a user account on the remote host.

Flags

-a

Indicates that the standard error of the remote command is the same as standard output. No provision is made for sending arbitrary signals to the remote process.

-f

Causes the credentials to be forwarded. This flag is ignored if Kerberos 5 is not the current authentication method. Authentication fails if the current DCE credentials are not marked forwardable.

-F

Causes the credentials to be forwarded. In addition the credentials on the remote system is marked forwardable (allowing them to be passed to another remote system). This flag is ignored if Kerberos 5 is not the current authentication method. Authentication fails if the current DCE credentials are not marked forwardable.

-k realm

Allows the user to specify the realm of the remote station if it is different from the local systems realm. For these purposes, a realm is synonymous with a DCE cell. This flag is ignored if Kerberos 5 is not the current authentication method.

-l User

Specifies that the **rsh** command must log in to the remote host as the user specified by the *User* variable instead of the local user name. If this flag is not specified, the local and remote user names are the same.

-n

Specifies that the **rsh** command must not read from standard input.

-S

Secure option, force remote IP address of the standard error connection to be the same as the standard output connection.

-u

Use standard AIX authentication only.

Exit Status

This command returns the following exit values:

0

Successful completion.

>0

An error occurred.

Security

The remote host allows access only if at least one of the following conditions is satisfied:

- The local user ID is listed as a principal in the authentication database and had performed a **kinit** to obtain an authentication ticket.
- If a **\$HOME/.klogin** file exists, it must be in the local user's **\$HOME** directory on the target system. The local user and any user must be listed or the services that are allowed to the **rsh** command is considered. This file performs a similar function to a local **.rhosts** file. Each line in this file must contain a principal in the form of *principal.instance@realm*. If the originating user is authenticated as one of the principals that are named in the **.klogin** file, access is granted to the account. The owner of the account is granted access if the **.klogin** file is not present.

For security reasons, any **\$HOME/.klogin** file must be owned by the remote user and only the AIX owner ID has read and write access (permissions = 600) to the **.klogin** file.

Attention RBAC users: This command can run privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

In the following examples, the local host, **host1**, is listed in the **/etc/hosts.equiv** file at the remote host, **host2**.

1. To check the amount of free disk space on a remote host, enter:

```
rsh host2 df
```

The amount of free disk space on **host2** is displayed on the local system.

2. To append a remote file to another file on the remote host, place the **>>** metacharacters in quotation marks, and enter:

```
rsh host2 cat test1 ">>" test2
```

The file **test1** is appended to **test2** on remote host **host2**.

3. To append a remote file at the remote host to a local file, omit the quotation marks, and enter:

```
rsh host2 cat test2 >> test3
```

The remote file **test2** on **host2** is appended to the local file **test3**.

4. To append a remote file to a local file and use a remote user's permissions at the remote host, enter:

```
rsh host2 -l jane cat test4 >> test5
```

The remote file **test4** is appended to the local file **test5** at the remote host, with user **jane**'s permissions.

5. This example shows how the root user can issue an **rcp** on a remote host when the authentication is Kerberos 4 on both the target and server. The root user must be in the authentication database and must have already issued **kinit** on the local host. The command is issued at the local host to copy the file, **stuff**, from node **r05n07** to node **r05n05** on an SP.

```
/usr/lpp/ssp/rcmd/bin/rsh r05n07 'export KRBTKFILE=/tmp/rcmdtktk$$; \  
/usr/lpp/ssp/rcmd/bin/rcmdtgt; \  
/usr/lpp/ssp/rcmd/bin/rcp /tmp/stuff r05n05:/tmp/stuff;'
```

The root user sets the **KRBTKFILE** environment variable to the name of a temporary ticket-cache file and then obtains a service ticket by issuing the **rcmdtgt** command. The **rcp** uses the service ticket to authenticate from host **r05n07** to host **r05n05**.

Files

Item	Description
\$HOME/.klogin	Specifies remote users that can use a local user account.
/usr/lpp/spp/rcmd/bin/rsh	Link to AIX Secure /usr/bin/rsh that calls the SP Kerberos 4 rsh routine if applicable.
/usr/lpp/spp/rcmd/bin/remsh	Link to AIX Secure /usr/bin/rsh that calls the SP Kerberos 4 rsh routine if applicable.

Prerequisite Information

Refer to the chapter on security in IBM Parallel System Support Programs for AIX: *Administration Guide* for an overview. You can access this publication at the following Web site: http://www.rs6000.ibm.com/resource/aix_resource

Refer to the "RS/6000 SP Files and Other Technical Information" section of IBM Parallel System Support Programs for AIX: Command and Technical Reference for additional Kerberos information. You can access this publication at the following Web site: http://www.rs6000.ibm.com/resource/aix_resource

rshd Daemon

Purpose

Provides the server function for remote command execution.

Syntax

Note: The **rshd** daemon is usually started by the **inetd** daemon. It can also be controlled from the command line, using SRC commands.

/usr/sbin/rshd [**-c**] [**-s**] [**p**]

Description

The **/usr/sbin/rshd** daemon is the server for the **rcp** and **rsh** commands. The **rshd** daemon provides remote execution of shell commands. These commands are based on requests from privileged sockets on trusted hosts. The shell commands must have user authentication. The **rshd** daemon listens at the socket defined in the **/etc/services** file.

Changes to the **rshd** daemon can be made using the System Management Interface Tool (SMIT) or System Resource Controller (SRC), by editing the **/etc/inetd.conf** or **/etc/services** file. Entering **rshd** at the command line is not recommended. The **rshd** daemon is started by default when it is uncommented in the **/etc/inetd.conf** file.

The **inetd** daemon get its information from the **/etc/inetd.conf** file and the **/etc/services** file.

After changing the **/etc/inetd.conf** or **/etc/services** file, run the **refresh -s inetd** or **kill -1 InetdPID** command to inform the **inetd** daemon of the changes to its configuration file.

Service Request Protocol

When the **rshd** daemon receives a service request, it initiates the following protocol:

1. The **rshd** daemon checks the source port number for the request. If the port number is not in the range 512 through 1023, the **rshd** daemon terminates the connection.
2. The **rshd** daemon reads characters from the socket up to a null byte. The string read is interpreted as an ASCII number (base 10). If this number is nonzero, the **rshd** daemon interprets it as the port number of a secondary stream to be used as standard error. A second connection is created to the

specified port on the client host. The source port on the local host is also in the range 512 through 1023.

3. The **rshd** daemon uses the source address of the initial connection request to determine the name of the client host. If the name cannot be determined, the **rshd** daemon uses the dotted decimal representation of the client host's address.
4. The **rshd** daemon retrieves the following information from the initial socket:
 - A null-terminated string of at most 16 bytes interpreted as the user name of the user on the client host.
 - A null-terminated string of at most 16 bytes interpreted as the user name to be used on the local server host.
 - Another null-terminated string interpreted as a command line to be passed to a shell on the local server host.
5. The **rshd** daemon attempts to validate the user using the following steps:
 - a. The **rshd** daemon looks up the local user name in the **chdir** subroutine). If either the lookup or the directory change fails, the **rshd** daemon terminates the connection.
 - b. If the local user ID is a nonzero value, the **rshd** daemon searches the **/etc/hosts.equiv** file to see if the name of the client workstation is listed. If the client workstation is listed as an equivalent host, the **rshd** daemon validates the user.
 - c. If the **\$HOME/.rhosts** file exists, the **rshd** daemon tries to authenticate the user by checking the **.rhosts** file.
 - d. If either the **\$HOME/.rhosts** authentication fails or the client host is not an equivalent host, the **rshd** daemon terminates the connection.
6. After the **rshd** daemon validates the user, the **rshd** daemon returns a null byte on the initial connection and passes the command line to the user's local login shell. The shell then inherits the network connections established by the **rshd** daemon.

The **rshd** daemon should be controlled using the System Management Interface Tool (SMIT) or by changing the **/etc/inetd.conf** file. Typing **rshd** at the command line is not recommended.

Manipulating the rshd Daemon with the System Resource Controller

The **rshd** daemon is a subserver of the **inetd** daemon, which is a subsystem of the System Resource Controller (SRC). The **rshd** daemon is a member of the **tcpip** SRC subsystem group. This daemon is enabled by default in the **/etc/inetd.conf** file and can be manipulated by the following SRC commands:

Item	Description
startsrc	Starts a subsystem, group of subsystems, or a subserver.
stopsrc	Stops a subsystem, group of subsystems, or a subserver.
lssrc	Gets the status of a subsystem, group of subsystems, or a subserver.

Flags

Item	Description
c	Suppresses the sanity check of a host name lookup.
p	Runs your <i>.profile</i> file whenever you issues the rsh command in the non-interactive mode. Without this flag, your <i>.profile</i> file is not run in case of the rsh command in the non-interactive mode.
s	Turns on socket-level debugging.

Security

The **rshd** daemon is a PAM-enabled application with a service name of *rsh*. System-wide configuration to use PAM for authentication is set by modifying the value of the **auth_type** attribute, in the **usw** stanza of the **/etc/security/login.cfg** file, to the **PAM_AUTH** attribute as the root user.

The authentication mechanisms used when PAM is enabled depend on the configuration for the *rsh* service in the **/etc/pam.conf** file. The **rshd** daemon requires the **/etc/pam.conf** entries for the **auth**, **account**, and **session** module types. Following is the recommended configuration in the **/etc/pam.conf** file for the *rsh* service:

```
#
# AIX rsh configuration
#
rsh auth      sufficient  /usr/lib/security/pam_rhosts_auth
rsh account   required   /usr/lib/security/pam_aix
rsh session   required   /usr/lib/security/pam_aix
```

Examples

Note: The arguments for the **rshd** daemon can be specified by using SMIT or by editing the **/etc/inetd.conf** file.

1. To start the **rshd** daemon, type the following:

```
startsrc -t shell
```

This command starts the **rshd** subserver.

2. To stop the **rshd** daemon, type the following:

```
stopsrc -t shell
```

This command allows all pending connections to start and existing connections to complete but prevents new connections from starting.

3. To force stop the **rshd** daemon and all **rshd** connections, type the following: :

```
stopsrc -t -f shell
```

This command terminates all pending connections and existing connections immediately.

4. To display a short status report about the **rshd** daemon, type the following: :

```
lssrc -t shell
```

This command returns the daemon's name, process ID, and state (active or inactive).

rstatd Daemon

Purpose

Returns performance statistics obtained from the kernel.

Syntax

/usr/sbin/rpc.rstatd

Description

The **rstatd** daemon is a server that returns performance statistics obtained from the kernel. The **rstatd** daemon is normally started by the **inetd** daemon.

Files

Item	Description
<code>/etc/inetd.conf</code>	TCP/IP configuration file that starts RPC daemons and other TCP/IP daemons.
<code>/etc/services</code>	Contains an entry for each server available through Internet.

rsyslogd Daemon

Purpose

Logs system messages.

Description

The **rsyslogd** daemon reads a socket and sends the message line to a destination that is specified by the `/etc/rsyslog.conf` configuration file. The **rsyslogd** daemon reads the configuration file when it is activated. You can start the **rsyslogd** daemon from the source master by using the following commands:

```
startsrc -s rsyslogd
stopsrc -s rsyslogd
```

The `startsrc` option starts the **rsyslogd** daemon. To start multiple **rsyslogd** daemons, run the `startsrc` option repeatedly with a new pid file by using the `-i` command-line option. The `startsrc` command specifies the arguments for the **rsyslogd** daemon by using the `startsrc -a` flag. The arguments must be protected from interpretation by the shell with double quotation marks.

The `stopsrc` option stops all instances of the **rsyslogd** daemon. To stop a specific instance, you must specify the `-p <pid>` option.

```
stopsrc -p <pid of rsyslogd daemon>
```

Default logging application:

After the **rsyslogd** daemon is installed, it cannot be started immediately and **syslogd** daemon continues to be used to log system messages. To configure the **rsyslogd** daemon to log messages by default, run the `syslog_ssw` script by using the `-r` option.

After the **rsyslogd** daemon is configured to log system messages, the **rsyslogd** daemon starts with a default command-line argument of `-c5`. This option ensures that the **rsyslogd** daemon starts in a normal mode and is not compatible with an earlier version.

Default `rsyslog.conf` file:

To configure and use the **rsyslogd** daemon, see the reference section of the documentation.

After installation, the default `/etc/rsyslog.conf` configuration file has the following information:

```
#####
# Rsyslog is free software: it is distributed under the      #
# terms of the GNU General Public License as published by   #
# the Free Software Foundation, under version 3 of the License. #
#                                                            #
# if you experience problems, check                         #
# http://www.rsyslog.com/doc/troubleshoot.html for assistance #
#                                                            #
# Load the UNIX socket for local communication              #
$ModLoad imuxsock                                           #
#                                                            #
# Load the UDP module for remote communication             #
$ModLoad imudp                                              #
#                                                            #
# Run the UDP server on the default port 514                 #
$UDPServerRun 514                                           #
```

```
#
#####
```

Almost all parameters in the `syslog.conf` file functions with the **rsyslogd** daemon except for the AIX specific parameters such as pureScale® API support. To convert a `syslog.conf` file into a supported `rsyslog.conf` file, the switching script must be used with the `-c` option.

Switching script usage

```
syslog_ssw [ -r | -s | -c SourceSyslogConffile DestRsyslogConffile ]
```

Item	Descriptor
-r	Switch to rsyslog daemon as the default logging application.
-s	Switch to syslog daemon as the default logging application.
-c	Convert configuration rules in the <code>syslog.conf</code> file to the rules in the <code>rsyslog.conf</code> file. However, the AIX specific parameters that are not understood by the rsyslogd daemon are removed during conversion.

When you switch the default logging application by using the `-r` or the `-s` option, this choice remains persistent across restart.

The `startsrc -s syslogd` command starts the **rsyslogd** or the **syslogd** daemon that is based on the default logging application that is set.

The `syslog_ssw` script is not present by default, and is available after the **rsyslogd** daemon is installed.

Examples

1. To stop the existing **syslogd** daemon and to start the **rsyslogd** daemon, run the following command:

```
syslog_ssw -r
```

2. To stop the existing **rsyslogd** daemon and to start the **syslogd** daemon, run the following command:

```
syslog_ssw -s
```

3. To convert the `syslog.conf` file to `rsyslog.conf` file, and to create an `rsyslog.conf` file if the file does not exist, run the following command:

```
syslog_ssw -c syslog.conf rsyslog.conf
```

This conversion removes the AIX specific parameters and allows the newly created file to be used with the **rsyslogd** daemon.

4. To start the default logging application, run the following command:

```
startsrc -s syslogd
```

The default logging application can be the **syslogd** daemon or the **rsyslogd** daemon.

Files

Item	Descriptor
/etc/rsyslog.conf	Controls the output of the rsyslogd daemon.
/etc/rsyslogd.pid	Contains the process ID.

rtcd Daemon

Purpose

Monitors the file modification events, checks for the resulting compliance violations, and alerts the administrators.

Description

The **rtcd** daemon reads the configuration information that is defined in the `/etc/security/rtc/rtcd.conf` file. The **rtcd** daemon runs the **aixpert** command to check for compliance violation during startup. It alerts the recipients who are specified in the `/etc/security/rtc/rtcd.conf` file by email if any violation is determined.

The **rtcd** daemon continuously monitors the files that are specified in the `/etc/security/rtc/rtcd_policy.conf` file for file changes. If any files change, the **rtcd** runs the **aixpert** command to check for the compliance violations and sends an alert email for any violations.

The **rtcd** daemon is placed under the SRC control after successful configuration of Real-Time Compliance. You must manage the **rtcd** daemon by using the System Resource Controller (SRC) commands.

Security

The **rtcd** daemon is owned by the root user and the system group. Only the root user and users with `aix.system.config.src` authorization are authorized to manage the command.

Examples

1. To start the **rtcd** daemon, enter the following command:

```
# startsrc -s rtcd
```

2. To check the **rtcd** daemon, enter the following command:

```
# lssrc -s rtcd
```

3. To stop the **rtcd** daemon, enter the following command:

```
# stopsrc -s rtcd
```

Files

Item	Description
/etc/security/rtc/rtcd_policy.conf	Contains the configuration information for the rtcd daemon.
/etc/security/rtc/rtcd.conf	Grants read (r) and write (w) access to the root user.

rtl_enable Command

Purpose

Relinks shared objects to enable the runtime linker to use them.

Syntax

```
rtl_enable [ -R | -o Name ] [ -l ] [ -s ] File [ ldFlag ... ] [ -F ObjsLibs ... ]
```

Description

The **rtl_enable** command relinks a module, or an archive containing modules, with the **-G** flag, to enable runtime linking. A module is an XCOFF file containing a loader section. A shared object is a module with the F_SHROBJ flag set in the XCOFF header.

In its simplest form, the **rtl_enable** command creates a new file with the name *File.new*. If *File* is a module, *File.new* will be the same kind of module. If *File* is an archive, *File.new* will be an archive whose members have the same names as the members of *File*. The **rtl_enable** command relinks the modules in the new archive to enable run-time linking. The **rtl_enable** command archives other members unchanged into the output file.

The **rtl_enable** command uses the loader section in *File* (or its members) to create import and export files, to determine the **libpath** information, and to determine the entry point.

Flags

Item	Description
-F <i>ObjsLibs</i> ...	Adds <i>ObjsLibs</i> to the beginning of the generated ld command. The <i>ObjsLibs</i> parameter is either an object file or a library (specified with the ld command's -l (lowercase L) flag). If you are enabling an archive, adds the <i>ObjsLibs</i> to the ld command for all shared objects in the archive.
-l	(Lowercase L) Leaves the import and export files in the current directory instead of deleting them. Import files have the suffix .imp and export files, the suffix .exp . The rtl_enable command adds the suffixes to the input file name if <i>File</i> is a module. It adds the suffixes to the names of members that are modules if <i>File</i> is an archive.
-o <i>Name</i>	Specifies an alternate output file name instead of <i>File.new</i> . Do not use this flag with the -R flag.
-R	Replaces the input file instead of creating a new file. It will not overwrite the input file if any errors occur. Do not use this flag with the -o flag.
-s	Generates a script of commands in the current directory that you can use to create a new output file or archive, but does not relink anything. It names the script <i>Base.sh</i> , where <i>Base</i> is the basename of the input file with any suffix stripped off. It writes generated import and export files in the current directory as well. You can modify the script and the import and export files to customize the output objects.

Parameters

Item	Description
<i>File</i>	Specifies the input file.

Item	Description
<i>ldFlag ...</i>	Copies the specified ld command flags to the end of the generated ld command, overriding default options. Note: Do not use the -o flag in the <i>ldFlag</i> parameter to name the output file. To specify an alternate output file name, use the rtl_enable command's -o Name flag.

Exit Status

This command returns the following exit values:

Ite m	Description
0	Successful completion.
>0	An error occurred.

Note: Depending on the error, some output files may have been created.

Security

Access Control: Any User

Auditing Events: N/A

Examples

To create a new version of **libc.a** with runtime linking enabled, enter:

1. Create a directory for runtime version by entering:

```
mkdir /tmp/rtllibs
```

2. Make /tmp/rtllibs your current directory by entering:

```
cd /tmp/rtllibs
```

3. To create the runtime version of libc.a with the same name, enter:

```
rtl_enable -o libc.a /lib/libc.a
```

To use this version of libc.a when linking programs, use **-L /tmp/rtllibs** with the **ld** command.

Files

Item	Description
/usr/bin/rtl_enable	Contains the rtl_enable command. This is a symbolic link to /usr/ccs/bin/rtl_enable .

runacct Command

Purpose

Runs daily accounting.

Syntax

```
/usr/sbin/acct/runacct [ mdd [ State ] ]
```


Description

The **runacct** command is the main daily accounting shell procedure. Normally initiated by the **cron** daemon, the **runacct** command processes connect, fee, disk, queuing system (printer), and process accounting data files for the current day to produce the binary daily report, **/var/adm/acct/nite(x)/dayacct**. The **runacct** command also prepares summary files for the **prdaily** procedure to prepare the ASCII daily report, **/var/adm/acct/sum(x)/rprtmmdd**, or for billing purposes.

The **acctmerg** command adds the **dayacct** report to the cumulative summary report for the accounting period, **/var/adm/acct/sum(x)/tacct**. The **tacct** report is used by the **monacct** command to produce the monthly report, **/var/adm/acct/fiscal(x)**.

This command has two parameters that must be entered from the keyboard should you need to restart the **runacct** procedure. The date parameter, *mmdd*, enables you to specify the day and month for which you want to rerun the accounting. The *State* parameter enables a user with administrative authority to restart the **runacct** procedure at any of its states. For more information on restarting **runacct** procedures and on recovering from failures.

The **runacct** command protects active accounting files and summary files in the event of run-time errors, and records its progress by writing descriptive messages into the **/var/adm/acct/nite(x)/active** file. When the **runacct** procedure encounters an error, it sends mail to users root and adm, and exits.

The **runacct** procedure also creates two temporary files, **lock** and **lock1**, in the directory **/var/adm/acct/nite(x)**, which it uses to prevent two simultaneous calls to the **runacct** procedure. It uses the **lastdate** file (in the same directory) to prevent more than one invocation per day.

The **runacct** command breaks its processing into separate, restartable states. As it completes each state, it writes the name of the next state in the **/var/adm/acct/nite(x)/state** file. The **runacct** procedure processes the various states in the following order:

State	Actions
SETUP	Moves the active accounting files to working files and restarts the active files.
WTMPFIX	Verifies the integrity of the wtmp file, correcting date changes if necessary.
CONNECT1	Calls the acctcon1 command to produce connect session records.
CONNECT2	Converts connect session records into total accounting records (tacct.h format).
PROCESS	Converts process accounting records into total accounting records (tacct.h format).
MERGE	Merges the connect and process total accounting records.
FEES	Converts the output of the chargefee command into total accounting records (tacct.h format) and merges them with the connect and process total accounting records.
DISK	Merges disk accounting records with connect, process, and fee total accounting records.
QUEUEACCT	Sorts the queue (printer) accounting records, converts them into total accounting records (tacct.h format), and merges them with other total accounting records.
MERGETACCT	Merges the daily total accounting records in the daytacct report file with the summary total accounting records in the /var/adm/acct/sum(x)/tacct report file.
CMS	Produces command summaries in the file /var/adm/acct/sum(x)/cms .
USEREXIT	If the /var/adm/siteacct shell file exists, calls it at this point to perform site-dependent processing.
CLEANUP	Deletes temporary files and exits.

Restarting runacct Procedures

To restart the **runacct** command after a failure, first check the **/var/adm/acct/nite(x)/active** file for diagnostic messages, then fix any damaged data files, such as **pacct** or **wtmp**. Remove the **lock** files

and **lastdate** file (all in the **/var/adm/acct/nite(x)** directory), before restarting the **runacct** command. You must specify the **mmdd** parameter if you are restarting the **runacct** command. It specifies the month and day for which the **runacct** command is to rerun the accounting. The **runacct** procedure determines the entry point for processing by reading the **/var/adm/acct/nite(x)/statefile** file. To override this default action, specify the desired **state** on the **runacct** command line.

It is not usually a good idea to restart the **runacct** command in the **SETUP state**. Instead, perform the setup actions manually and restart accounting with the **WTMPFIX** state, as follows:

```
/usr/lib/acct/runacct mmdd WTMPFIX
```

If the **runacct** command fails in the **PROCESS** state, remove the last **ptacct** file, because it will be incomplete.

Flags

Item	Description
-X	Processes all available characters for each user name instead of truncating to the first 8 characters. The -X flag will also cause the runacct command and all commands it calls to use the /var/adm/acct/sumx and /var/adm/acct/nitex directories instead of the /var/adm/acct/sum and /var/adm/acct/nite directories.

Security

Access Control: This command should grant execute (x) access only to members of the **adm** group.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To start daily accounting procedures for system resources, add the following command line to a **crontab** file so the **runacct** command will be run automatically by the **cron** daemon:

```
0 4 * * 1-6 /usr/sbin/acct/runacct 2> \  
/var/adm/acct/nite/accterr
```

To start daily accounting procedures with long user name support add the following line to the crontab file:

```
0 4 * * 1-6 /usr/sbin/acct/runacct -X 2> \  
/var/adm/acct/nitex/accterr
```

This example shows the instructions that the **cron** daemon will read and act upon. The **runacct** command will run at 4 a.m. (04) every Monday through Saturday (1-6) and write all standard error output (2>) to the **/var/adm/acct/nite(x)/accterr** file. This command is only one of the accounting instructions normally given to the **cron** daemon.

2. To start daily accounting procedures for system resources from the command line (start the **runacct** command), enter the following:

```
nohup /usr/sbin/acct/runacct 2> \  
/var/adm/acct/nite/accterr &
```

Although it is preferable to have the **cron** daemon start the **runacct** procedure automatically (see example 1), you can give the command from the keyboard. The **runacct** command will run in the background (&), ignoring all INTERRUPT and QUIT signals (the **nohup** command), and write all standard error output (2>) to the **/var/adm/acct/nite/accterr** file.

3. To restart the system accounting procedures for a specific date, enter a command similar to the following:

```
nohup /usr/sbin/acct/runacct 0601 2>> \
/var/adm/acct/nite/accterr &
```

This example restarts **runacct** for the day of June 1 (0601). The **runacct** command reads the file **/var/adm/acct/nite(x)/statefile** to find out the state with which to begin. The **runacct** command will run in the background (&), ignoring all INTERRUPT and QUIT signals (**nohup**). Standard error output (2) is added to the end (>>) of the **/var/adm/acct/nite(x)/accterr** file.

4. To restart the system accounting procedures for a particular date at a specific state, enter a command similar to the following:

```
nohup /usr/sbin/acct/runacct 0601 MERGE 2>> \
/var/adm/acct/nite(x)/accterr &
```

This example restarts the **runacct** command for the day of June 1 (0601), starting with the MERGE state. The **runacct** command will run in the background (&), ignoring all INTERRUPT and QUIT signals (the **nohup** command). Standard error output (2) is added to the end (>>) of the **/var/adm/acct/nite(x)/accterr** file.

Files

Item	Description
/var/adm/wtmp	Log in/log off history file.
/var/adm/pacct*	Process accounting file.
/var/adm/acct/nite(x)/daytacct	Disk usage accounting file.
/var/adm/qacct	Active queue accounting file.
/var/adm/fee	Record of fees charged to users.
/var/adm/acct/sum(x)/*	Command and total accounting summary files.
/var/adm/acct/nite(x)/ptacct*.mmd	Concatenated version of pacct files.
/var/adm/acct/nite(x)/active	The runacct message file.
/var/adm/acct/nite(x)/lock*	Prevents simultaneous invocation of runacct .
/var/adm/acct/nite(x)/lastdate	Contains last date runacct was run.
/var/adm/acct/nite(x)/statefile	Contains current state to process.

runact Command

Purpose

Runs an action on a resource class.

Syntax

```
runact -s "selection_string" [-N { node_file | "-" }] [-f resource_data_input_file] [-l | -t | -d | -D delimiter] [-x] [-h] [-TV] resource_class action [in_element=value...] [rsp_element...]
```

```
runact -r [-f resource_data_input_file] [-l | -t | -d | -D delimiter] [-x] [-h] [-TV] resource_handle action [in_element=value...] [rsp_element...]
```

```
runact -c [-f resource_data_input_file] [-n node_name] [-l | -t | -d | -D delimiter] [-x] [-h] [-TV] resource_class action [in_element=value...] [rsp_element...]
```

```
runact -C domain_name... [-f resource_data_input_file] [-l | -t | -d | -D delimiter] [-x] [-h] [-TV]
resource_class action [in_element=value...] [rsp_element...]
```

Description

The `runact` command requests that the RMC subsystem run the specified action on the specified resource class.

Instead of specifying multiple node names in *selection_string*, you can use the `-N node_file` flag to indicate that the node names are in a file. Use `-N "-"` to read the node names from standard input.

Before you run this command, use the `lsactdef` command to list the resource class actions that are supported by this resource class. Also, use the `lsactdef` command to list the required input action elements that must be specified when invoking an action. The `lsactdef` command also identifies the data type for each input element. The value specified for each input element must match this data type.

Flags

-c

Invokes the action on the resource class.

To invoke the class action on a globalized resource class on all peer domains defined on the management server, set **CT_MANAGEMENT_SCOPE=3** and use the **-c** flag.

-C domain_name...

Invokes a class action on a globalized resource class on one or more RSCT peer domains that are defined on the management server. Globalized classes are used in peer domains and management domains for resource classes that contain information about the domain.

-f resource_data_input_file

Specifies the name of the file that contains resource action input elements and values. Use the `lsactdef` command with the `-i` flag to generate a template for this input file.

-d

Specifies delimiter-formatted output. The default delimiter is a colon (:). Use the `-D` flag if you want to change the default delimiter.

-D delimiter

Specifies delimiter-formatted output that uses the specified delimiter. Use this flag to specify a delimiter other than the default colon (:). An example is when the data to be displayed contains colons. Use this flag to specify a delimiter of one or more characters.

-l

Specifies "long" format — one entry per line. This is the default display format.

-n node_name

Specifies the name of the node on which to run the class action. You can only use this flag in conjunction with the **-c** flag.

-N { node_file | "-" }

Specifies that node names are read from a file or from standard input. Use `-N node_file` to indicate that the node names are in a file.

- There is one node name per line in *node_file*
- A number sign (#) in column 1 indicates that the line is a comment
- Any blank characters before a node name are ignored
- Any characters after a node name are ignored

Use `-N "-"` to read the node names from standard input.

The `CT_MANAGEMENT_SCOPE` environment variable determines the scope of the cluster. If `CT_MANAGEMENT_SCOPE` is not set, management domain scope is chosen first (if a management domain exists), peer domain scope is chosen next (if a peer domain exists), and then local scope is chosen, until the scope is valid for the command. The command runs once for the first

valid scope it finds. For example, if a management domain and a peer domain both exist and CT_MANAGEMENT_SCOPE is not set, this command applies to the management domain. If you want this command to apply to the peer domain, set CT_MANAGEMENT_SCOPE to 2.

-r "resource_handle"

Specifies a resource handle. The resource handle must be specified in this format:

```
"0xn0000 0xn0000 0xn00000000 0xn00000000 0xn00000000 0xn00000000"
```

where n is a hexadecimal character. Use this flag to invoke the action on the resource that matches *resource_handle*.

-s "selection_string"

Specifies a selection string. All selection strings must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
-s 'Name ?= "test"'
```

Only persistent attributes can be listed in a selection string. For information on how to specify selection strings, see the *Administering RSCT*.

-t

Specifies table format. Each attribute is displayed in a separate column, with one resource per line.

-x

Suppresses header printing.

-h

Writes the command's usage statement to standard output.

-T

Writes the command's trace messages to standard error. For your software-service organization's use only.

-V

Writes the command's verbose messages to standard output.

Parameters

action

Specifies the name of the action to be invoked.

in_element=value...

Specifies the action input element names and values. If you use the -f flag, don't enter any *in_element=value* pairs on the command line.

in_element is any of the input structured data element names. There should be one *in_element_n=value* pair for each of the defined structured data (SD) input elements for the specified action. Use `lsactdef` with the -s i flag to list the input elements for a particular resource class and action. Use `lsactdef -i` to generate an input file template, which, after appropriate editing, can be used as the input file.

value must be the appropriate datatype for the specified element. For example, if `NodeNumber` is defined as a `uint32` datatype, enter a positive numeric value.

resource_class

Specifies the name of the resource class with the actions that you want to invoke.

resource_handle

Specifies the resource handle for the resource and class with the actions that you want to invoke.

rsp_element

Specifies one or more of action response structured data element names. If you specify one or more element names, only those elements are displayed in the order specified. If you do not specify any element names, all elements of the response are displayed.

Security

This command requires `root` authority.

Exit Status

- 0**
The command has run successfully.
- 1**
An error occurred with RMC.
- 2**
An error occurred with the command-line interface (CLI) script.
- 3**
An incorrect flag was specified on the command line.
- 4**
An incorrect parameter was specified on the command line.
- 5**
An error occurred with RMC that was based on incorrect command-line input.

Environment Variables

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. CT_IP_AUTHENT only has meaning if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon to monitor and control the resources and resource classes. The management scope determines the set of possible target nodes where the resources and resource classes can be monitored and controlled. The valid values are:

- 0**
Specifies *local* scope.
 - 1**
Specifies *local* scope.
 - 2**
Specifies *peer domain* scope.
 - 3**
Specifies *management domain* scope.
- If this environment variable is *not* set, *local* scope is used.

Implementation Specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) fileset for AIX.

Standard Output

When the `-h` flag is specified, this command's usage statement is written to standard output.

The command output and all verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

1. To invoke the **TestClassAction** resource class action on the resource class **IBM**. Example, enter:

```
runact -c IBM.Example TestClassAction Int32=99
```

The output will look like this:

```
Resource Class Action Response for: TestClassAction
sd_element 1:
  Int32 = 99
```

Location

/opt/rsct/bin/runact

Contains the **runact** command

runcat Command

Purpose

Pipes output data from the **mkcatdefs** command to the **gencat** command.

Syntax

runcat *CatalogName SourceFile* [*CatalogFile*]

Description

The **runcat** command invokes the **mkcatdefs** command and pipes the message catalog source data (the output from **mkcatdefs**) to the **gencat** program.

The file specified by the *SourceFile* parameter contains the message text with your symbolic identifiers. The **mkcatdefs** program uses the *CatalogName* parameter to generate the name of the symbolic definition file by adding **_msg.h** to the end of the *CatalogName* value, and to generate the symbolic name for the catalog file by adding **MF_** to the beginning of the *CatalogName* value. The definition file must be included in your application program. The symbolic name for the catalog file can be used in the library functions (such as the **catopen** subroutine).

The *CatalogFile* parameter is the name of the catalog file created by the **gencat** command. If you do not specify this parameter, the **gencat** command names the catalog file by adding **.cat** to the end of the *CatalogName* value. This file name can also be used in the **catopen** library function.

Example

To generate a catalog named **test.cat** from the message source file **test.msg**, enter:

```
runcat test test.msg
```

File

Item	Description
<code>/usr/bin/runcat</code>	Contains the runcat command.

runlpcmd Command

Purpose

Runs a least-privilege (LP) resource.

Syntax

To run an LP resource:

- On the local node:

```
runlpcmd -N resource_name | RunCmdName [-h] [-TV] ["flags_and_parms"]
```

- On all nodes in a domain:

```
runlpcmd -a -N resource_name | RunCmdName [-h] [-TV] ["flags_and_parms"]
```

- On a subset of nodes in a domain:

```
runlpcmd -n host1 [, host2, ...] -N resource_name | RunCmdName [-h] [-TV] ["flags_and_parms"]
```

Description

The `runlpcmd` command runs an LP resource, which is a root command or script to which users are granted access based on permissions in the LP access control lists (ACLs). You can use the `runlpcmd` command to call the LP command corresponding to a particular *RunCmdName* value with access permissions that match the permissions of the calling user. When `runlpcmd` is called with the `-N` flag, the LP command that is specified by the *resource_name* parameter is run. Specify all parameters and flag needed for command invocation using the *flags_and_parms* parameter. If this parameter is not specified, an empty string is passed to the LP command. This is the default.

If the `Checksum` attribute value is 0, `runlpcmd` returns an error if the `ControlFlags` value is set to check for `Checksum`; otherwise, no errors are returned. If the `ControlFlag` attribute of the LP command was set to validate the `Checksum` before the LP command was run, `runlpcmd` performs such a check. The command is run only if the calculated `Checksum` matches the value of the corresponding `Checksum` attribute. If the two do not match, the command is rejected. If, however, the `ControlFlags` attribute is set to the default value, `Checksum` validation is not performed.

You can specify the *RunCmdName* parameter along with with the `-N resource_name` flag and parameter combination. However, one restriction applies when you use the *RunCmdName* parameter. If more than one resource matches the *RunCmdName* value and the permissions of the calling user, `runlpcmd` returns an error. If one match exists for the *RunCmdName* value and the the permissions of the calling user, `runlpcmd RunCmdName` returns successfully. In order to circumvent this restriction, `runlpcmd` also lets users run LP commands by specifying their unique names, using the `-N resource_name` flag and parameter combination.

Before calling the LP command, `runlpcmd` checks to see if a `FilterScript` value exists. If so, it passes the `FilterArg` value and the *flags_and_parms* parameter string specified on the command line to `FilterScript`. If `FilterScript` returns a 0, `runlpcmd` calls the LP command. If `FilterScript` execution resulted in a non-zero value, `runlpcmd` returns an error. If `FilterScript` was empty, `runlpcmd` performs some checks, as specified in `ControlFlags`, and then calls the LP command directly.

The output of this command may include "`RC=return_code`" as the last line.

This command runs on any node. If you want this command to run on all of the nodes in a domain, use the `-a` flag. If you want this command to run on a subset of nodes in a domain, use the `-n` flag. Otherwise, this command runs on the local node.

Flags

-a

Changes one or more resources on all nodes in the domain. The `CT_MANAGEMENT_SCOPE` environment variable's setting determines the cluster scope. If `CT_MANAGEMENT_SCOPE` is not set, the LP resource manager uses scope settings in this order:

1. The management domain, if it exists
2. The peer domain, if it exists
3. Local scope

The `runlpcmd` command runs once for the first valid scope that the LP resource manager finds. For example, suppose a management domain and a peer domain exist and the `CT_MANAGEMENT_SCOPE` environment variable is not set. In this case, `runlpcmd -a` runs in the management domain. To run `runlpcmd -a` in the peer domain, you must set `CT_MANAGEMENT_SCOPE` to 2.

-n *host1[,host2,...]*

Specifies the node or nodes in the domain on which the LP resource is to be changed. By default, the LP resource is changed on the local node. The `-n` flag is valid only in a management or peer domain. If the `CT_MANAGEMENT_SCOPE` variable is not set, the LP resource manager uses scope settings in this order:

1. The management domain, if it exists
2. The peer domain, if it exists
3. Local scope

The `runlpcmd` command runs once for the first valid scope that the LP resource manager finds.

-N *resource_name*

Specifies the name of the LP resource that you want to run on one or more nodes in the domain.

-h

Writes the command's usage statement to standard output.

-T

Writes the command's trace messages to standard error.

-v

Writes the command's verbose messages to standard output.

Parameters

RunCmdName

Specifies the name of the LP resource that you want to run on one or more nodes in the domain.

"flags_and_parms"

Specifies the flags and parameters that are required input for the LP command or script. If this parameter is not specified, an empty string is passed to the LP command. This is the default.

Security

To run the `runlpcmd` command, you need:

- read permission in the Class ACL of the `IBM.LPCCommands` resource class.
- execute permission in the Resource ACL.

As an alternative, the Resource ACL can direct the use of the Resource Shared ACL if this permission exists in the Resource Shared ACL.

Permissions are specified in the LP ACLs on the contacted system. See the `lpac1` file for general information about LP ACLs and the *RSCT Administration Guide* for information about modifying them.

Exit Status

- 0**
The command has run successfully.
- 1**
An error occurred with RMC.
- 2**
An error occurred with the command-line interface (CLI) script.
- 3**
An incorrect flag was specified on the command line.
- 4**
An incorrect parameter was specified on the command line.
- 5**
An error occurred with RMC that was based on incorrect command-line input.
- 6**
The resource was not found.

Environment Variables

CT_CONTACT

Determines the system that is used for the session with the RMC daemon. When `CT_CONTACT` is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the LP resources that are processed.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon to process the LP resources. The management scope determines the set of possible target nodes where the resources can be processed. The valid values are:

- 0**
Specifies *local* scope.
- 1**
Specifies *local* scope.
- 2**
Specifies *peer domain* scope.
- 3**
Specifies *management domain* scope.

If this environment variable is not set, *local* scope is used.

Implementation Specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) fileset for AIX.

Standard Output

When the `-h` flag is specified, this command's usage statement is written to standard output. When the `-V` flag is specified, this command's verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

To run the LP resource called LP1, which has required input flags and parameters -a -p User Group, enter:

```
runlpcmd LP1 "-a -p User Group"
```

Location

/opt/rsct/bin/runlpcmd

Contains the runlpcmd command

rup Command

Purpose

Shows the status of a remote host on the local network.

Syntax

```
/usr/bin/rup [ -h | -l | -t ] [ Host ... ]
```

Description

The **rup** command displays the status of a remote host by broadcasting on the local network and then displaying the responses it receives. Specify a flag if you want to sort the output. If you do not specify a flag, the **rup** command displays responses in the order they are received. If you specify multiple hosts on the command line, the **rup** command ignores any flags and displays output in the order you specified the hosts. You must use the **inetd** daemon.

Note:

1. Broadcasting does not work through gateways. Therefore, if you do not specify a host, only hosts on your network can respond to the **rup** command.
2. Load-average statistics are not kept by the kernel. The load averages are always reported as 0 (zero) by this command.

Flags

Item	Description

-h	Sorts the display alphabetically by host name.
-l	Sorts the display by load average.
-t	Sorts the display by length of runtime on the network.

Examples

1. To find out the status of all hosts on the network and to sort the list alphabetically by host name, enter:

```
/usr/bin/rup -h
```

2. To display a list of all hosts on the network according to each machine's load average, enter:

```
/usr/bin/rup -l
```

3. To display the status of a host, enter:

```
/usr/bin/rup brutus
```

In this example, the **rup** command displays the status of the host named brutus.

4. To display the status of all hosts on the network sorted by each machine's length of runtime, enter:

```
/usr/bin/rup -t
```

Files

Item	Description
html	

runtime Command

Purpose

Shows the status of each host on a network.

Syntax

```
runtime [ -a ] [ -r ] [ -l | -t | -u ]
```

Description

The **/usr/bin/runtime** command displays the status of each host that is on a local network and is running the **rwhod** daemon. The status lines are sorted by host name unless the **-l**, **-t**, or **-u** flag is indicated. The status information is provided in packets broadcast once every 3 minutes by each network host running the **rwhod** daemon. Any activity (such as power to a host being turned on or off) that takes place between broadcasts is not reflected until the next broadcast. Hosts for which no status information is received for 11 minutes are reported as down.

Output is in the following format: hostname, status, time, number of users, and load average. Load average represents the load averages over 1-, 5-, and 15-minute intervals prior to a server's transmission. The load averages are multiplied by 10 to represent the value in decimal format.

Flags

Item	Description

-a	Includes all users. Without this flag, users whose sessions are idle an hour or more are not included.
-l	Sorts the list by the load average.
-r	Reverses the sort order. The -r flag should be used with the -l , -t or -u flag.
-t	Sorts the list by the uptime.
-u	Sorts the list by the number of users.

Examples

1. To get a status report on the hosts on the local network, enter:

```
runtime
```

Information similar to the following is displayed:

```
host1    up      5:15,   4 users,   load 0.09, 0.04, 0.04
host2    up      7:45,   3 users,   load 0.08, 0.07, 0.04
host7    up      7:43,   1 user,    load 0.06, 0.12, 0.11
```

2. To get a status report sorted by load average, enter:

```
uptime -l
```

Information similar to the following is displayed:

```
host2    up      7:45,   3 users,   load 0.08, 0.07, 0.04
host1    up      5:18,   4 users,   load 0.07, 0.07, 0.04
host7    up      7:43,   1 user,    load 0.06, 0.12, 0.11
```

Files

Item	Description
<code>/var/spool/rwho/whod.*</code>	Indicates data files received from remote rwhod daemons.

ruser Command

Purpose

Directly manipulates entries in three separate system databases that control foreign host access to programs.

Syntax

To Add or Delete a Database File Name Entry

```
ruser { -a | -d } { -f "UserName ..." | -p "HostName ..." | -r "HostName ..."
```

To Delete or Display All Name Entries in a Database File

```
ruser { -X | -s } { -F | -P | -R } [ -Z ]
```

Description

The **ruser** low-level command adds or deletes entries in three separate system databases. Which database you are manipulating is determined by using the **-p**, **-r**, or **-f** flags. In addition, the **ruser** command can show one or all entries in one of the databases. Each database is a list of names. The three databases are as follows:

- `/etc/ftpusers` file
- `/etc/hosts.equiv` file
- `/etc/hosts.lpd` file

Note: The **-p** and **-r** options can be used together to add a name to databases at the same time, but the **-f** option cannot be used with either.

You could also use the System Management Interface Tool (SMIT) **smit users** fast path to run this command or type:

```
smit rprint
```

Flags

Item	Description
-a	Adds a name to the database. The -a flag must be used with either the -p , -r , or -f flag.
-d	Deletes a name from the database. Must be used with either the -p , -r , or -f flag.
-F	Deletes or shows all entries in the /etc/ftpusers file. Use this flag with the -X flag to delete all entries. Use this flag with the -s flag to show all entries.
-f "UserName ..."	Adds or deletes the user name specified by the <i>UserName</i> variable to the /etc/ftpusers database that contains a list of local user names that cannot be used by remote FTP clients. The -f flag must be used with either the -a or -d flag.
-P	Deletes or shows all entries in the /etc/hosts.lpd file. Use this flag with the -X flag to delete all entries. Use this flag with the -s flag to show all entries.
-p "HostName ..."	Adds or deletes the host name, specified by the <i>HostName</i> variable, in the database that specifies which foreign host may print on your machine. The -p flag must be used with either the -a or -d flag.
-R	Deletes or shows all entries in the /etc/hosts.equiv file. Use this flag with the -X flag to delete all entries. Use this flag with the -s flag to show all entries.
-r "HostName ..."	Adds or deletes the host name, specified by the <i>HostName</i> variable, in the /etc/hosts.equiv database that specifies which foreign host may perform the remote commands (rlogin , rcp , rsh , or print) on your machine. The -r flag must be used with either the -a or -d flag.
-s	Shows all entries in the database. Use this flag with either the -P , -R , or -F flag.
-X	Deletes all names from the database. Use this flag with either the -P , -R , or -F flag.
-Z	The -s flag is required when the -Z flag is specified. If the -Z flag is specified, a brief title is displayed before the database display.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To add an entry in the **/etc/hosts.lpd** database, which specifies which foreign host may print on the local machine, type the command in the following format:

```
ruser -a -p "host1"
```

In this example, the foreign host is host1.

2. To delete an entry in the database that controls printing only (**/etc/hosts.lpd**), and also delete the same name from the database that controls remote access for the **rlogin**, **rcp**, and **rsh** commands (**/etc/hosts.equiv**), type:

```
ruser -d -r "host2" -p "host1"
```

In this example, the host from which the database entry is deleted is host1.

rusers Command

Purpose

Reports a list of users logged on to remote machines.

Syntax

```
/usr/bin/rusers [ -a ] [ -l ] [ -u | -h | -i ] [ Host ...]
```

Description

The **rusers** command produces a list of users who are logged on to remote machines. The **rusers** command does this by broadcasting to each machine on the local network and printing the responses it receives. Normally, the system prints the responses in the order they are received. To change this order, specify one of the flags. In addition, when you provide a *Host* parameter, the **rusers** command queries the host or hosts you specify, rather than broadcasting to all hosts.

By default, each entry contains a list of users for each machine. Each of these entries includes the names of all users logged in that machine. In addition, when the user does not type into the system for a minute or more, the **rusers** command reports the user's idle time.

A remote host responds only if it is running the **rusersd** daemon, which is normally started from the **inetd** daemon.

Note: Broadcasting does not work through gateways. Therefore, if you do not specify a host, only hosts on your network can respond to the **rusers** command.

Flags

It	Description

-a	Gives a report for a machine even if no users are logged in.
-h	Sorts alphabetically by host name.
-i	Sorts by idle time.
-l	Gives a longer listing similar to the who command.
-u	Sorts by number of users.

Examples

1. To produce a list of the users on your network that are logged in remote machines, enter:

```
rusers
```

2. To produce a list of users sorted alphabetically by host name, enter:

```
rusers -h
```

3. To produce a list of users on a host, enter:

```
rusers -h pluto
```

In this example, the **rusers** command produces a list of users on the host named *pluto*.

4. To produce a list of users logged in remote machines and sorted according to each machine's length of idle time, enter:

```
rusers -i
```

5. To produce a list of users logged in remote machines and sorted by the number of users logged in, enter:

```
rusers -u
```

Files

Item	Description
<code>/etc/inetd.conf</code>	TCP/IP configuration file that starts RPC daemons and other TCP/IP daemons.

rusersd Daemon

Purpose

Responds to queries from the **rusers** command.

Syntax

```
/usr/lib/netsvc/rusers/rpc.rusersd
```

Description

The **rusersd** daemon is a server that responds to queries from the **rusers** command by returning a list of users currently on the network. This daemon is normally started by the **inetd** daemon.

Files

Item	Description
<code>/etc/inetd.conf</code>	TCP/IP configuration file that starts RPC daemons and other TCP/IP daemons.
<code>/etc/inetd.conf</code>	Contains information on users logged in to the system.

rvsdrestrict Command

Purpose

rvsdrestrict – Displays and sets the run level of the Recoverable virtual shared disk subsystem. This command must be issued before the RVSD subsystem will start.

Syntax

```
rvsdrestrict  
{-l | -s {RVSD4.1 | RESET}}
```

Description

The **rvsdrestrict** command is used to restrict the level at which the Recoverable virtual shared disk subsystem will run. If a node has a low level of the RVSD software installed than what is set with this command, then the RVSD subsystem will not start on that node.

This command does not dynamically change RVSD subsystem run levels across the peer domain. An RVSD subsystem instance will only react to this information after being restarted. If your peer domain runs at a given level, and you want to override this level, you must:

1. Stop the RVSD subsystem on all nodes.
2. Override the level.

3. Restart the RVSD subsystem.

Flags

- l**
Lists the current RVSD subsystem run level.
- s**
Sets the RVSD subsystem run level.

Parameters

None.

Security

You must have root authority to run this command.

Exit Status

- 0**
Indicates the successful completion of the command.
- nonzero**
Indicates that an error occurred.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **starttrpdomain** command. To bring a particular node online in an existing peer domain, use the **starttrpnode** command. For more information on creating and administering an RSCT peer domain, refer to the *RSCT: Administration Guide*.

Standard Output

Current RVSD subsystem run level.

Examples

1. To set the RVSD subsystem run level to RVSD4.1, you would issue the command:

```
rvsdrestrict -s RVSD4.1
```

Location

/opt/rsct/vsd/bin/rvsdrestrict

rwall Command

Purpose

Sends messages to all users on the network.

Syntax

To Send a Message to Specified Hosts

/usr/sbin/rwall *HostName* ...

To Send a Message to Specified Networks

`/usr/sbin/rwall -n NetworkGroup ...`

To Send a Message to Specified Hosts on a Network

`/usr/sbin/rwall -h HostName ... -n NetworkGroup`

Description

The **rwall** command sends messages to all users on the network. To do this, the **rwall** command reads a message from standard input until it reaches an end-of-file character. The **rwall** command takes this message, which begins with the line `Broadcast Message . . .`, and broadcasts it to all users logged in to the specified host machines. Users receive messages only if they are running the **rwalld** daemon, which is started by the **inetd** daemon.

Note: The time out is fairly short. This enables the **rwall** command to send messages to a large group of machines (some of which may be down) in a reasonable amount of time. Thus the message may not get through to a heavily loaded machine.

Flags

Item	Description

-h	Sends the message to machines specified by the <i>HostName</i> parameter.
-n	Sends the message to specific network groups only. Network groups are defined in the netgroup file.

Examples

1. To send a message to a host named neptune, enter:

```
/usr/sbin/rwall neptune
```

Type in your message. When you are done, enter:

```
Ctrl D
```

2. To send a message to a host named neptune and every host in the cosmos netgroup, enter:

```
rwall -n cosmos -h neptune
```

Type in your message. When you are done, enter:

```
Ctrl D
```

Files

Item	Description
<code>/etc/inetd.conf</code>	TCP/IP configuration file that starts RPC daemons and other TCP/IP daemons.
<code>/etc/netgroup</code>	Contains information about each user group on the network.

rwalld Daemon

Purpose

Handles requests from the **rwall** command.

Syntax

/usr/lib/netsvc/rwall/rpc.rwalld

Description

The **rwalld** daemon handles requests from the **rwall** command. The **inetd** daemon invokes the **rwalld** daemon.

Files

Item	Description
/etc/inetd.conf	Specifies the TCP/IP configuration.

rwho Command

Purpose

Shows which users are logged in to hosts on the local network.

Syntax

rwho [**-a**]

Description

The **/usr/bin/rwho** command displays the user name, host name, and start date and time of each session for everyone on the local network who is currently logged in to a host running the **rwhod** daemon. If a workstation is inactive for at least 3 minutes, the **rwho** command reports the idle time as a number of minutes in the last column. After an hour of inactivity, a user is not included unless the **-a** flag is specified.

Note: Since this command displays a lot of output, use this command with caution if the local network has a large number of users.

Status information is broadcast once every 3 minutes by each network host running the **rwhod** daemon. Any activity (such as a user logging on or off) that takes place between broadcasts is not reflected until the next broadcast.

Flags

Item	Description

-a	Includes all users. Without this flag, users whose sessions are idle an hour or more are not included in the report.

Example

To get a report of all users currently logged in to hosts on the local network, enter:

```
rwho
```

Information similar to the following is displayed:

```
bob      host2:pts5      Nov 17 06:30 :20
bob      host7:console Nov 17 06:25 :25
fran     host1:pts0      Nov 17 11:20 :51
fran     host1:pts8      Nov 16 15:33 :42
fran     host4:console Nov 17 16:32
```

```
server host2:console Nov 17 06:58 :20
alice host2:pts6 Nov 17 09:22
```

Files

Item	Description
/var/spool/rwho/whod.*	Indicates data files received from remote rwhod daemons.

rwhod Daemon

Purpose

Provides the server function for the **rwho** and **ruptime** commands.

Syntax

Note: Use SRC commands to control the **rwhod** daemon from the command line. Use the **rc.tcpip** file to start the daemon with each system startup.

/usr/sbin/rwhod

Description

The **/usr/sbin/rwhod** daemon maintains the database used by the **rwho** and **ruptime** commands. Once started, the **rwhod** daemon operates as both producer and consumer of status information.

As a producer of status information, the **rwhod** daemon queries the state of the local host approximately every 3 minutes. It then constructs status messages and broadcasts them to the local network.

As a consumer of status information, the **rwhod** daemon listens for status messages from **rwhod** servers on remote hosts. When the **rwhod** daemon receives a status message, it validates the received status message. It then records the message in the **/var/spool/rwho** directory. (The **rwho** and **ruptime** commands use the files in the **/var/spool/rwho** directory to generate their status listings.)

The **rwhod** daemon broadcasts and receives status messages using the **rwho** socket as specified in the **/etc/services** file.

When creating these messages, the **rwhod** daemon calculates the entries for the average CPU load for the previous 1-, 5-, and 15-minute intervals. Before broadcasting these messages, the **rwhod** daemon converts them to the byte order that the network can use.

When the **rwhod** daemon receives messages on the **rwho** socket, it discards any that do not originate from an **rwho** socket. Additionally, it discards any messages that contain unprintable ASCII characters. When the **rwhod** daemon receives a valid message, it places the message in a **whod.HostName** file in the **/var/spool/rwho** directory, overwriting any file with the same name.

The **rwhod** daemon should be controlled using the System Resource Controller (SRC). Entering **rwhod** at the command line is not recommended.

Manipulating the rwhod Daemon with the System Resource Controller

The **rwhod** daemon is a subsystem controlled by the System Resource Controller (SRC). The **rwhod** daemon is a member of the **tcpip** system group. This daemon is disabled by default and can be manipulated by the following SRC commands:

Item	Description
stopsrc	Stops a subsystem, group of subsystems, or a subserver.
traceson	Enables tracing of a subsystem, group of subsystems, or a subserver.

Item	Description
tracesoff	Disables tracing of a subsystem, group of subsystems, or a subserver.
tracesoff	Gets the status of a subsystem, group of subsystems, or a subserver.

Examples

1. To start the **rwhod** daemon, enter the following:

```
startsrc -s rwhod
```

This command starts the daemon. You can use this command in the **rc.tcpip** file or on the command line. The **-s** flag specifies that the subsystem that follows is to be started.

2. To stop the **rwhod** daemon normally, enter the following:

```
stopsrc -s rwhod
```

This command stops the daemon. The **-s** flag specifies that the subsystem that follows is to be stopped.

3. To get a short status report from the **rwhod** daemon, enter the following:

```
lssrc -s rwhod
```

This command returns the name of the daemon, the process ID of the daemon, and the state of the daemon (active or inactive).

4. To enable tracing for **rwhod** daemon, enter the following:

```
traceson -s rwhod
```

This command enables socket level debugging. Use the **trpt** command to look at the output of this example command.

Files

Item	Description
/etc/utmp	Contains status information on users that are logged in to the local host.
/var/spool/rwho/*	Contains files used by the rwho and ruptime commands to generate their status list.
/var/spool/rwho/whod.HostName	Contains the latest status information for the host specified by the <i>HostName</i> parameter.

S

The following AIX commands begin with the letter s.

sa Command

Purpose

Summarizes accounting records.

Syntax

```
/usr/sbin/sa [ -a ] [ -b ] [ -c ] [ -C ] [ -d ] [ -D ] [ -i ] [ -j ] [ -k ] [ -K ] [ -l ] [ -m ] [ -n ] [ -r ] [ -s ] [ -t ]  
[ -u ] [ -vNumber ] [ -f ] [ -SSaveFile ] [ -UUserFile ] [ File ... ]
```

Description

The **sa** command summarizes the information in the file that collects the raw accounting data, either the **/var/adm/pacct** file or the file specified by the *File* parameter, and writes a usage summary report to the **/var/adm/savacct** file. Then the **sa** command deletes the data in the **/var/adm/pacct** file so it can collect new accounting information. The next time the **sa** command executes, it reads the usage summary and the new data and incorporates all the information in its report.

The flags used with the **sa** command vary the type of information that is reported. The reports can contain the following fields:

Item	Description
avio	Indicates the average number of I/O operations per execution.
cpu	Indicates the sum of user and system time (in minutes).
k	Indicates the average K-blocks of CPU-time per execution.
k*sec	Indicates the CPU storage integral in kilo-core seconds.
re	Indicates the minutes of real time.
s	Indicates the minutes of system CPU time.
tio	Indicates the total number of I/O operations.
u	Indicates the minutes of user CPU time.

If you run the **sa** command without specifying any flags, the summary report includes the number of times each command was called as well as the **re**, **cpu**, **avio**, and **k** fields.

Note: The **-b**, **-d**, **-D**, **-k**, **-K**, and **-n** flags determine how output is sorted. If you specify more than one of these flags on the command line, only the last one specified will take effect.

Summary files created under this release of the base operating system are saved in a format that supports large user IDs (8 characters or longer). Summary files created under previous releases may be in the old format that supports only user IDs of up to 7 characters. The **sa** command recognizes and supports both formats of the summary file. If you need to convert old format summary files to the new format, use the **-C** flag instead of the **-s** flag. You need to do this conversion only once. After converting you can use either the **-s** or the **-C** flag.

Flags

Item	Description
-a	Prints all command names, including those with unprintable characters. Commands that were used once are placed under the other category.
-b	Sorts output by the sum of user and system time divided by the number of calls. Otherwise, output is the sum of user and system time.
-c	Prints the time used by each command as a percentage of the time used by all the commands. This is in addition to the user, system and real time.
-C	Merges the accounting file into the summary file. If the summary file is in the old format, it is converted into the new format.
-d	Sorts the output by the average number of disk I/O operations.
-D	Sorts and prints the output by the total number of disk I/O operations.
-f	Does not force interactive threshold compression. This flag must be used with the -v flag.
-i	Reads only the raw data, not the summary file.
-j	Prints the number of seconds per call instead of the total minutes per category.
-k	Sorts the output by the average CPU time.
-K	Sorts and prints the output by the CPU-storage integral.
-l	Separates system and user time, instead of combining them.
-m	Prints the number of processes and the number of CPU minutes for each user.
-n	Sorts output by the number of calls.
-r	Reverses the order of the sort.
-s	Merges the accounting file into the summary file.
-S SaveFile	Uses the specified saved file as the command summary file, instead of the /var/adm/savacct file.
-t	Prints the ratio of real time to the sum of user and system time for each command.
-u	Suspends all other flags and prints the user's numeric ID and the command name for each command.
-U UserFile	Uses the specified file instead of the /var/adm/usracct file to accumulate the per-user statistics printed by the -m flag.
-v Number	Types the name of each command used the specified number times or fewer. When queried, if you type y (yes), the command is added to the junk category and appears in future summaries as part of that category.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To summarize accounting records for all the commands in the **/var/adm/pacct** file, enter:

```
sa -a
```


Commands used only once are placed under the other field.

2. To summarize accounting records by average CPU time, enter:

```
sa -k
```

Files

Item	Description
<code>/usr/sbin/sa</code>	Contains the sa command.
<code>/etc/sa</code>	Contains the symbolic link to the sa command.
<code>/var/adm/pacct</code>	Contains raw accounting records.
<code>/var/adm/savacct</code>	Contains summary accounting records.
<code>/var/adm/usracct</code>	Contains summary accounting records by user.

sa1 Command

Purpose

Collects and stores binary data in the `/var/adm/sa/sadd` file.

Syntax

```
/usr/lib/sa/sa1 [ Interval Number ]
```

Description

The **sa1** command is a shell procedure variant of the **sadc** command and handles all of the flags and parameters of that command. The **sa1** command collects and stores binary data in the `/var/adm/sa/sadd` file, where *dd* is the day of the month. The *Interval* and *Number* parameters specify that the record should be written *Number* times at *Interval* seconds. If you do not specify these parameters, a single record is written. You must have permission to write in the `/var/adm/sa` directory to use this command.

The **sa1** command is designed to be started automatically by the **cron** command. If the **sa1** command is not run daily from the **cron** command, the **sar** command displays a message about the nonexistence of the `/usr/lib/sa/sa1` data file.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

To create a daily record of **sar** activities, place the following entry in your adm **crontab** file:

```
0 8-17 * * 1-5 /usr/lib/sa/sa1 1200 3 &
```

Files

Item	Description
<code>/var/adm/sa</code>	Specifies the directory containing the daily data files.

Item	Description
<code>/var/adm/sa/sadd</code>	Contains the daily data file, where the <i>dd</i> parameter is a number representing the day of the month.
<code>/usr/lib/sa/sa1</code>	Contains the sa1 command.

sa2 Command

Purpose

Writes a daily report in the `/var/adm/sa/sardd` file.

Syntax

`/usr/lib/sa/sa2`

Description

The **sa2** command is a variant shell procedure of the **sar** command, which writes a daily report in the `/var/adm/sa/sardd` file, where *dd* is the day of the month. The **sa2** command handles all of the flags and parameters of the **sar** command.

The **sa2** command is designed to be run automatically by the **cron** command and run concurrently with the **sa1** command.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

To run the **sa2** command daily, place the following entry in the root **crontab** file:

```
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 3600 -ubcwyavm &
```

This will generate a daily report called `/var/adm/sa/sardd`. It will also remove a report more than one week old.

Files

Item	Description
<code>/var/adm/sa</code>	Specifies the directory containing the daily data files.
<code>/var/adm/sa/sar<i>dd</i></code>	Contains daily data file, where the <i>dd</i> parameter is a number representing the day of the month.
<code>/usr/lib/sa/sa2</code>	The path to the shell script of the sa2 command.

sact Command

Purpose

Displays current SCCS file-editing status.

Syntax

sact *File* ...

Description

The **sact** command reads Source Code Control System (SCCS) files and writes to standard output the contents, if any, of the p-file associated with the specified value of the *File* variable. The p-file is created by the **get -e** command. If a - (minus sign) is specified for the *File* value, the **sact** command reads standard input and interprets each line as the name of an SCCS file. If the *File* value is a directory, the **sact** command performs its actions on all SCCS files.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Examples

To display the contents of a p-file, enter:

```
sact File
```

Files

Item	Description
/usr/bin/sact	Contains the path to the SCCS sact command.

sadc Command

Purpose

Provides a system data collector report.

Syntax

/usr/lib/sa/sadc [*Interval Number*] [*Outfile*]

/usr/lib/sa/sa1 [*Interval Number*]

/usr/lib/sa/sa2

Description

The **sadc** command, the data collector, samples system data a specified number of times (*Number*) at a specified interval measured in seconds (*Interval*). It writes in binary format to the specified outfile or to the standard output. When both *Interval* and *Number* are not specified, a dummy record, which is used at system startup to mark the time when the counter restarts from 0, will be written. The **sadc** command is intended to be used as a backend to the **sar** command.

The operating system contains a number of counters that are incremented as various system actions occur. The various system actions include:

- System Configuration Parameters
- System unit utilization counters
- Buffer usage counters
- Disk and tape I/O activity counters
- Tty device activity counters
- Switching and subroutine counters
- File access counters
- Queue activity counters
- Interprocess communication counters

Note: The **sadc** command reports only local activity.

Security

Access Control: These commands should grant execute (x) access only to members of the **adm** group.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

To write 10 records of one second intervals to the **/tmp/rpt** binary file, enter:

```
sadc 1 10 /tmp/rpt
```

Files

Item	Description
/var/adm/sa/sadd	Contains the daily data file, <i>dd</i> represents the day of the month.
/var/adm/sa/sardd	Contains the daily report file, <i>dd</i> represents the day of the month.
/tmp/rpt	Contains the binary file used for input by the sar command.
/tmp/sa.adrf1	Contains the address file.

sar Command

Purpose

Collects, reports, or saves system activity information.

Syntax

```
/usr/sbin/sar [{-A [-M]] | [-a] [-b] [-c] [-d] [-k] [-m] [-q] [-r] [-u] [-v] [-w] [-y] [-M]]} [-P  
processoridentifier, ...] ALL | RST [-O {sortcolumn=col_name[,sortorder={asc|desc}],topcount=n}]] [ [-@  
wparname] [-e[YYYYMMDD]hh[:mm[:ss]]] [-ffile] [-iseconds] [-ofile] [-s[YYYYMMDD]hh[:mm  
[:ss]]] [-x] [Interval [Number]]  
sar [-X [-o filename]] [interval[count]]
```

Description

The **sar** command writes to standard output the contents of selected cumulative activity counters in the operating system. The accounting system, based on the values in the *number* and *interval* parameters, writes information the specified number of times spaced at the specified intervals in seconds. The default sampling interval for the *number* parameter is 1 second. The collected data can also be saved in the file specified by the **-o** *file* flag.

The **sar** command generates an XML file when the **-X** option is specified.

The **sar** command extracts and writes to standard output records previously saved in a file. This file can be either the one specified by the **-f** flag or, by default, the standard system activity daily data file, the **/var/adm/sa/sadd** file, where the *dd* parameter indicates the current day.

Without the **-P** flag, the **sar** command reports system-wide (global among all processors) statistics, which are calculated as averages for values expressed as percentages, and as sums otherwise. If the **-P** flag is given, the **sar** command reports activity which relates to the specified processor or processors. If **-P ALL** is given, the **sar** command reports statistics for each individual processor, followed by system-wide statistics. If **-P ALL** is used in a workload partition environment and the WPAR is associated with an **rset** registry, the resource set statistics and the system-wide statistics are displayed; the processors that belong to the resource set are prefixed with an asterisk symbol (*).

You can select information about specific system activities using flags. If you do not specify any flags, you select only system and WPAR unit activity. Specifying the **-A** flag selects all activities. The **sar** command prints the number of processors and the number of disks that are currently active before starting to print the statistics.

The default version of the **sar** command (processor utilization report) might be one of the first facilities the user runs to begin system activity investigation, because it monitors major system resources. If processor utilization is near 100 percent (user + system), the workload sampled is processor-bound. If a considerable percentage of time is spent in I/O wait, it implies that processor execution is blocked waiting for disk I/O. The I/O may be required file accesses or it may be I/O associated with paging due to a lack of sufficient memory.

Note: The time the system spends waiting for *remote* file access is *not* accumulated in the I/O wait time. If CPU utilization and I/O wait time for a task are relatively low, and the response time is not satisfactory, consider investigating how much time is being spent waiting for remote I/O. Since no high-level command provides statistics on remote I/O wait, trace data may be useful in observing this. If there is a change in system configuration that affects the output of the **sar** command, **sar** prints the average values up to the current iteration and then a warning message about the configuration change. It then continues the output, after printing the updated system configuration information.

Methods Used to Compute CPU Disk I/O Wait Time

The AIX operating system contains enhancements to the method used to compute the percentage of processor time spent waiting on disk I/O (*wio* time). The *wio* time is reported by the commands **sar** (**%wio**), **vmstat** (**wa**) and **iostat** (**%iowait**).

At each clock interrupt on each processor (100 times a second per processor), a determination is made as to which of the four categories (usr/sys/wio/idle) to place the last 10 ms of time. If the processor was busy in *usr* mode at the time of the clock interrupt, then *usr* gets the clock tick added into its category. If the processor was busy in kernel mode at the time of the clock interrupt, then the *sys* category gets the tick. If the processor was not busy, a check is made to see if any I/O to disk is in progress. If any disk I/O is in progress, the *wio* category is incremented. If no disk I/O is in progress and the processor is not busy, the *idle* category gets the tick. The inflated view of *wio* time results from all idle processors being categorized as *wio* regardless of the number of threads waiting on I/O. For example, systems with just one thread doing I/O could report over 90 percent *wio* time regardless of the number of processors it has.

The AIX operating system marks an idle processor as *wio* if an outstanding I/O was started on that processor. This method can report much lower *wio* times when just a few threads are doing I/O and the system is otherwise idle. For example, a system with four processors and one thread doing I/O will report a maximum of 25 percent *wio* time. A system with 12 processors and one thread doing I/O will report

a maximum of 8 percent wio time. NFS client reads/writes go through the VMM, and the time that biods spend in the VMM waiting for an I/O to complete is now reported as I/O wait time.

If multiple samples and multiple reports are desired, it is convenient to specify an output file for the **sar** command. Direct the standard output data from the **sar** command to `/dev/null` and run the **sar** command as a background process. The syntax for this is:

```
sar -A -o data.file interval count > /dev/null &
```

All data is captured in binary form and saved to a file (`data.file`). The data can then be selectively displayed with the **sar** command using the **-f** option.

The **sar** command calls a process named **sadc** to access system data. Two shell scripts (`/usr/lib/sa/sa1` and `/usr/lib/sa/sa2`) are structured to be run by the **cron** command and provide daily statistics and reports. Sample stanzas are included (but commented out) in the `/var/spool/cron/crontabs/adm crontab` file to specify when the **cron** daemon should run the shell scripts. Collection of data in this manner is useful to characterize system usage over a period of time and determine peak usage hours.

You can insert a dummy record into the standard system activity daily data file at the time of system start by un-commenting corresponding lines in the `/etc/rc` script. The **sar** command reports `time change not positive` for any record where processor times are less than the previous record. This occurs if you reboot the system with the dummy record insertion lines in `/etc/rc` commented out.

Beginning with AIX 5.3, the **sar** command reports utilization metrics `physc` and `%entc` which are related to Micro-Partitioning and simultaneous multithreading environments. These metrics will only be displayed on Micro-Partitioning and simultaneous multithreading environments. `physc` indicates the number of physical processors consumed by the partition (in case of system wide utilization) or logical processor (if the **-P** flag is specified) and `%entc` indicates the percentage of the allocated entitled capacity (in case of system wide utilization) or granted entitled capacity (if the **-P** flag is specified). When the partition runs in capped mode, the partition cannot get more capacity than it is allocated. In uncapped mode, the partition can get more capacity than it is actually allocated. This is called granted entitled capacity. If the **-P** flag is specified and there is unused capacity, **sar** prints the unused capacity as separate processor with `cpu id U`.

Beginning with AIX 6.1, the **sar** command reports the utilization metric `%resc`, which is related to the workload partition (WPAR) environment. The `%resc` metric indicates the percentage of processor resource that the WPAR consumes. This field is displayed only if the processor-resource limit is enforced in the WPAR. The **sar -P** command reports the resource set (RSET) utilization metrics `R` for the WPAR.

Restriction: The **sar** command only reports on local activities.

You could also use the System Management Interface Tool (SMIT) **smit sar** fast path to run this command.

Flags

Item	Description
<code>-@ wparname</code>	The <code>-@</code> flag specifies that the command reports the processor use in WPAR from the global environment. The <code>wparname</code> parameter specifies which WPAR processor statistics are to be reported. Note: The <code>-@</code> flag is not supported when executed within a workload partition. Note: Do not use the <code>-@</code> flag with the <code>-d</code> , <code>-r</code> , <code>-y</code> , <code>-f</code> , or <code>-X</code> flags.

Item	Description
-A	Without the -P flag, using the -A flag is equivalent to specifying -abcdkmqruvwxy . When used with the -P flag, the -A is equivalent to specifying -acmuw . Without the -M flag, headers are only printed once in multiple lines grouped together before the data for the first interval. When this flag is used with the -M flag, each line of data at each iteration is preceded by the appropriate header.
-a	Reports use of file access system routines specifying how many times per second several of the system file access routines have been called. When used with the -P flag, the information is provided for each specified processor; otherwise, it is provided only system-wide. The following values are displayed: dirblk/s Number of 512-byte blocks read by the directory search routine to locate a directory entry for a specific file. iget/s Calls to any of several i-node lookup routines that support multiple file system types. The iget routines return a pointer to the i-node structure of a file or device. lookupn/s Calls to the directory search routine that finds the address of a v-node given a path name.
-b	Reports buffer activity for transfers, accesses, and cache (kernel block buffer cache) hit ratios per second. Access to most files in Version 3 bypasses kernel block buffering and therefore does not generate these statistics. However, if a program opens a block device or a raw character device for I/O, traditional access mechanisms are used making the generated statistics meaningful. The following values are displayed: bread/s, bwrit/s Reports the number of block I/O operations. These I/Os are generally performed by the kernel to manage the block buffer cache area, as discussed in the description of the lread/s value. lread/s, lwrit/s Reports the number of logical I/O requests. When a logical read or write to a block device is performed, a logical transfer size of less than a full block size may be requested. The system accesses the physical device units of complete blocks and buffers these blocks in the kernel buffers that have been set aside for this purpose (the block I/O cache area). This cache area is managed by the kernel, so that multiple logical reads and writes to the block device can access previously buffered data from the cache and require no real I/O to the device. Application read and write requests to the block device are reported statistically as logical reads and writes. The block I/O performed by the kernel to the block device in management of the cache area is reported as block reads and block writes. pread/s, pwrit/s Reports the number of I/O operations on raw devices. Requested I/O to raw character devices is not buffered as it is for block devices. The I/O is performed to the device directly. %rcache, %wcache Reports caching effectiveness (cache hit percentage). This percentage is calculated as: $[(100) \times (\text{lreads} - \text{bread}) / (\text{lreads})]$.

Item	Description
-c	Reports system calls. When used with the -P flag, the information is provided for each specified processor; otherwise, it is provided only system-wide. The following values are displayed: exec/s, fork/s Reports the total number of fork and exec system calls. sread/s, swrit/s Reports the total number of read/write system calls. rchar/s, wchar/s Reports the total number of characters transferred by read/write system calls. scall/s Reports the total number of system calls. Tip: The sar command itself can generate a considerable number of reads and writes depending on the interval at which it is run. Run the sar statistics without the workload to understand the sar command's contribution to your total statistics.
-d	Reports activity for each block device with the exception of tape drives. The following data is reported: %busy Reports the portion of time the device was busy servicing a transfer request. avque Reports the average number of requests waiting to be sent to disk. read/s, write/s, blk/s Reports the read-write transfers from or to a device in kilobytes/second. await, avserv Average wait time and service time per request in milliseconds. Restriction: The -d flag is restricted in workload partitions.
-e [YYYYMMDD] hh[:mm[:ss]]	Sets the ending time of the report. The default ending time is 18:00. • If you specify the year, month, and date in the YYYYMMDD format, then the -x flag is turned on implicitly. • If you do not specify the year, month, and date in the YYYYMMDD format, then the year, month, and date are considered to be that of the first record in the activity data file that matches the specified time
-f <i>file</i>	Extracts records from the <i>file</i> (created by -o file flag). The default value of the <i>file</i> parameter is the current daily data file, the /var/adm/sa/sadd file. Restriction: If you specify the [<i>interval</i> [<i>number</i>]] parameter, the -f flag is ignored. The -f flag is restricted in workload partitions.
-i <i>seconds</i>	Selects data records at seconds as close as possible to the number specified by the <i>Seconds</i> parameter. Otherwise, the sar command reports all seconds found in the data file.

Item	Description
-k	Reports kernel process activity. The following values are displayed: kexit/s Reports the number of kernel processes terminating per second. kproc-ov/s Reports the number of times kernel processes could not be created because of enforcement of process threshold limit. ksched/s Reports the number of kernel processes assigned to tasks per second.
-M	Enables multiple headers in output when used with at least two combinations of [abckmqꞑuvwy] or with the -A flag. In this mode, each line of data is preceded by the corresponding header at each iteration. Restriction: This flag is ignored when used without [interval [number]].
-m	Reports message (sending and receiving) and semaphore (creating, using, or destroying) activities per second. When used with the -P flag, the information is provided for each specified processor; otherwise, it is provided only system-wide. The following values are displayed: msg/s Reports the number of IPC message primitives. sema/s Reports the number of IPC semaphore primitives.
-o file	Saves the readings in the file in binary form. Each reading is in a separate record and each record contains a tag identifying the time of the reading.
-P processoridentifier, ... ALL RST	Reports per-processor statistics for the specified processor or processors. Specifying the ALL keyword reports statistics for each individual processor, and globally for all processors. Specifying the RST option reports statistics for the processors present in the rset registry that is associated with the WPAR. Of the flags that specify the statistics to be reported, only the -a, -c, -m, -u, and -w flags are meaningful with the -P flag in the global environment. In the WPAR environment, do not use any flag with the -P flag. Note: The statistics for each processor that the sar command reports for WPAR are always system-wide.
-q	Reports queue statistics. The following values are displayed: runq-sz Reports the average number of kernel threads in the run queue. %runocc Reports the percentage of the time the run queue is occupied. swpq-sz Reports the average number of kernel threads that are waiting in the virtual memory manager queue for resource, input, or output. %swpocc Reports the percentage of the time the swap queue is occupied. Tip: A blank value in any column indicates that the associated queue is empty.

Item	Description
-r	Reports paging statistics. The following values are displayed: cycle/s Reports the number of page replacement cycles per second. fault/s Reports the number of page faults per second. This is not a count of page faults that generate I/O, because some page faults can be resolved without I/O. slots Reports the number of free pages on the paging spaces. odio/s Reports the number of non paging disk I/Os per second. Restriction: The -r flag is restricted in workload partitions.
-s [YYYYMMDD] hh[:mm[:ss]]	Sets the starting time of the data, causing the sar command to extract records time-tagged at, or following, the time specified. The default starting time is 08:00. • If you specify the year, month, and date in the YYYYMMDD format, then the -x flag is turned on implicitly. • If you did not specify the year, month, and date in the YYYYMMDD format, then the year, month, and date are considered to be that of the first record in the activity data file that matches the specified time.

Item	Description
-u	Reports per processor or system-wide statistics. When used with the -P flag, the information is provided for each specified processor; otherwise, it is provided only system-wide. Because the -u flag information is expressed as percentages, the system-wide information is simply the average of each individual processor's statistics. Also, the I/O wait state is defined system-wide and not per processor. The following values are displayed: %idle Reports the percentage of time the processor or processors were idle with no outstanding disk I/O requests. %sys Reports the percentage of time the processor or processors spent in execution at the system (or kernel) level. %usr Reports the percentage of time the processor or processors spent in execution at the user (or application) level. %wio Reports the percentage of time the processor(s) were idle during which the system had outstanding disk/NFS I/O request(s). See detailed description above. physc Reports the number of physical processors consumed. This data will be reported if the partition is dedicated and enabled for donation, or is running with shared processors or simultaneous multithreading enabled. %entc Reports the percentage of entitled capacity consumed. This will be reported only if the partition is running with shared processors. Because the time base over which this data is computed can vary, the entitled capacity percentage can sometimes exceed 100%. This excess is noticeable only with small sampling intervals. %resc Reports the percentage of processor resource consumed. This metric is applicable only for the WPAR environment. It is reported only if the WPAR enforces processor-resource limit. Tips: • The sar command reports system unit activity if no other specific content options are requested. If the -P flag is used and the partition is running with shared processors, and if the partition capacity usage is what is allocated, then a processor row with <code>cpu id U</code> will be reported to show the system-wide unused capacity. If the partition is running with shared processors in uncapped mode, then %entc will report the percentage of granted entitled capacity against each processor row and percentage of allocated entitled capacity in the system-wide processor row. The individual processor utilization statistics is calculated against the actual physical consumption (physc). The system wide statistics is computed against the entitlement and not physical consumption. However, in an uncapped partition, the system wide statistics is still calculated against the physical consumption. • Since the time base over which the data is computed varies, the sum of all of the %utilization fields (%user, %sys, %idle, and %wait) can exceed 100 percent.

Item	Description
-v	Reports status of the process, kernel-thread, i-node, and file tables. The following values are displayed: file-sz, inod-sz, proc-sz , thrd-sz Reports the number of entries in use for each table.
-w	Reports system switching activity. When used with the -P flag, the information is provided for each specified processor; otherwise, it is provided only system-wide. The following value is displayed: pswch/s Reports the number of context switches per second.
-y	Reports tty device activity per second. canch/s Reports tty canonical input queue characters. This field is always 0 (zero). mdmin/s Reports tty modem interrupts. outch/s Reports tty output queue characters. rawch/s Reports tty input queue characters. revin/s Reports tty receive interrupts. xmtin/s Reports tty transmit interrupts. Restriction: The -y flag is restricted in workload partitions.
-x	Displays the date and time for each entry. The -x flag is turned on implicitly whenever the user specifies the data in the YYYYMMDD format for the -s flag or the -e flag.
-OOptions	Allows users to specify the command option. -O options=value... Following are the supported options: • sortcolumn = Name of the metrics in the sar command output • sortorder = [asc desc] • topcount = Number of CPUs to be displayed in the sar command sorted output
-X	Generates the XML output. The default file name is sar_DDMMYYHHMM.xml unless the user specifies a different file name using with the -o option.
-o	Specifies the file name for the XML output.

Security

Access Control: These commands should grant execute (x) access only to members of the **adm** group.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To report system unit activity, enter the following command:

```
sar
```

2. To report current tty activity for each 2 seconds for the next 40 seconds, enter the following command:

```
sar -y -r 2 20
```

3. To watch system unit for 10 minutes and sort data, enter the following command:

```
sar -o temp 60 10
```

4. To report processor activity for the first two processors, enter the following command:

```
sar -u -P 0,1
```

This produces output similar to the following:

cpu	%usr	%sys	%wio	%idle
0	45	45	5	5
1	27	65	3	5

5. To report message, semaphore, and processor activity for all processors and system-wide, enter the following command:

```
sar -mu -P ALL
```

On a four-processor system, this produces output similar to the following (the last line indicates system-wide statistics for all processors) :

cpu	msgs/s	sema/s	%usr	%sys	%wio	%idle
0	7	2	45	45	5	5
1	5	0	27	65	3	5
2	3	0	55	40	1	4
3	4	1	48	41	4	7
-	19	3	44	48	3	5

6. To see physical processor consumed and entitlement consumed for all processors system-wide, run sar command in a shared processor logical partition machine, as follows:

```
sar -P ALL
```

On a two-logical processor system, this produces output similar to the following (the last line indicates system-wide statistics for all processors, and the line with cpuid U indicates the system-wide Unused capacity):

cpu	%usr	%sys	%wio	%idle	physc	%entc
0	0	0	0	100	0.02	3.1
1	0	0	0	100	0.00	1.0
U	-	-	0	96	0.48	96.0
-	0	0	0	100	0.02	4.0

7. To report system call, kernel process, and paging activities with separate headers for each of the three lines of data at each iteration for every 2 seconds for the next 40 seconds, enter the following command:

```
sar -Mckr 2 20
```

8. To report all activities with multiple sets of headers for every 2 seconds for the next 40 seconds, enter the following command:

```
sar -MA 2 20
```

9. To report the processor use statistics in a WPAR from the global environment, enter the following command:

```
sar -@ wparname
```

10. To report the processor activities for all of the processors present in the **rset** registry associated with the WPAR from inside a WPAR, enter the following command:

```
sar -P RST 1 1
```

In a WPAR that is associated with an RSET of two logical processors, the previous command generates a report similar to the following:

19:34:39	cpu	%usr	%sys	%wio	%idle	physc
19:34:40	0	0	2	0	98	0.54
	1	0	0	0	100	0.46
	R	0	1	0	99	1.00

11. To report all of the processor activities from inside a WPAR, enter the following command:

```
sar -P ALL 1 1
```

In a WPAR that is associated with an RSET of two logical processors, the previous command generates a report similar to the following:

19:34:39	cpu	%usr	%sys	%wio	%idle	physc
19:34:40	*0	0	2	0	98	0.54
	*1	0	0	0	100	0.46
	R	0	1	0	99	1.00
	-	0	1	0	99	1.00

12. To display the sorted output for the column **cswch/s** with the **-w** flag, enter the following command:

```
sar -w -P ALL -O sortcolumn=cswch/s 1 1
```

13. To list the top ten CPUs, sorted on the **scall/s** column, enter the following command:

```
sar -c -O sortcolumn=scall/s,sortorder=desc,topcount=10 -P ALL 1
```

Files

Item	Description
/usr/sbin/sar	Contains the sar command.
/bin/sar	Indicates the symbolic link to the sar command.
/var/adm/sa/sadd	Indicates the daily data file, where the <i>dd</i> parameter is a number representing the day of the month.

savebase Command

Purpose

Saves information about base-customized devices in the Device Configuration database onto the boot device.

Syntax

```
savebase [ -o Path ] [ -d File ] [ -v ]
```

Description

The **savebase** command stores customized information for base devices for use during phase 1 of system boot. By default, the **savebase** command retrieves this information from the **/etc/objrepos** directory.

However, you can override this action by using the **-o** flag to specify an ODM directory. The `savebase` command is typically run without any parameters. It uses the `/dev/ipl_b1v` special file link to identify the output destination.

Alternatively, use the **-d** flag to specify a destination file or a device, such as the `/dev/hdisk0` device file. To identify a specific output destination, the `-d` flag identifies the file to which `savebase` writes the base customized device data. This file can be either a regular file or a device special file. The device special file identifies either a disk device special file or a boot logical volume device special file.

A disk device special file can be used where there is only one boot logical volume on the disk. The `savebase` command ensures that the given disk has only one boot logical volume present and is bootable. If neither of these conditions is true, `savebase` does not save the base customized device data to the disk and exits with an error.

When a second boot logical volume is on a disk, the boot logical volume device special file must be used as the destination device to identify which boot image the base customized device data will be stored in. A boot logical volume device special file can be used even if there is only one boot logical volume on the disk. The `savebase` command ensures that the given device special file is a boot logical volume and it is bootable before saving any data to it. If either of these checks fails, `savebase` exits with an error.

Note: The `-m` flag is no longer used by the `savebase` command. For compatibility reasons, the flag can be specified, but `savebase` effectively ignores it.

Flags

Item	Description
-d <i>File</i>	Specifies the destination file or device to which the base information will be written.
-o <i>Path</i>	Specifies a directory containing the Device Configuration database.
-v	Causes verbose output to be written to standard output.

Examples

1. To save the base customized information and see verbose output, enter:

```
savebase -v
```

2. To specify an ODM directory other than the `/usr/lib/objrepos` directory, enter:

```
savebase -o /tmp/objrepos
```

3. To save the base customized information to the `/dev/hdisk0` device file instead of to the boot disk, enter:

```
savebase -d /dev/hdisk0
```

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see *Privileged Command Database in Security*. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Files

Item	Description
<code>/usr/include/sys/cfgdb.h</code>	Defines the type of boot mask for base devices.

Item	Description
/usr/lib/objrepos/PdDv	Contains entries for all known device types supported by the system.
/etc/objrepos/CuDv	Contains entries for all device instances defined in the system.
/etc/objrepos/CuAt	Contains customized device-specific attribute information.
/etc/objrepos/CuDep	Describes device instances that depend on other device instances.
/etc/objrepos/CuDvDr	Stores information about critical resources that need concurrency management through the use of the Device Configuration Library routines.

savecore Command

Purpose

Saves a system dump.

Syntax

savecore { [[**-c**] [**-d**] [**-f**]] [**-F** [**-d**]] } *DirectoryName* *SystemName*

Description

The function of the **savecore** command is to save a system dump and is usually run at system startup.

The **savecore** command checks to see that you have a recent dump and that there is enough space to save it. The system dump is saved in the *DirectoryName/vmcore.n* file, and the system is saved in the *DirectoryName/vmunix.n* file. The *n* variable is specified in the *DirectoryName/bounds* file. If this file does not exist, it is created with a default of **0**, and the *n* variable uses this value. With each subsequent dump, the *n* variable is increased by 1.

The compressed dump is copied to a file named *DirectoryName/vmcore. n.Z*, where **.Z** is the standard indication that a file is compressed.

If the system dump was from a system other than */unix*, the name of the system must be supplied as *SystemName*.

Note: The **savecore** command saves only the current dump and the dump prior to the current one.

The directory may contain a file named **minfree**. This file contains the number of kbytes to leave free in the directory. The **minfree** file can be used to ensure a minimum amount of free space is left after the dump is copied.

Flags

Item Description

- c** Marks the dump invalid (not recent), but does not copy it.
- d** Copies only the dump. It does not copy the system.
- f** Copies the dump even if it appears to be invalid.
- F** Reports the amount of space available for a dump in the copy directory. This may be more than the free space since the **savecore** command keeps the current dump and the previous dump, deleting others. No copying is done if the **-F** flag is specified. This flag is only valid with the **-d** flag.

Security

The Role Based Access Control (RBAC) Environment: This command implements and can perform privileged operations. Only privileged users can execute such privileged operations.

To review the list of privileges and the authorizations associated with this command, refer to the **/etc/security/privcmds** database.

Examples

1. To copy the dump (not the system) to *DirectoryName*, enter:

```
savecore -d DirectoryName
```

2. To copy the dump even if it is invalid, enter:

```
savecore -f -d DirectoryName
```

3. To mark the dump invalid, enter:

```
savecore -c
```

4. To copy the dump and the system, enter:

```
savecore -d DirectoryName SystemName
```

5. To see how much space is available for a dump, enter:

```
savecore -d -F DirectoryName
```

savevg Command

Purpose

Finds and backs up all files belonging to a specified volume group.

Syntax

```
savevg [ -a ] [ -A ] [ -b Blocks ] [ -e ] [ -f Device ] [ -i | -m ] [ -p ] [ -r ] [ -T ] [ -v ] [ -V ] [ -x file ] [ -X ]  
VGName [ -Z ]
```

Description

The **savevg** command finds and backs up all files belonging to a specified volume group. The volume group must be varied-on, and the file systems must be mounted. The **savevg** command uses the data file created by the **mkvgdata** command. This data file can be one of the following:

/image.data

Contains information about the root volume group (**rootvg**). The **savevg** command uses this file to create a backup image that can be used by Network Installation Management (NIM) to reinstall the volume group to the current system or to a new system.

/tmp/vgdata/vgname/vgname.data

Contains information about a user volume group. The *VGName* variable reflects the name of the volume group. The **savevg** command uses this file to create a backup image that can be used by the **restvg** command to remake the user volume group.

To create a backup of the operating system to CD, use the **mkcd** command.

Note: The **savevg** command will not generate a bootable tape if the volume group is the root volume group. Although the tape is not bootable, the first three images on the tape are dummy replacements for the images normally found on a bootable tape. The actual system backup is the fourth image.

Flags

Item	Description
-a	Does not back up extended attributes or NFS4 ACLs.
-A	Backs up DMAPI file system files.
-b Blocks	Specifies the number of 512-byte blocks to write in a single output operation. If this parameter is not specified, the backup command uses a default value appropriate for the physical device selected. Larger values result in larger physical transfers to tape devices. The value specified must be a multiple of the physical block size of the device being used.
-e	Excludes files specified in the /etc/exclude.vgname file from being backed up by this command. Note: If you want to exclude certain files from the backup, create the /etc/exclude.rootvg file, with an ASCII editor, and enter the patterns of file names that you do not want included in your system backup image. The patterns in this file are input to the pattern matching conventions of the grep command to determine which files will be excluded from the backup. If you want to exclude files listed in the /etc/exclude.rootvg file, select the Exclude Files field and press the Tab key once to change the default value to yes. For example, to exclude all the contents of the directory called scratch, edit the exclude file to read as follows: <pre>/scratch/</pre> For example, to exclude the contents of the directory called /tmp, and avoid excluding any other directories that have /tmp in the pathname, edit the exclude file to read as follows: <pre>^./tmp/</pre> All files are backed up relative to . (current working directory). To exclude any file or directory for which it is important to have the search match the string at the beginning of the line, use ^ (caret character) as the first character in the search string, followed by . (dot character), followed by the filename or directory to be excluded. If the filename or directory being excluded is a substring of another filename or directory, use ^. (caret character followed by dot character) to indicate that the search should begin at the beginning of the line and/or use \$ (dollar sign character) to indicate that the search should end at the end of the line.
-f Device	Specifies the device or file name on which the image is to be stored. The default is the /dev/rmt0 device.
-i	Creates the data file by calling the mkvgdata command.
-m	Creates the data file with map files by calling the mkvgdata command with the -m flag.
-p	Disables software packing of the files as they are backed up. Some tape drives use their own packing or compression algorithms.
-r	Backs up user volume group information and administration data files. This backs up files such as /tmp/vgdata/vgname/vgname.data and map files if any exist. This does not back up user data files. This backup can be used to create a user volume group without restoring user data files. This cannot be done to rootvg.

Item	Description
-T	Create a backup using snapshots. This flag applies only for JFS2 file systems. When you specify the -T flag to use snapshots for creating a volume group backup, external JFS2 snapshots are created. Snapshots allow for a point-in-time image of a JFS2 file system and thus, do not require a system to be put into a temporarily inactive state. The size of the snapshot is 2 - 15% of the size of the file system. The snapshot logical volumes are removed when backup is finished. However, snapshots are not removed if a file system already has other snapshots. Additionally, if a file system has internal snapshots, then external snapshots cannot be created and snapshots are not used for creating the backup of the file system. The use of the -T flag does not affect any JFS file systems that are present in the volume group that is being backed up. These file systems are backed up in the same manner as done previously.
-v	Verbose mode. Lists files as they are backed up.
-V	Verifies a tape backup. This flag causes savevg to verify the file header of each file on the backup tape and report any read errors as they occur.
-x file	Exclude the file systems listed in the file from the volume group backup. One file system mount point is listed per line.
-X	Specifies to automatically expand the /tmp file system if necessary. The /tmp file system may need to be extended to make room for the boot image when creating a bootable backup to tape.
-Z	Specifies that the Encrypted File System (EFS) information for all the files, directories, and file systems is not backed up. The flag runs the backup command without the -Z flag.

Parameters

Item	Description

<i>VGName</i>	Specifies the name of the volume group to be backed up.

SMIT Fast Paths

1. To list the contents of a root volume group backup that is created with the **savevg** command, enter the following SMIT fast path:

```
smit lsmksysb
```

2. To list the contents of a user volume group backup that is created with the **savevg** command, enter the following SMIT fast path:

```
smit lsbackvg
```

3. To restore individual files from a root volume group backup, enter the following SMIT fast path:

```
smit restmksysb
```

4. To restore individual files from a user volume group backup, enter the following SMIT fast path:

```
smit restsavevg
```

Examples

1. To back up the root volume group (operating system image) to the **/mysys/myvg/myroot** backup file and create an **/image.data** file, enter:

```
savevg -i -f/mysys/myvg/myroot rootvg
```

2. To back up the **uservg** volume group to the default tape drive (**/dev/rmt0**) and create a new **uservg.data** file, enter:

```
savevg -i uservg
```

3. To back up the **data2** volume group and create map files along with a new **data2.data** file on **rmt1** device, enter:

```
savevg -mf/dev/rmt1 data2
```

4. To back up the **data2** volume group, excluding the files listed in the **/etc/exclude.data2** file, enter:

```
savevg -ief/dev/rmt1 data2
```

5. To back up the volume group **my_vg** to the tape in **/dev/rmt0** and then verify the readability of file headers, enter:

```
savevg -f /dev/rmt0 -V my_vg
```

6. To back up the **uservg** volume group to the UDFS capable device **/dev/usbms0**, enter the following command:

```
savevg -i -f /dev/usbms0
```

Files

Item	Description
/image.data	Used when the volume group is rootvg .
/tmp/vgdata/vgname /vgname.data	Used when the volume group is not rootvg and where <i>vgname</i> is the name of the volume group.

savewpar Command

Purpose

Finds and backs up all files belonging to a specified workload partition.

Syntax

```
savewpar [-a] [-A] [-B] [-b Blocks] [-e] [-f Device] [-i] [-m] [-N] [-p] [-T] [-v] [-V] [-X] [-Z]  
[-P] WparName
```

Description

The **savewpar** command finds and backs up all files belonging to a specified workload partition (WPAR). The **savewpar** command uses the data file created by the **mkwpardata** command. This data file is located in the following directory, using the form:

```
/tmp/wpardata/WparName/image.data
```

The *WparName* variable reflects the name of the WPAR. The **savewpar** command uses this file to create a backup image that can be used by the **restwpar** command to re-create a workload partition. For more information, see the **restwpar** command.

To back up customized (not including *rootvg*) volume groups, see the **savevg** command.

Restriction:

- You cannot use the **savewpar** command to create a bootable tape. For best performance, properly end applications that open and close files frequently before you run the **savewpar** command.
- You must not run the **savewpar** command during an AIX live kernel update operation.

You cannot use the **savewpar** command to create a bootable tape. For best performance, properly end applications that open and close files frequently before you run the **savewpar** command.

Flags

Item	Description
-a	Does not backup extended attributes or NFS version 4 (NFS4) access control lists (ACLs).
-A	Backs up the data management application programming interface (DMAPI) file system files.
-B	Does not backup the files residing in the writable <i>namefs-mounted</i> file systems. The default is to include files from the writable <i>namefs-mounted</i> file systems in the backup.
-b Blocks	Specifies the number of 512-byte blocks to write in a single output operation. If you do not specify this parameter, the backup command uses a default value for the physical device that you selected. Larger values result in larger physical transfers to tape devices. The value that you specified must be a multiple of the physical block size of the device being used.

Item	Description
-e	Excludes files specified in the /etc/exclude.WparName file from being backed up by this command. Tip: If you want to exclude certain files from the backup, create the /etc/exclude.WparName file, with an ASCII editor, and enter the patterns of file names that you do not want to be included in the WPAR backup image. The patterns in this file are input to the pattern-matching conventions of the grep command to determine which files is to be excluded from the backup. All of the files are backed up relatively from the base directory (marked with the dot character ".") of the WPAR. To exclude any file or directory for which it is important to have the search match the string at the beginning of the line, use the caret character (^) as the first character in the search string, followed by the dot character (.), and the file name or directory to be excluded. For example, to exclude all of the contents of the /tmp directory, and avoid excluding any other directories that have the /tmp in the path name, edit the exclude file to read as follows: <pre>^./tmp/</pre> If the file name or the directory being excluded is a substring of another file name or directory, use the caret character (^) followed by the dot character (.) to indicate that the search begins at the beginning of the line, or use the dollar sign (\$) to indicate that the search ends at the end of the line.
-f Device	Specifies the device or the file name that the image is to be stored on. The default value is the /dev/rmt0 device.
-i	Creates the data file by calling the mkwpardata command.
-m	Creates the data file with map files by calling the mkwpardata command with the -m flag.
-N	Backs up files from writable NFS-mounted file systems in the mount group for the workload partition. By default, the command does not back up files from writable NFS-mounted file systems. Requirement: For NFS4-mounted file systems, the local and remote system must belong to the same security domain to properly establish ownership of the files on the remote server. If this is not the case, do not use the -N flag.
-p	Disables software packing of the files when they are backed up. Some tape drives use their own packing or compression algorithms.

Item	Description
-T	Create a backup by using snapshots. This flag applies only for JFS2 file systems. When you specify the -T flag to use snapshots for creating a backup for the workload partition, external JFS2 snapshots are created. Snapshots allow for a point-in-time image of a JFS2 file system and thus, do not require a system to be set in a temporarily inactive state. The size of the snapshot is 2% - 15% of the size of the file system. The snapshot logical volumes are removed when backup operation is complete. However, snapshots are not removed if a file system already has other snapshots. Additionally, if a file system has internal snapshots, external snapshots cannot be created and snapshots are not used for creating the backup of the file system. The use of the -T flag does not affect any JFS file systems that are present in the volume group that is being backed up.
-v	Specifies the verbose mode. Lists files when they are backed up.
-V	Verifies a tape backup. With the -V flag, the savewpar command verifies each file header on the backup tape and reports any reading errors when they occur.
-X	Specifies that the /tmp file system must be automatically expanded if necessary. Requirement: The -X flag is only applicable with the -i or -m flag, if necessary. Note: This file system expansion is not used to expand the device file system, where the backup image is saved even if device file system is the same /tmp file system.
-Z	Specifies that the Encrypted File System (EFS) information for all the files, directories, and file systems is not backed up. The flag runs the backup command with the -Z flag.
-P	Exclude files from the packing option listed in the /etc/exclude_packing directory.

Parameters

Item	Description
<i>WparName</i>	Specifies the name of the workload partition to be backed up.

Examples

1. To back up the userwpar workload partition to the default tape drive (**dev/rmt0**) and create a new **/tmp/wpardata/userwpar/image.data** file, enter the following command:

```
savewpar -i userwpar
```

2. To back up the wpar2 workload partition and create map files along with a new **/tmp/wpardata/wpar2/image.data** file on the **rmt1** device, enter the following command:

```
savewpar -mf/dev/rmt1 wpar2
```

3. To back up the wpar2 workload partition, exclude the files listed in the `/etc/exclude.wpar2` file, enter the following command:

```
savewpar -ief/dev/rmt1 wpar2
```

4. To back up the my_wpar workload partition to the tape in tape drive `/dev/rmt0` and then verify the readability of the file headers, enter the following command:

```
savewpar -f /dev/rmt0 -V my_wpar
```

5. To exclude all of the contents of the scratch directory, edit the exclude file to read as follows:

```
/scratch/
```

6. To exclude all of the contents of the `/tmp` directory, and avoid excluding any other directories that have the `/tmp` in the path name, edit the exclude file to read as follows:

```
^./tmp/
```

7. To back up the **wpar2** workload partition and create a new `/tmp/wpardata/userwpar/image.data` file to the UDFS capable device `/dev/usbms0`, enter the following command:

```
savewpar -f /dev/usbms0 wpar2
```

SMIT Fast Path

1. To create a workload partition backup, enter the following SMIT fast path:

```
smit savewpar
```

2. To list the contents of a workload partition backup that was created with the **savewpar** command, enter the following SMIT fast path:

```
smit lssavewpar
```

3. To restore individual files from a workload partition backup, enter the following SMIT fast path:

```
smit restwpar
```

Files

Item	Description
<code>/tmp/wpardata/WparName /WparName.data</code>	Used where the value for the <i>WparName</i> is the name of the tworkload partition.
<code>/etc/exclude.WparName</code>	Contains the files to be excluded from backup.

scan Command

Purpose

Produces a one line per message scan listing.

Syntax

```
scan [ +Folder ] [ Messages ] [ -form FormFile | -format String ] [ -noheader | -header ] [ -clear | -noclear ]  
[ -help ]
```


Description

The **scan** command displays a line of information about the messages in a specified folder. Each line gives the message number, date, sender, subject, and as much of the message body as possible. By default, the **scan** command displays information about all of the messages in the current folder.

If a + (plus sign) is displayed after the message number, the message is the current message in the folder. If a - (minus sign) is displayed, you have replied to the message. If an * (asterisk) is displayed after the date, the Date: field was not present and the displayed date is the last date the message was changed.

Flags

Item	Description
-clear	Clears the display after sending output. The scan command uses the values of the \$TERM environment variable to determine how to clear the display. If standard output is not a display, the scan command sends a form feed character after sending the output.
+Folder	Specifies which folder to scan. The default is the current folder.
-form <i>FormFile</i>	Displays the scan command output in the alternate format described by the <i>FormFile</i> variable.
-format <i>String</i>	Displays the scan command output in the alternate format described by the <i>String</i> variable.
-header	Displays a heading that lists the folder name and the current date and time.
-help	Lists the command syntax, available switches (toggles), and version information. Note: For Message Handler (MH), the name of this flag must be fully spelled out.
<i>Messages</i>	Displays information about each specified message in the specified folder. You can use the following references when specifying messages: Number Specifies the number of the message. Sequence Specifies a group of messages specified by the user. Recognized values include: all All messages in a folder. This is the default. cur or . (period) Current message. first First message in a folder. last Last message in a folder. next Message following the current message. prev Message preceding the current message.
-noclear	Prevents clearing of the terminal after sending output. This is the default.
-noheader	Prevents display of a heading. This is the default.

Item	Description
-width <i>Number</i>	Sets the number of columns in the scan command output. The default is the width of the display.

Profile Entries

The following entries are entered in the *UserMhDirectory/.mh_profile* file:

Item	Description
Alternate-Mailboxes:	Specifies the mailboxes.
Current-Folder:	Sets the default current folder.
Path:	Specifies the <i>UserMhDirectory</i> .

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To get a one-line list of all the messages in the current folder, enter:

```
scan
```

The system responds with a message similar to the following:

```
3  04/17 dale@athena Status meeting <<The weekly status meeting
5  04/20 tom@venus   Due Dates    <<Your project is due to
6  04/21 dawn@tech   Writing Clas  <<There will be a writing
```

2. To get a one-line list of messages 11 through 15 in the test folder, enter:

```
scan  +test 11-15
```

The system responds with a message similar to the following:

```
11 04/16 karen@anchor Meeting    <<Today's meeting is at 2 p.m.
12 04/18 tom@venus   Luncheon  <<There will be a luncheon to
14 04/20 dale@athena First Draft <<First drafts are due
15 04/21 geo@gtwn    Examples  <<The examples will be written
```

Files

Item	Description
\$HOME/.mh_profile	Contains the MH user profile.
/etc/mh/scan.size	Contains a sample scan format string.
/etc/mh/scan.time	Contains a sample scan format string.
/etc/mh/scan.timely	Contains a sample scan format string.
/usr/bin/scan	Contains the executable form of the scan command.

sccs Command

Purpose

Administration program for SCCS commands.

Syntax

sccs [**-r**] [**-dPath**] [**-pPath**] *Command* [*CommandFlags*] *File* ...

Description

The **sccs** command is an administration program that incorporates the set of Source Code Control System (SCCS) commands into the operating system. Additionally, the **sccs** command can be used to assign or reassign file ownership (see the **-r** flag).

The **sccs** command activates a specified *Command* having the specified flags and arguments. Each file is normally placed in a directory named SCCS and named **s.filename**. The directory SCCS is assumed to exist relative to the working directory (unless the **-p** flag is used).

Two types of commands can be used in the **sccs** command syntax sentence. The first type consists of 14 **sccs** commands that can be entered at the prompt. The second type, pseudo-commands, can be used only as part of the **sccs** command syntax. There are 12 pseudo-commands, which perform the following actions:

Item	Description
edit	Equivalent to the get -e command.
delget	Performs a delta command on the named files and then gets new versions. The new versions of the files have expanded identification keywords and are not editable. Flags: -m, -p, -r, -s, -y Can be passed to the delta command. -b, -c, -i, -l, -s, -x Can be passed to the get command.
deledit	Equivalent to the delget pseudo-command, except that the get portion of the sentence includes the -e flag. The deledit pseudo-command is useful for creating a checkpoint in your current editing session. Flags: -m, -p, -r, -s, -y Can be passed to the delta command. -b, -c, -i, -l, -s, -x Can be passed to the get command.
create	Creates an SCCS file, copying the initial contents from a file of the same name. If the file is successfully created, the original file is renamed with a comma on the front. You do not have to move or remove the original file as with the admin command. Flags: Accepts the same flags as the admin command. The -i flag is implied.

Item	Description
fix	Removes a named delta, but leaves a copy of the delta with changes intact. This pseudo-command is useful for fixing small compiler errors. This pseudo-command does not keep a record of changes made to the file. Flags: -rSID Indicates a required flag.
clean	Removes all files from the current directory or from the designated directory that can be recreated from SCCS files. Does not remove files that are in the process of being edited. Flags: -b Ignores branches when determining which files are being edited. Branches being edited in the same directory can be lost.
unedit	Equivalent to the unget command. Any changes made since the get command was used are lost.
info	Lists all files being edited. Flags: -b Ignores branches when determining which files are being edited. -u [Argument] Lists only the files being edited by you or the user named by the <i>Argument</i> parameter.
check	Prints all files being edited. Returns a nonzero exit status if a file is being edited. The check program can be used in a makefile to ensure that files are complete before a version is installed. Check the return code before performing the install. Flags: -b Ignores branches when determining which files are being edited. -u [Argument] Lists only the files being edited by you or the user named by the <i>Argument</i> parameter.
tell	Lists all files being edited, with a new line after each entry, on standard output. Flags: -b Ignores branches when determining which files are being edited. -u [Argument] Lists only the files being edited by you or the user named by the <i>Argument</i> parameter.

Item	Description
diffs	Shows the difference between the current version of the program you are editing and the previous deltas. Flags: -r, -c, -i, -x, -t Can be passed to the get command. -l, -s, -e, -f, -h, -b Can be passed to the diff (not sccsdiff) command. -C Can be passed to the diff (not sccsdiff) command as a -c flag.
print (<i>filename(s)</i>)	Prints verbose information about the named files. If the PROJECTDIR environment variable is set, its value determines the working directory. If this value begins with a / (slash), it is used directly. Otherwise, the value is interpreted as a user name whose home directory is examined for a subdirectory named src or source . If found, that subdirectory is used as the working directory.

Flags

Item	Description
-dPath	Specifies a working directory for the SCCS files. The default is the current directory. The -d flag is prefixed to the entire path name of a file. When the PROJECTDIR environment variable is set and the -d flag is used, the command line overrides the environment value in determining the working directory.
-p	Specifies a path name for the SCCS files. The default is the SCCS directory. The -p flag is inserted before the final component of the path name. All flags specified after the command are passed to that command during execution. For a description of command flags, see the appropriate command description. Example: <code>sccs -d/x -py get a/b</code> converts to: <pre>get /x/a/y/s.b</pre> This option is used to create aliases. For example: <pre>alias syssccs sccs -d/usr/src</pre> causes the syssccs command to become an alias command that can be used as follows: <pre>syssccs get cmd/who.c</pre> When used in this context, the above command will check the /usr/src/cmd/SCCS directory for the s.who.c file.
-r	Runs the sccs command as the real user instead of as the effective user to which the sccs command is set (using the set user id command). Certain commands, such as the admin command, cannot be run as set user id , which would allow anyone to change the authorizations. Such commands are always run as the real user.

Exit Status

This command returns the following exit values:

Item	Description

0	Successful completion.
>0	An error occurred.

Examples

1. To get a file for editing, edit it, and then produce a new delta, enter:

```
sccs get -e file.c
ex file.c
sccs delta file.c
```

2. To get a file from another directory, enter:

```
sccs -p/usr/src/sccs/ get cc.c
```

OR

```
sccs get /usr/src/sccs/s.cc.c
```

3. To get a list of files being edited that are not on branches, enter:

```
sccs info -b
```

Files

Item	Description
/usr/bin/sccs	Contains the sccs command, which is the administration program for the SCCS commands.

sccsdiff Command

Purpose

Compares two versions of a SCCS file.

Syntax

```
sccsdiff -rSID1 -rSID2 [ -p ] [ -sNumber ] File ...
```

Description

The **sccsdiff** command reads two versions of an Source Code Control System (SCCS) file, compares them, and then writes to standard output the differences between the two versions. Any number of SCCS files can be specified, but the same arguments apply to all files.

Flags

Item	Description
-p	Pipes the output through the pr command.
-rSID1	Specifies <i>SID1</i> as one delta of the SCCS file for the sccsdiff command to compare.

Item	Description
-rSID2	Specifies <i>SID2</i> as the other delta of the SCCS file for the sccsdiff command to compare.
-sNumber	Specifies the file-segment size for the bdiff command to pass to the diff command. This is useful when the diff command fails due to a high system load.

Examples

To display the difference between versions 1.1 and 1.2 of SCCS file `s.test.c`, enter:

```
sccsdiff -r1.1 -r1.2 s.test.c
```

Files

Item	Description
/usr/bin/sccsdiff	Contains the SCCS sccsdiff command. The sccsdiff command supports multibyte character set (MBCS) data for the file names.

sccshelp Command

Purpose

Provides information about a SCCS message or command.

Syntax

sccshelp [*ErrorCode*] [*Command*]

Description

The **sccshelp** command displays information about the use of a specified Source Code Control System (SCCS) command or about messages generated while using the commands. Each message has an associated code, which can be supplied as part of the argument to the **sccshelp** command. Zero or more arguments may be supplied. If you do not supply an argument, the **sccshelp** command prompts for one. You may include any of the SCCS commands as arguments to the **sccshelp** command.

The *ErrorCode* parameter specifies the code, consisting of numbers and letters, that appears at the end of a message. For example, in the following message, (cm7) is the code:

```
There are no SCCS identification keywords in the file. (cm7)
```

Examples

To get **sccshelp** on the **rmDEL** command and two error codes, enter:

```
$ sccshelp rmDEL gee ad3
```

The **sccshelp** command replies:

```
rmDEL:
rmDEL -r<SID> <file> ...
ERROR:
1255-141 gee is not a valid parameter. Specify a valid command or error code.
ad3:
The header flag you specified is not recognized.
The header flag you supplied with the -d or the -f flag is not correct.
Choose a valid header flag.
```

File

Item	Description
<code>/usr/bin/sccshelp</code>	Contains the SCCS sccshelp command.

schedo Command

Purpose

Manages scheduler tuning parameters of the processor.

Syntax

```
schedo [ -p | -r [-K] [-y] { -o Tunable[= Newvalue] }
```

```
schedo [ -p | -r [-K] [-y] { -d Tunable }
```

```
schedo [ -p | -r [-K] [-y] -D
```

```
schedo [ -p | -r | -K] [-F] -a
```

```
schedo -h [Tunable]
```

```
schedo [-F] [-K] -L [Tunable]
```

```
schedo [-F] [-K] -x [Tunable]
```

Note: Multiple flags -o, -d, -x, and -L flags are allowed

Description

Note: The **schedo** command can only be executed by root.

Use the **schedo** command to configure scheduler tuning parameters. This command sets or displays current or next boot values for all scheduler tuning parameters. This command can also make permanent changes or defer changes until the next reboot. Whether the command sets or displays a parameter is determined by the accompanying flag. The -o flag performs both actions. It can either display the value of a parameter or set a new value for a parameter.

Understanding the effect of changing tunable parameters

Misuse of this command can cause performance degradation or operating-system failure. Be sure that you have studied the appropriate tuning sections in the *Performance management* before using the **schedo** command to change system parameters.

Before modifying any tunable parameter, you must first carefully read about all its characteristics in the [Tunable Parameters](#) section, and follow any Refer To pointer, in order to fully understand its purpose.

You must then make sure that the Diagnosis and Tuning sections for this parameter truly apply to your situation and that changing the value of this parameter could help improve the performance of your system.

If the Diagnosis and Tuning sections both contain only "N/A", you must never change this parameter unless specifically directed by AIX development.

Priority-calculation parameters

The priority of most user processes varies with the amount of processor time the process has used recently. The processor scheduler's priority calculations are based on two parameters that are set

with **schedo**, *sched_R* and *sched_D*. The *sched_R* and *sched_D* values are in thirty-seconds (1/32); that is, the formula used by the scheduler to calculate the amount to be added to a process's priority value as a penalty for recent processor use is:

```
CPU penalty = (recently used CPU value of the process) * (r/32)
```

and the once-per-second recalculation of the recently used processor value of each process is:

```
new recently used CPU value = (old recently used CPU value of the process) * (d/32)
```

Both *r* (*sched_R* parameter) and *d* (*sched_D* parameter) have default values of 16. This maintains the processor scheduling behavior of previous versions of the operating system. Before experimenting with these values, you must be familiar with Tuning the processor scheduler in the Performance Management Guide.

Memory-load-control parameters

The operating system scheduler performs memory load control by suspending processes when memory is over committed. The system does not swap out processes; instead pages are *stolen* as they are needed to fulfill the current memory requirements. Typically, pages are stolen from suspended processes. Memory is considered over committed when the following condition is met:

Item	Description
$p * h s$	where: <i>p</i> is the number of pages written to paging space in the last second <i>h</i> is an integer specified by the v_repage_hi parameter <i>s</i> is the number of page steals that have occurred in the last second

A process is suspended when memory is over committed and the following condition is met:

Item	Description
$r * p f$	where: <i>r</i> is the number of repages that the process has accumulated in the last second <i>p</i> is an integer specified by the v_repage_proc parameter <i>f</i> is the number of page faults that the process has experienced in the last second

In addition, fixed-priority processes and kernel processes are exempt from being suspended.

The term *repages* refers to the number of pages belonging to the process, which were reclaimed and are soon after referenced again by the process.

The user also can specify a minimum multiprogramming level with the **v_min_process** parameter. Doing so ensures that a minimum number of processes remain active throughout the process-suspension period. Active processes are those that are runnable and waiting for page I/O. Processes that are waiting for events and processes that are suspended are not considered active, nor is the wait process considered active.

Suspended processes can be added back into the mix when the system has stayed below the over committed threshold for *n* seconds, where *n* is specified by the **v_sec_wait** parameter. Processes are added back into the system based, first, on their priority and, second, on the length of their suspension period.

Before experimenting with these values, you must be thoroughly familiar with [VMM memory load control tuning with the schedo command](#) in the Performance Management Guide.

Time-slice-increment parameter

The **schedo** command can also be used to change the amount of time the operating system allows a given process to run before the dispatcher is called to choose another process to run (the time

slice). The default value for this interval is a single clock tick (10 milliseconds). The timeslice tuning parameter allows the user to specify the number of clock ticks by which the time slice length is to be increased.

In AIX Version 4, this parameter only applies to threads with the SCHED_RR scheduling policy. See Scheduling Policy for Threads.

fork() retry interval parameter

If a **fork()** subroutine call fails because there is not enough paging space available to create a new process, the system retries the call after waiting for a specified period of time. That interval is set with the **pacefork** tuning parameter.

Special terminology for symmetric multithreading

Multiple run queues are supported. Under this scheme each processor has its own run queue. POWER5 processors support symmetric multithreading, where each physical processor has two execution engines, called *hardware threads*. Each hardware thread is essentially equivalent to a single processor. Symmetric multithreading is enabled by default, but it can be disabled (or re-enabled) dynamically. When symmetric multithreading is enabled, each hardware thread services a separate run queue. For example, on a 4-way system when symmetric multithreading is disabled or not present, there are 4 run queues in addition to the global run queue. When symmetric multithreading is enabled, there are 8 run queues in addition to the global run queue.

The hardware threads belonging to the same physical processor are referred to as *sibling threads*. A *primary sibling thread* is the first hardware thread of the physical processor. A *secondary sibling thread* is the second hardware thread of the physical processor.

Virtual processor management

More virtual processors can be defined than are needed to handle the work in a partition. The overhead of dispatching virtual processors can be reduced by using fewer virtual processors without a decrease in overall processor usage or a lack of virtual processors. Virtual processors are not dynamically removed from the partition, but instead are not used and are used again only when additional work is available. Each virtual processor uses a maximum of one physical processor. The number of virtual processes needed is determined by rounding up the sum of the physical processor utilization and the **vpm_xvcpus** tunable:

```
number = ceiling( p_util + vpm_xvcpus )
```

Where *number* is the number of virtual processors that are needed, *p_util* is the physical processor utilization, and **vpm_xvcpus** is a tunable that specifies the number of additional virtual processors to enable. If *number* is less than the number of currently enabled virtual processors, a virtual processor will be disabled. If *number* is greater than the number of currently enabled virtual processors, a disabled virtual processor will be enabled. Threads that are attached to a disabled virtual processor are still allowed to run on the disabled virtual processor.

Node load

The *node load*, or simply *load*, is the average run queue depth across all run queues, including the global run queue multiplied by 256, and is strongly smoothed over time. For example, a load of 256 means that if we have 16 processors (including symmetric multithreading processors), then we have had approximately 16 runnable jobs in the system for the last few milliseconds.

>VPM throughput mode changes for Power10

When Power10 systems run in shared processor mode, the default value of the throughput mode for the virtual processor management (VPM) is 2. When you migrate the system to or from a Power10 system, and later, the AIX operating system automatically changes the default value of the throughput mode for the VPM. During boot operation, the AIX operating system selects the default value of the **vpm_throughput_mode** tunable parameter of the **schedo** command based on the type of server on which the LPAR is running. The value of the **vpm_throughput_mode** tunable parameter that is selected by the AIX operating system is preserved and used on the destination server. On Power10

systems, you can switch to the recommended value of 2 for the **vpm_throughput_mode** tunable parameter by using the following command:

```
schedo -d vpm_throughput_mode
```

Note: When you migrate to or from a Power10 system, and later, it is recommended to move the operating system level to 7300-00, 7200-05-03-2147, 7200-04-05-2148, 7100-05-09-2148 or later. Without this feature that enables you to change the **vpm_throughput_mode** tunable parameter by using the **schedo** command, updating the operating system levels to 7200-05-00-2037, 7200-04-03-2038, and 7100-05-07-2037 might override the value set for the **vpm_throughput_mode** tunable parameter.



Flags

Item	Description
-a	Displays current value, reboot value (when used with the -r option), Live Update value (when used with the -K option), or permanent values for all tunable parameters (when used with the -p option), one per line in pairs <i>Tunable = Value</i> . For the permanent option, a value is only displayed for a parameter if its reboot and current values are equal. Otherwise NONE displays as the value.
-d <i>Tunable</i>	Resets the <i>Tunable</i> parameter to its default value. If a tunable parameter needs to be changed (that is, it is currently not set to its default value), and -r is not used in combination, it won't be changed but a warning is displayed.
-D	Resets all <i>Tunable</i> parameters to their default values. If <i>Tunable</i> parameters that need to be changed because they are not set to their default values meet one or more of the following sets of criteria, a warning message is displayed and no change is made: If a tunable parameter that you want to change, and that is currently not set to its default value, meets one or more of the following sets of criteria, a warning message is displayed and no change is made to the parameter: • The tunable is of type Bosboot or Reboot. • The tunable is of type Incremental and was changed from its default value, and -r is not used in combination. • The tunable parameter is of type Reboot and is supported across the Live Update operation.
-F	Forces the display of restricted tunable parameters when you specify the -a , -L or -x options on the command line, to list all of the tunables. If you do not specify the -F flag, restricted tunables are not included, unless they are specifically named in association with a display option.
-h [<i>Tunable</i>]	Displays help about the <i>Tunable</i> parameter if one is specified. Otherwise, displays the schedo command usage statement.
-k	Sets the tunable parameter value in both <code>/etc/tunables/nextboot</code> and <code>/etc/tunables/nextliveupdate</code> files. The -K flag can be used only with the -r flag. When you specify the -K flag with the -r and -d (or -D) flags, the tunable parameter value is set to its default value in the <code>/etc/tunables/nextboot</code> and <code>/etc/tunables/nextliveupdate</code> files to be used during the next boot or Live Update operations. When you specify the -K flag with the -L or -x flag, the schedo command displays the Live Update values.

Item	Description
-L [<i>Tunable</i>]	Lists the characteristics of one or all tunable parameters, one per line, by using the following format. If you specify the -K flag with the -L flag, the Live Update values are also displayed. <pre> NAME TYPE DEPENDENCIES ----- v_repage_hi 0 0 0 0 2047M D v_repage_proc 4 4 4 0 2047M D v_sec_wait 1 1 1 0 2047M seconds D ... where: CUR = current value DEF = default value BOOT = reboot value MIN = minimal value MAX = maximum value UNIT = tunable unit of measure TYPE = parameter type: D (for Dynamic), S (for Static), R (for Reboot), B (for Bosboot), M (for Mount), I (for Incremental), C (for Connect), and d (for Deprecated) DEPENDENCIES = list of dependent tunable parameters, one per line </pre>
-o <i>Tunable</i> [=Newvalue]	Displays or sets the value of a tunable parameter to a new value. > The <code>/etc/tunables/usermodified</code> file is updated with the new tunable parameter value whenever you use the -o flag to change the value of a dynamic tunable. < If a tunable parameter that you want to change meets one or more of the following sets of criteria, a warning message is displayed and no change is made to the parameter: - The tunable parameter is of type Bosboot or Reboot. - The tunable parameter is of type Incremental, and its current value is greater than the specified value, and the -r flag is not used in combination. - The tunable parameter is of type Reboot, and the tunable parameter is supported across the Live Update operation. When you specify the -r flag with the -o flag without specifying a new value, the next boot value for tunable is displayed. > When you specify the -K flag with the -o flag without specifying a new value, the next Live Update value for the tunable parameter is displayed. < When you specify the -p flag with the -o flag without specifying a new value, a value is displayed only if the current and next boot values for the tunable are the same. Otherwise, NONE is displayed as the value.
-p	Makes changes apply to both current and reboot values, when used in combination with -o, -d or -D, that is, turns on the updating of the <code>/etc/tunables/nextboot</code> file in addition to the updating of the current value. These combinations cannot be used on Reboot and Bosboot type parameters because their current value can't be changed. When used with -a or -o without specifying a new value, values are displayed only if the current and next boot values for a parameter are the same. Otherwise NONE displays as the value.
-r	Makes changes apply to reboot values when used in combination with -o, -d or -D, that is, turns on the updating of the <code>/etc/tunables/nextboot</code> file. If any parameter of type Bosboot is changed, the user will be prompted to run bosboot. When used with -a or -o without specifying a new value, next boot values for tunables display instead of current values. > When used with the -K flag, changes apply to both the <code>/etc/tunables/nextboot</code> and <code>/etc/tunables/nextliveupdate</code> files. <
-x [<i>Tunable</i>]	Lists characteristics of one or all tunable parameters, one per line, using the following (spreadsheet) format. If you specify the -K flag with the -x flag, the Live Update values are also displayed. <pre> tunable,current,default,reboot,min,max,unit,type,{dtunable} where: current = current value default = default value reboot = reboot value min = minimal value max = maximum value unit = tunable unit of measure type = parameter type: D (for Dynamic), S (for Static), R (for Reboot), B (for Bosboot),M (for Mount), I (for Incremental), C (for Connect), and d (for Deprecated) dtunable = space separated list of dependent tunable parameters </pre>
-y	Suppresses the confirmation prompt before executing the bosboot command.

Note: Options **-o**, **-d**, and **-D** are not supported within a workload partition because they attempt to change the value of a scheduler tunable parameter.

If you make any change (with the **-o**, **-d**, or **-D** options) to a restricted tunable parameter, it results in a warning message that a tunable parameter of the restricted-use type, has been modified. If you also specified the **-r** or **-p** options on the command line, you will be prompted to confirm the change. In addition, at system reboot, restricted tunables that are displayed in the **/etc/tunables/nextboot** file, which were modified to values that are different from their default values (using a command line specifying the **-r** or **-p** options), causes an error log entry that identifies the list of these modified tunables.

When modifying a tunable, you can specify the tunable value using the abbreviations such as K, M, G, T, P and E to indicate units. See units. The following table shows the prefixes and values that are associated with the number abbreviations:

Abbreviation	Power of 2
K	1024
M	1 048 576
G	1 073 741 824
T	1 099 511 627 776
P	1 125 899 906 842 624
E	1 152 921 504 606 846 976

Thus, a tunable value of 1024 might be specified as 1K.

Any change (with **-o**, **-d** or **-D**) to a parameter of type Mount results in a message displaying to warn you that the change is only effective for future mountings.

Any change (with **-o**, **-d** or **-D** flags) to a parameter of type Connect will result in **inetd** being restarted, and in a message being displayed to warn the user that the change is only effective for future socket connections.

Any attempt to change (with **-o**, **-d** or **-D**) a parameter of type Bosboot or Reboot without **-r**, results in an error message.

Any attempt to change (with **-o**, **-d** or **-D** but without **-r**) the current value of a parameter of type Incremental with a new value smaller than the current value, results in an error message.

Tunable parameters type

All the tunable parameters manipulated by the tuning commands (**no**, **nfso**, **vmo**, **ioo**, **raso**, and **schedo**) have been classified into these categories:

Item	Description
Dynamic	If the parameter can be changed at any time
Static	If the parameter can never be changed
Reboot	If the parameter can only be changed during reboot
Bosboot	If the parameter can only be changed by running bosboot and rebooting the machine
Mount	If changes to the parameter are only effective for future file systems or directory mounts
Incremental	If the parameter can only be incremented, except at boot time
Connect	If changes to the parameter are only effective for future socket connections
Deprecated	If changing this parameter is no longer supported by the current release of AIX.

For parameters of type Bosboot, whenever a change is performed, the tuning commands automatically prompt the user to ask if they want to execute the **bosboot** command. For parameters of type Connect, the tuning commands automatically restart the **inetd** daemon.

Note that the current set of parameters managed by the **schedo** command only includes Dynamic, and Reboot types.

Compatibility mode

When running in pre 5.2 compatibility mode (controlled by the **pre520tune** attribute of **sys0**, see **AIX 5.2 compatibility mode** in the *Performance management*), reboot values for parameters, except those of type Bosboot, are not really meaningful because in this mode they are not applied at boot time.

In pre 5.2 compatibility mode, setting reboot values to tuning parameters continues to be achieved by imbedding calls to tuning commands in scripts called during the boot sequence. Parameters of type **Reboot** can therefore be set without the **-r** flag, so that existing scripts continue to work.

This mode is automatically turned ON when a machine is MIGRATED to AIX 5.2. For complete installations, it is turned OFF and the reboot values for parameters are set by applying the content of the **/etc/tunables/nextboot** file during the reboot sequence. Only in that mode are the **-r** and **-p** flags fully functional. See **Kernel Tuning** in the *Performance Tools Guide and Reference* for more information.

Tunable parameters

For default values and range of values for tunables, refer **schedo** command help (**-h** *<tunable_parameter_name>*).

Item	Description
affinity_lim	Purpose: Sets the number of intervening dispatches after which the SCHED_FIFO2 policy no longer favors a thread. Tuning: Once a thread is running with SCHED_FIFO2 policy, tuning of this variable may or may not have an effect on the performance of the thread and workload. Ideal values must be determined by trial and error.
big_tick_size	Purpose: Sets physical tick interval and synchronizes ticks across cpus. Tuning: The big_tick_size value times 10 ms as a tick interval, and must evenly divide into 100. Use of this parameter will make system statistics less accurate.
ded_cpu_donate_thresh	Purpose: Specifies the utilization threshold for donation of a dedicated processor. Tuning: In a dedicated processor partition that is enabled for donation, idle processor capacity can be donated to the shared processor pool for use by shared processor partitions. If a dedicated processor's utilization is less than this threshold, the dedicated processor will be donated for use by other partitions when the processor is idle. If a dedicated processor's utilization is equal to or greater than this threshold, the dedicated processor will not be donated for use by other partitions when the dedicated processor is idle.
fixed_pri_global	Purpose: Keep fixed priority threads on global run queue. Tuning: If 1, then fixed priority threads are placed on the global run queue.
force_grq	Purpose: Keep non-MPI threads on the global run queue. Tuning: If 1, only MPI and bound threads will use local run queues.
> hash_instructions <	> Purpose: Enables return-oriented programming (ROP) protection instructions that are supported by Power10 systems, and later. Tuning: If this tunable parameter is set to 1, ROP protection instructions are enabled for the new processes. If this tunable parameter is set to 0, ROP protection instructions are disabled for the new processes. <

Item	Description
maxspin	Purpose: Sets the number of times to spin on a kernel lock before going to sleep. Tuning: Increasing the value on MP systems may reduce idle time; however, it might also waste CPU time in some situations. Increasing it on uniprocessor systems is not recommended.
pacefork	Purpose: The number of clock ticks to wait before retrying a failed fork call that has failed for lack of paging space. Tuning: Use when the system is running out of paging space and a process cannot be forked. The system will retry a failed fork five times. For example, if a fork() subroutine call fails because there is not enough paging space available to create a new process, the system retries the call after waiting the specified number of clock ticks.
proc_disk_stats	Purpose: A value of 1 enables and a value of 0 disables the process scope disk statistics. The default value is 1 and ranges from 0 to 1. Tuning: Disabling process scope disk statistics improves performance when the statistics are not wanted.
sched_D	Purpose: Sets the short term CPU usage decay rate. Tuning: The default is to decay short-term CPU usage by 1/2 (16/32) every second. Decreasing this value enables foreground processes to avoid competition with background processes for a longer time.
sched_R	Purpose: Sets the weighting factor for short-term CPU usage in priority calculations. Tuning: Run the command ps al. If you find that the PRI column has priority values for the foreground processes (those with NI values of 20) that are higher than the PRI values of some background processes (NI values > 20), you can reduce the r value. The default is to include 1/2 (16/32) of the short term CPU usage in the priority calculation. Decreasing this value makes it easier for foreground processes to compete.
tb_balance_S0	Purpose: Controls SMT-cores busy balancing. Tuning: A value of 0 indicates that the balancing is disabled. A value of 1 indicates that the balancing is enabled only within MCMs (S2 groups). A value of 2 indicates fully enabled.
tb_balance_S1	Purpose: Controls processor busy balancing. Tuning: A value of 0 indicates that the balancing is disabled. A value of 1 indicates that the balancing is enabled only within MCMs (S2 groups). A value of 2 indicates fully enabled.
tb_threshold	Purpose: Number of ticks to consider a thread busy for the purposes of optimization for thread_busy load balancing. Tuning: A value of 100 corresponds to 1 second. The values 10 and 1000 correspond to 0.1 and 10 seconds, respectively.
timeslice	Purpose: The number of clock ticks a thread can run before it is put back on the run queue. Tuning: Increasing the timeslice value can reduce overhead of dispatching threads. The value refers to the total number of clock ticks in a timeslice and only affects fixed-priority processes.
vpm_fold_policy	Purpose: Controls the application of the virtual processor management feature of processor folding in a partition. Tuning: The virtual processor management feature of processor folding can be enabled or disabled based on whether a partition has shared or dedicated processors. In addition, when the partition is in static power saving mode, processor folding is automatically enabled for both shared or dedicated processor partitions. When processor folding is enabled, the vpm_vxcpus tunable can be used to control processor folding. There are 3 bits in vpm_fold_policy to control processor folding: • Bit 0 (0x1): When set to 1, this bit indicates processor folding is enabled if the partition is using shared processors. • Bit 1 (0x2): When set to 1, this bit indicates processor folding is enabled if the partition is using dedicated processors. • Bit 2 (0x4): When set to 1, this bit disables the automatic setting of processor folding when the partition is in static power saving mode. You can perform an OR operation on the Bit 0, Bit 1, and Bit 2 values to form the desired value. Note: > If the Idle power saver option is set to Enabled in the Advanced System Management Interface (ASMI), and if processor utilization falls below the specified threshold values, then you can set the processors to a low frequency or low voltage state. You can also set the vpm_fold_policy tunable parameter at runtime to 0x3. <
vpm_throughput_core_threshold	Specifies the number of cores that must be unfolded before vpm_throughput_mode parameter is honored. Till that, the system behaves with the value of vpm_throughput_mode parameter set as 1.

Item	Description
vpm_throughput_mode	Specifies the desired level of SMT exploitation for scaled throughput mode. A value of 0 gives raw throughput mode behavior. A value of 1 , 2 , or 4 selects the scaled throughput mode and the desired level of SMT exploitation. > On Power10 and later systems, the default value of VPM throughput mode is 2 (scaled throughput mode) when the system is running in shared processor mode. In dedicated processor mode and on POWER9 and earlier systems that are running in shared processor mode, the default value of VPM throughput mode is 0 (raw throughput mode). <
vpm_xvcpus	Purpose: Setting this tunable to a value greater than -1 will enable the scheduler to enable and disable virtual processors based on the partition's CPU utilization. Tuning: The value specified signifies the number of virtual processors to enable in addition to the virtual processors required to satisfy the workload.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To list the current and reboot value, range, unit, type and dependencies of all tunable parameters managed by the schedo command, enter:

```
schedo -L
```

2. To list (spreadsheet format) the current and reboot value, range, unit, type, and dependencies of all tunable parameters managed by the **schedo** command, enter:

```
schedo -x
```

3. To reset v_sec_wait to default, enter:

```
schedo -d v_sec_wait
```

4. To display help on sched_R, enter:

```
schedo -h sched_R
```

5. To set v_min_process to 4 after the next reboot, enter:

```
schedo -r -o v_min_process=4
```

6. To permanently reset all schedo tunable parameters to default, enter:

```
schedo -p -D
```

7. To list the reboot value for all schedo parameters, enter:

```
schedo -r -a
```

scls Command

Purpose

Produces a list of module and driver names.

Syntax

```
scls [ -c | -l ] [ -m sc_module_name ] [ Module ... ]
```


Description

The **scls** command provides a method for the user to query the current **Portable Streams Environment** (PSE) configuration. The **scls** command produces a list of module and driver names. Flags can be used to produce enhanced lists. Any further parameters on the command line are module or driver names, and the output produced is for only those names.

Note: The **scls** command requires the **sc** STREAMS module and the **nuls** driver. If either one is not available, the **scls** command will not be successful.

Flags

Item	Description

-c	Produces a listing showing the number of times an interface routine was called.
-l	Produces a long listing that shows the extension type, major number, and information pertaining to the module_info structure.
-m	Pushes the module pointed to by the <i>sc_module_name</i> to the top of the current stream, just below the stream head.

The **-c** and **-l** flags are mutually exclusive.

Parameters

Item	Description
<i>module</i>	Specifies the name of the modules or drivers for which to output information.
<i>sc_module_name</i>	Specifies a module name that needs to be pushed to the current stream, just below the stream head.

Files

Item	Description
sc	Dynamically loadable STREAMS configuration module
nuls	Dynamically loadable STREAMS null device.

script Command

Purpose

Makes a typescript of a terminal session.

Syntax

script [**-a**] [**-q**] [*File*]

Description

The **script** command makes a typescript of everything displayed on your terminal. The typescript is written to the file specified by the *File* parameter. The typescript can later be sent to the line printer. If no file name is given, the typescript is saved in the current directory with the file name **typescript**.

The script ends when the forked shell exits.

This command is useful for producing hardcopy records when hardcopy terminals are in short supply. For example, use the **script** command when you are working on a CRT display and need a hardcopy record of the dialog.

Because the **script** command sets the **SetUserID** mode bit, due to security reasons the value of LIBPATH variable is unset when the command is invoked. However, LIBPATH is automatically reset in the forked shell if it is defined in the environment file. This behavior is also true for the NLSPATH environment variable. For related information, see the **exec** subroutine.

Flags

Item	Description

-a	Appends the typescript to the specified file or to the typescript file.
-q	Suppresses diagnostic messages.

Files

Item	Description
/usr/bin/script	Contains the script command.

sctpctrl Command

Purpose

Controls and configures Sctp.

Syntax

```
sctpctrl {load|dump|set}  
sctpctrl stats [reset] [interval]  
sctpctrl set {name=value|default [name]}  
sctpctrl get [name]
```

Description

The **sctpctrl** command is used to control and configure the Sctp kernel extension. This command can be used to load and unload the Sctp kernel extension. This can also be used to dump Sctp data and set or retrieve various Sctp tunable. Further, **sctpctrl** command can be used to read and reset the Sctp specific network statistics.

Parameters

Item	Description
load	Loads the Sctp kernel extension if not loaded.
dump	Dump information about internal Sctp structures.
stats [reset] [interval]	Displays Sctp statistics. The optional reset command will clear (zero) the statistics. If the <i>interval</i> parameter (in seconds) is added, the program does not exit, but outputs the statistics every [interval] seconds.

Item	Description
<code>set {name=value default [name]}</code>	Sets the SCTP tunable to a value. If <i>default</i> is specified then it will set all the tunable to their default values. If optional <i>[name]</i> is specified followed by <i>default</i> then it will set tunable described by <i>name</i> to its default value.
<code>get [name]</code>	Gets the value of the tunable described by their optional <i>name</i> parameter. If <i>name</i> parameter is not specified then it gets the values of all the tunable.

Tunable Parameters

The **sctpctrl** command is also used to configure the SCTP tuning parameters. The changes made are not permanent and they have to be set every time a system gets rebooted. The tunables parameters are explained in the following table.

Item	Description		
Parameter	Purpose	Scope	Default
<code>sctp_low_rto</code>	When nonzero, this value is used in place of <i>RTO.min</i> (retransmission time-out). It is specified in terms of milliseconds. Values less than 200 are not allowed. The available time-out values are 200, 250, 300, 350, and so on.	This value is examined each time a new RTT (round trip time) measurement is made and also when RTO is adjusted due to packet loss.	As specified in the RFC 4960 (Request for Comment) document, the default value for this tunable is zero, which means the minimum value of <i>RTO.Min</i> is used, which is 1 second.
<code>sctp_enable_shutdown_guard</code>	When nonzero, this tunable enables a T5-shutdown guard-timer. It is not RFC compliant because it begins timing when association enters shutdown-pending state.	This value is only examined at an association shutdown.	The default value for this tunable is zero, which means that the T5-shutdown guard-timer is not used.
<code>sctp_max_init_attempts</code>	This tunable parameter specifies the maximum number of retransmission attempts that are allowed for the retransmission of the INIT packets.		The default value for this tunable parameter is 8.
<code>sctp_max_init_timeo</code>	When the value of this tunable parameter is nonzero, the tunable value is used instead of the <i>RTO.max</i> parameter (retransmission timeout) for INIT packets. It is specified in milliseconds. Values less than 200 milliseconds are not allowed. The available timeout values are 200, 250, 300, 350, and so on.	This tunable parameter specifies the maximum value to be used when the INIT packets are retransmitted.	The default value for this tunable parameter is zero seconds. In this case, the <code>sctp_rttmax</code> tunable parameter is used.

Item	Description		
<i>sctp_shutdown_guard_timer</i>	When the <i>sctp_enable_shutdown_guard</i> parameter is a nonzero value, this tunable defines the shutdown time-out value in seconds.	This value is only examined at an association shutdown.	The default value is 300 seconds, which is the RFC-specified value for the T5-shutdown guard-timer.
<i>sctp_peerchangespath</i>	When nonzero, this tunable causes a primary path change based on an incoming data chunk from a different path than the current primary path.	This value is examined on every inbound data chunk.	The default value for this tunable is 1, which retains the existing behavior.
<i>sctp_delack_timer</i>	This tunable specifies the timer value in ticks (1 tick = 50 ms (milliseconds)) for the delayed-ack timer.	For an <i>ACCEPTCONN</i> socket, this value is established during setup and is used for all associations that share that socket. For a socket other than an <i>ACCEPTCONN</i> socket, it is set at association creation. So changes to this tunable do not affect associations already in existence.	The default value is 4 ticks (200 ms).
<i>sctp_drop_gapacks</i>	If set to 1, it causes the sender side to drop all <i>GAPACKED</i> packets from the socket send buffer, thus making some space free for new packets. Note: This is an RFC noncompatible tunable and could impact interoperability with other implementations, potentially resulting in a message loss.	This tunable is checked each time <i>GAPACKED</i> packets are processed.	The default value is 0, which means disabled.
<i>sctp_dontdelayack</i>	If set to 1, a <i>SACK</i> packet is sent for every other <i>DATA</i> packet. Otherwise, a delayed-ack timer is started.	Any updates to this tunable have an immediate impact.	The default value is 1.

Item	Description		
<i>sctp_nagle</i>	If set to 1, it ensures that at least 1 MTU (maximum transmission unit) of data is sent.	Any updates to this tunable have an immediate impact.	The default value is 1 (a <i>nagle</i> is enabled).
<i>sctp_maxburst</i>	If nonzero, it limits the maximum number of packets sent out to this value.	Any updates to this tunable have an immediate impact.	The default value is 8 packets.
<i>sctp_rttmax</i>	This tunable specifies the maximum value to be used when RTO computations are made.	Similar to the <i>sctp_low_rto</i> parameter, this value is examined each time a new RTT measurement is made (and RTO calculated with that) and also when RTO is adjusted due to packet loss.	The default value is 60 seconds.
<i>sctp_rttmin</i>	This tunable specifies the minimum value to be used when RTO computations are made.	If the <i>sctp_low_rto</i> parameter is nonzero, this value is ignored. Otherwise, it is examined each time a new RTT measurement is made and when RTO is stopped due to packet loss.	The default value is 1 second, which ensures that the minimum RTO cannot go below that.

Item	Description		
<i>sctp_assoc_maxerr</i>	This tunable sets the overall association error count. If an error count exceeds this value, the association is ended. Currently, this value is ignored. The <i>assoc_maxerr</i> parameter is calculated based on the path error count and number of <i>faddrs</i> .	For an <i>ACCEPTCONN</i> socket, this value is established during setup and is used for all associations that share that socket. For a socket that is not an <i>ACCEPTCONN</i> socket, it is set at association creation. So changes to this tunable do not affect associations already in existence.	The default value is 10.
<i>sctp_path_maxerr</i>	This tunable sets the maximum error count for each destination. If the error count exceeds this value, the path is marked down and an alternative path is chosen.	For an <i>ACCEPTCONN</i> socket, this value is established during setup and is used for all associations that share that socket. For a socket that is not an <i>ACCEPTCONN</i> socket, it is set at association creation. So changes to this tunable do not affect associations already in existence.	The default value is 5.

Item	Description		
<i>sctp_use_checksum</i>	This tunable allows an administrator to use different checksum computation methods. Possible values follows: • 0: CRC32 checksum • 1: No checksum computation is made • 2: Internet checksum. The packets are dropped if different values are used by two peers.	This parameter is examined for each outgoing and incoming packet.	The default value is zero, which is the RFC-specified CRC32 checksum.
<i>sctp_sendspace</i>	This tunable specifies the socket buffer size for sending data. The optimum buffer size is the product of the media bandwidth and the average round-trip time of a packet: $\text{optimum\_window} = \text{bandwidth} * \text{average\_round\_trip\_time}$	This parameter is accessed when a new association is created. Use the <i>setsockopt</i> function to override this parameter.	The default value is 65536.
<i>sctp_recvspace</i>	This tunable specifies the socket buffer size for receiving data.	This parameter is accessed when a new association is created. Use the <i>setsockopt</i> function to override this parameter.	The default value is 65536.
<i>sctp_send_fewsacks</i>	When enabled, this tunable parameter implements <i>recv side silly window avoidance</i>. It prevents sending a window update until a receiver can fit in 1 MTU of data.	This parameter is accessed each time data is read by an application and a window update is being sent.	The default value is 0.
<i>sctp_cookie_life</i>	This tunable specifies the time duration in seconds for which a cookie is considered to be valid.	This parameter is used to determine a stale cookie during connection establishment.	The default value is 60 seconds.
<i>sctp_ecn</i>	This tunable enables or disables the explicit congestion notification (RFC 3168).	It is accessed during connection establishment.	The default value is 1.

Item	Description		
<i>sctp_ephemeral_high</i>	This tunable specifies the largest port number to allocate for the SCTP (Stream Control Transmission Protocol) ephemeral ports.	It is used when an application is trying to bind to a port.	The default value is 65535.
<i>sctp_ephemeral_low</i>	This tunable specifies the lowest port number to allocate for the SCTP ephemeral ports.	It is used when an application is trying to bind to a port.	The default value is 32768.
<i>sctp_instreams</i>	This tunable specifies the default number of inbound streams that an association uses.	It is used during connection establishment.	The default is 2048.
<i>sctp_outstreams</i>	This tunable specifies the default number of outbound streams that an association uses.	It is used during connection establishment.	The default value is 10.
<i>sctp_pmtu_discover</i>	If enabled, sets the <i>Don't Fragment</i> bit in an IP header of an outgoing packet.	It is accessed when the sending packets are sent out.	The default value is 1.
<i>sctp_rcv_multibuf</i>	This tunable controls the socket receive buffer accounting. The default value is 0 and it indicates that all the associations belonging to the socket share the same receive buffer space. When set to nonzero, each association has its own receive buffer space of this value. The <i>setsockopt</i> function overrides this value.	It is accessed when an association is being created.	The default value is 0 (<i>multibuf</i> is not used).
<i>sctp_send_multibuf</i>	This tunable controls the socket send buffer accounting. The default value is 0 and indicates that all the associations belonging to a socket share the same send buffer space. When set to nonzero, each association has its own send buffer space of this value. The <i>setsockopt</i> function overrides this value.	It is accessed when an association is being created.	The default is 0 (<i>multibuf</i> is not used).
<i>sctp_failover_type</i>	When enabled, it causes a new path to be chosen after every retransmit timeout. Otherwise, failover happens only after the <i>path error count</i> value exceeds <i>max path error count</i> value.	It is accessed whenever RTO starts (when there is a packet drop).	The default value is 1.

Item	Description		
<i>sctp_check_associd</i>	Governs the pattern related to checking the association ID passed by an application when sending an <i>ABORT</i> packet. If set to 0, it ignores the association ID. The association is found by using the foreign address. If set to 1, it performs strict association ID matching. If an association is not found with the passed <i>assoc_id</i> value, an <i>EINVAL</i> error is returned. If set to 2, it performs association ID matching, but uses the foreign address when a reserved <i>assoc_id</i> value is used.	It is accessed whenever a user application issues an ABORT packet.	The default value is 0.

Examples

1. To load the SCTP kernel extension, type the following:

```
sctpctrl load
```

2. To reset the SCTP statistics, type the following:

```
sctpctrl stats reset
```

This command will zero-out all the SCTP statistics.

3. To get the values of the SCTP tunable, type the following:

```
sctpctrl get
```

This will list all the SCTP tunable and their values. Here is a sample output.

```
sctp_assoc_maxerr = 10
sctp_cookie_life = 60
sctp_delack_timer = 4
sctp_dontdelayack = 1
sctp_ecn = 1
sctp_ephemeral_high = 65535
sctp_ephemeral_low = 32768
sctp_instreams = 2048
sctp_maxburst = 8
sctp_outstreams = 10
sctp_path_maxerr = 5
sctp_pmtu_discover = 1
sctp_rttmax = 60
sctp_rttmin = 1
sctp_recvspace = 65536
sctp_sendspace = 65536
sctp_send_fewsacks = 0
```

4. To set **sctp_path_maxerr** to a value of 6, type the following:

```
sctpctrl set sctp_path_maxerr=6
```

Location

/usr/sbin/sctpctrl

Files

Item	Description
<code>/usr/sbin/sctpctrl</code>	Contains the sctpctrl command.
<code>/usr/lib/drivers/sctp</code>	Contains the SCTP kernel extension.

sdiff Command

Purpose

Compares two files and displays the differences in a side-by-side format.

Syntax

sdiff [**-l** | **-s**] [**-o** *OutFile*] [**-w** *Number*] *File1 File2*

Description

The **sdiff** command reads the files specified by the *File1* and *File2* parameters, uses the **diff** command to compare them, and writes the results to standard output in a side-by-side format. The **sdiff** command displays each line of the two files with a series of spaces between them if the lines are identical. It displays a < (less than sign) in the field of spaces if the line only exists in the file specified by the *File1* parameter, a > (greater than sign) if the line only exists in the file specified by the *File2* parameter, and a | (vertical bar) for lines that are different.

When you specify the **-o** flag, the **sdiff** command merges the files specified by the *File1* and *File2* parameters and produces a third file.

Note: The **sdiff** command invokes the **diff -b** command to compare two input files. The **-b** flag causes the **diff** command to ignore trailing spaces and tab characters and to consider other strings of spaces as equal.

Flags

Item	Description
-l	Displays only the left side when lines are identical.

Item	Description
-o OutFile	Creates a third file, specified by the <i>OutFile</i> variable, by a controlled line-by-line merging of the two files specified by the <i>File1</i> and the <i>File2</i> parameters. The following subcommands govern the creation of this file: e Starts the ed command with an empty file. e b o r e Starts the ed command with both sides. e l o r e < Starts the ed command with the left side. e r o r e > Starts the ed command with the right side. l Adds the left side to the output file. r Adds the right side to the output file. s Stops displaying identical lines. v Begins displaying identical lines. q Performs one of the following functions: - Exits the ed command. - Exits the sdiff command if no ed command is running. - Exits both commands. This action occurs when there are no more lines to be merged into the output file. Each time you exit from the ed command, the sdiff command writes the resulting edited file to the end of the file specified by the <i>OutFile</i> variable. If you do not save the changes before exiting (for example, you press the Ctrl-C key sequence), the sdiff command writes the initial input to the output file.
-s	Does not display identical lines.
-w Number	Sets the width of the output line. The default value of the <i>Number</i> variable is 130 characters. The maximum width of the <i>Number</i> variable is 2048. The minimum width of the <i>Number</i> variable is 20. The sdiff command uses 2048 if a value greater than 2048 is specified.

Exit Status

The **sdiff** command returns the following exit values:

Table 17. Exit status	
Item	Description
1	Successful completion.
2	An error occurred.

Examples

1. To print a comparison of two files, enter:

```
sdiff chap1.bak chap1
```

The **sdiff** command displays a side-by-side listing that compares each line of the chap1.bak and chap1 files.

2. To display only the lines that differ, enter:

```
sdiff -s -w 80 chap1.bak chap1
```

The **sdiff** command displays the differences at the workstation. The -w 80 flag and variable sets the page width to 80 columns. The -s flag indicates lines that are identical in both files will not be displayed.

3. To selectively combine parts of two files, enter:

```
sdiff -s -w 80 -o chap1.combo chap1.bak chap1
```

The **sdiff** command combines the chap1.bak and chap1 files into a new file called chap1.combo. For each group of differing lines, the **sdiff** command prompts you which group to keep or whether you want to edit them using the **ed** command.

4. To combine and edit two files, staff.jan and staff.apr, and write the results to the staff.year file, perform the steps indicated.

The staff.jan file contains the following lines:

```
Members of the Accounting Department
Andrea
George
Karen
Sam
Thomas
```

The staff.apr file contains the following lines:

```
Members of the Accounting Department
Andrea
Fred
Mark
Sam
Wendy
```

- a. Enter the following command:

```
sdiff -o staff.year staff.jan staff.apr
```

The **sdiff** command will begin to compare the contents of the staff.jan and staff.apr files and write the results to the staff.year file. The **sdiff** command displays the following:

```
Members of the Accounting Dept  Members of the Accounting Dept
Andrea                          Andrea
George                          | Fred
%
```

The % (percent sign) is the command prompt.

- b. Enter the **eb** subcommand to start editing the output file with the **ed** command.

The **sdiff** command displays a sequence of digits, indicating the byte count of lines being merged. In this case, the byte count is 23.

- c. Enter the **q** subcommand to exit the **ed** command and continue combining and editing the two files. The **sdiff** command displays the following:

```
Sam                               Sam
Thomas                           | Wendy
```

- d. Enter the **eb** subcommand again. The **ed** command must be run each time a set of lines from the original two files are to be merged into the output file. The byte count in this instance is 13.

- e. Enter the **q** subcommand to save the changes. When all the lines of the two files have been merged into the output file, the **q** subcommand exits the **ed** and **sdiff** commands.

The `staff.year` file now contains the following:

```
Members of the Accounting Department
Andrea
George
Karen
Fred
Mark
Sam
Thomas
Wendy
```

Files

Item	Description
<code>/usr/bin/sdiff</code>	Contains the sdiff command.

secldapclntd Daemon

Purpose

Provides and manages connection and handles transactions between the LDAP load module and the LDAP Security Information Server.

Syntax

`/usr/sbin/secldapclntd [ -C CacheSize ] [ -p NumOfThread ] [ -T HeartBeatIntv ] [ -o ldapTimeOut ]`

Description

The **secldapclntd** daemon accepts requests from the LDAP load module, forwards the request to the LDAP Security Information Server, and passes the result from the server back to the LDAP load module. This daemon reads the configuration information defined in the `/etc/security/ldap/ldap.cfg` file during its startup, authenticates to the LDAP Security Information Server using the specified server distinguished name and password, and establishes a connection between the local host and the server.

If multiple servers are specified in the `/etc/security/ldap/ldap.cfg` file, the **secldapclntd** daemon connects to all of the servers. At a specific time, however, it talks to only one of them. The priority of the server connection is determined by its location in the server list with the highest priority server listed first. The **secldapclntd** daemon can detect when the server it is currently communicating with is down, and automatically switches to another available server. It can also detect when a server becomes available again and re-establish connection to that server. If the reconnected server is of higher priority than the current server then communication is switched to it. This auto-detect feature is done by the **secldapclntd** daemon checking on each of the servers periodically. The time interval between subsequent checking is defaulted to 300 seconds, and can be changed at the daemon startup time from the command line with the **-T** option or by modifying the **heartbeatinterval** value in the `/etc/security/ldap/ldap.cfg` file.

At startup, the **secldapclntd** daemon tries to establish a connection to the LDAP servers. If it cannot connect to any of the servers, it goes to sleep, and tries again in 30 seconds. It repeats this process twice, and if it still cannot establish any connection, the **secldapclntd** daemon process exits.

The **secldapclntd** daemon is a multi-threaded program. The default number of threads used by this daemon is 10. An administrator can fine-tune the system performance by adjusting the number of threads used by this daemon.

The **secldapclntd** daemon caches information retrieved from the LDAP Security Information Server for performance purpose. If the requested data can be found in the cache and the cache entry is not expired,

the data in the cache is handed back to the requester. Otherwise, the **secdapclntd** daemon makes a request to the LDAP Security Information Server for the information.

The valid number of cache entries for users is in the range of 100-10,000, and that for groups is in the range of 10-1,000. The default is 1000 entries for users, and 100 entries for groups.

The cache timeout or TTL (time to live) can be from 60 seconds to 1 hour (60*60=3600 seconds). By default, a cache entry expires in 300 seconds. If the cache timeout is set to 0, the caching feature is disabled.

Communication between the **secdapclntd** daemon and the LDAP server is performed using asynchronous methods. This allows the daemon to request information from the server and then perform other steps while waiting for the request to return. The length of time that the client will wait for a response from a server is configurable by the administrator and defaults to 60 seconds.

When connecting to LDAP servers, the **secdapclntd** daemon needs to do host lookups. The **nis_ldap** resolver may cause the lookup to be routed back to the daemon itself, resulting in a hang situation. To avoid this problem, the **secdapclntd** daemon ignores the system order of name resolution. Instead, it uses the order defined by the **nsorder** attribute in the **/etc/security/ldap/ldap.cfg** file.

Flags

Note: By default, the **secdapclntd** daemon reads the configuration information specified in the **/etc/security/ldap/ldap.cfg** file at startup. If the following options are given on the command line when starting the **secdapclntd** process, the options from the command line will override the values in the **/etc/security/ldap/ldap.cfg** file.

Flag	Description
-C <i>CacheSize</i>	Sets the maximum cache entries used by the secdapclntd daemon to <i>CacheSize</i> number of entries. The valid range is 100-65536 entries for user cache entry. The default value is 1000. The valid range is 10-65536 for group cache entry. The default is value 100. If you set the user cache entry in the start-secdapclntd command, by using the -C option, the group cache entry is set to 10% of the user cache entry.
-o <i>ldapTimeOut</i>	Timeout period in seconds for LDAP client requests to the server. This value determines how long the client will wait for a response from the LDAP server. Valid range is 0 - 3600 (1 hour). Default is 60 seconds. Set this value to 0 to disable the timeout and force the client to wait indefinitely.
-p <i>NumOfThread</i>	Sets the number of threads used by the secdapclntd daemon to <i>NumOfThread</i> threads. Valid range is 1-256. The default is 10.
-t <i>CacheTimeout</i>	Sets the cache to expire in <i>CacheTimeout</i> seconds. Valid range is 60- 3600 seconds. The default is 300 seconds.
	Note: > Starting with AIX 7.3, the <i>CacheTimeout</i> attribute has been deprecated.<
-T <i>HeartBeatIntv</i>	Sets the time interval of heartbeat between this client and the LDAP server. Valid values are 60-3,600 seconds. Default is 300.

Examples

1. To start the **secdapclntd** daemon, type:

```
/usr/sbin/secdapclntd
```

2. To start the **secdapclntd** using 20 threads, type:

```
/usr/sbin/secdapclntd -p 20
```

Use of the **start-secdapclntd** command is recommended for starting the **secdapclntd** daemon. It is also recommended configuration values are specified in the **/etc/security/ldap/ldap.cfg** file instead of using command line flags, so that these values will be used each time you start the **secdapclntd** process.

secldifconv Command

Purpose

Converts user and group entries of an LDIF from one schema type to another.

Syntax

```
secldifconv [-R load_module] -S schematype -i inputFile [-r]
```

Description

The **secldifconv** command reads the ldif formatted input file specified by the **-i** option, converts the user and group data using the schema type specified by the **-S** option, and prints the result to stdout. If redirected to a file, the result can be added to an LDAP server with the **ldapadd** command or the **ldif2db** command.

The **-S** option specifies the conversion schema type used for the ldif output. The **secldifconv** command accepts the following schema types:

- **AIX** - AIX schema (aixaccount and aixaccessgroup objectclasses)
- **RFC2307** - RFC 2307 schema (posixaccount, shadowaccount, and posixgroup objectclasses)
- **RFC2307AIX** - RFC 2307 schema with full AIX support (posixaccount, shadowaccount, and posixgroup objectclasses, plus the aixauxaccount and aixauxgroup objectclasses).

The input file specified with the **-i** option can include entries in any of the above supported schemas. The **secldifconv** command will convert user and group entries according to the attribute mapping defined in the **/etc/security/ldap/*.map** files for the corresponding schema type. Only user and group entries will be converted, other entries are output unaltered.

Use of the **-r** option allows the removal of attributes in user and group entries that are not included in the specified output schema. If the option is not specified then unrecognized attributes are assumed to be valid and are output unaltered. Note that if the user or group attribute is defined in the schema **secldifconv** is converting from but not in the schema requested to convert into, then the attribute will not be output. This behavior allows for conversion between the **AIX** and **RFC2307AIX** schemas to the **RFC2307** schema which contains a subset of attributes.

If the **db2ldif** command is used to generate the input file for **secldifconv**, passwords without an encryption prefix are output in {IMASK} format. In order to convert the {imask} format into the proper {crypt} format, the **-R** option should be used to specify the Loadable I&A module to read the password from for conversions from **AIX** schema type, assuming the system has been previously configured to be an LDAP client.

Care should be taken when adding users and groups from other systems to the LDAP server using the **secldifconv** command output. The **ldapadd** and **ldif2db** commands check only for entry name (user name or group name) but not for the numeric ID when adding entries. Merging users and groups from multiple servers using **secldifconv** output can result in sharing of a numeric ID by multiple accounts, which is a security violation. Note that IBM Directory Server 5.2 and later supports a unique attribute feature that can be used to avoid this issue.

Flags

Item	Description
-R <i>load_module</i>	Specifies the loadable I&A module used to retrieve the user's password if necessary.
-S <i>schematype</i>	Specifies the output LDAP schema type. Valid values are AIX , RFC2307 , and RFC2307AIX .
-i <i>inputFile</i>	Specifies the input file in ldif format that contains user and group data to convert.
-r	Specifies to remove any attributes that are not defined in the specified schema type.

Exit Status

This command returns the following exit values:

Item	Description
0	The command completed successfully.
>0	An error occurred.
-1	Memory failure (that is, Memory allocation failure).

Examples

1. To convert entries in a ldif formatted file to the `rfc2307` schema, type the following:

```
secldifconv -S rfc2307 -i input.ldif
```

This displays the converted file to stdout in ldif format. User entries and group entries are converted into the **rfc2307** schema type.

2. To convert entries in a ldif formatted file to the `rfc2307aix` schema and remove unrecognized attributes, type the following:

```
secldifconv -R LDAP -S rfc2307aix -i input.ldif -r > convert.ldif
```

This sends the output of the command to the `convert.ldif` file in ldif format. Unrecognized attributes are removed during conversion and user passwords will be requested from the LDAP module if necessary.

Location

/usr/sbin/secldifconv

Files

Mode	File
r	/etc/security/ldap/2307aixgroup.map
r	/etc/security/ldap/2307aixuser.map
r	/etc/security/ldap/2307group.map
r	/etc/security/ldap/2307user.map
r	/etc/security/ldap/aixgroup.map
r	/etc/security/ldap/aixuser.map

sectoldif Command

Purpose

Prints users and groups defined locally to **stdout** in ldif format.

Syntax

`sectoldif -d baseDN [ -S schematype ] [ -u username ]`

Description

The **sectoldif** command reads users and groups defined locally, and prints the result to **stdout** in ldif format. If redirected to a file, the result can be added to a LDAP server with the **ldapadd** command or the **ldif2db** command.

The **-S** option specifies the schema type used for the ldif output. The **sectoldif** command accepts three schema types:

- **AIX** - AIX schema (**aixaccount** and **aixaccessgroup** objectclasses)
- **RFC2307** - RFC 2307 schema (**posixaccount**, **shadowaccount**, and **posixgroup** objectclasses)
- **RFC2307AIX** - RFC 2307 schema with full AIX support (**posixaccount**, **shadowaccount**, and **posixgroup** objectclasses, plus the **aixauxaccount** and **aixauxgroup** objectclasses).

The **sectoldif** command is called by the **mksecldap** command to export users and groups during LDAP server setup. One needs to be extra cautious when exporting additional users and groups from other systems to the LDAP server using the **sectoldif** output. The **ldapadd** and **ldif2db** commands check only for entry name (user name or group name) but not for the numeric id when adding entries. Exporting users and groups from multiple systems using **sectoldif** output can result in sharing of a numeric id by multiple accounts, which is a security violation.

The **sectoldif** command reads the `/etc/security/ldap/sectoldif.cfg` file to determine what to name the user, group and system sub-trees that the data will be exported to. The **sectoldif** command only exports data to the USER, GROUP and SYSTEM types. The names specified in the file will be used to create sub-trees under the base DN specified with the **-d** flag. Refer to the `/etc/security/ldap/sectoldif.cfg` file documentation for more information.

Flags

Item	Description
-d <i>baseDN</i>	Specifies the base DN under which to place the user and group data.
-S <i>schematype</i>	Specifies the LDAP schema used to represent user/group entries in the LDAP server. Valid values are AIX, RFC2307, and RFC2307AIX. Default is AIX.
-u <i>username</i>	Specifies to print a specific user.

Examples

1. To print all users and groups defined locally, enter the following:

```
sectoldif -d cn=aixsecdb,cn=aixdata -S rfc2307aix
```

This prints all users and groups defined locally to **stdout** in ldif format. User entries and group entries are represented using the rfc2307aix schema type. The base DN is set to cn=aixsecdb, cn=aixdata.

2. To print only locally defined user foo, enter the following:

```
sectoldif -d cn=aixsecdb,cn=aixdata -u foo
```

This prints locally defined user foo to **stdout** in ldif format. Without the **-S** option, the default AIX schema type is used to represent foo's ldif output.

Files

Mode	File
r	/etc/passwd
r	/etc/group
r	/etc/security/passwd
r	/etc/security/limits
r	/etc/security/user
r	/etc/security/envIRON
r	/etc/security/user.roles
r	/etc/security/lastlog
r	/etc/security/smitacl.user
r	/etc/security/mac_user
r	/etc/security/group
r	/etc/security/smitacl.group
r	/etc/security/login.cfg

securetcpip Command

Purpose

Enables the operating system network security feature.

Syntax

securetcpip

Description

The **securetcpip** command provides enhanced security for the network. This command performs the following:

1. Runs the **tcback -a** command, which disables the nontrusted commands and daemons: **rcp**, **rlogin**, **rlogind**, **rsh**, **rshd**, **tftp**, and **tftpd**. The disabled commands and daemons are not deleted; instead, they are changed to mode 0000. You can enable a particular command or daemon by re-establishing a valid mode.
2. Adds a TCP/IP security stanza to the **/etc/security/config** file. The stanza is in the following format:

```
tcpip:
  netrc = ftp,rexec      /* functions disabling netrc */
```

Before running the **securetcpip** command, acquiesce the system by logging in as root user and executing the **killall** command to stop all network daemons.

Attention: The **killall** command kills all processes except the calling process. If logged in or applications are running, exit or finish before executing the **killall** command.

After issuing the **securetcip** command, shut down and restart your system. All of your TCP/IP commands and network interfaces should be properly configured after the system restarts.

Files

Item	Description
/etc/security/config	Contains information for the security system.
/etc/security/sysck.cfg	Contains file definitions for the trusted computing base.

sed Command

Purpose

A stream editor.

Syntax

sed [**-n**] [**-u**] *Script* [*File* ...]

sed [**-n**] [**-u**] [**-e** *Script*] ... [**-f** *ScriptFile*] ... [*File* ...]

Description

The **sed** command modifies lines from the specified *File* parameter according to an edit script and writes them to standard output. The **sed** command includes many features for selecting lines to be modified and making changes only to the selected lines.

The **sed** command uses two workspaces for holding the line being modified: the pattern space, where the selected line is held; and the hold space, where a line can be stored temporarily.

An edit script consists of individual subcommands, each one on a separate line. The general form of **sed** subcommands is the following:

[*address-range*] *function*[*modifiers*]

The **sed** command processes each input *File* parameter by reading an input line into a pattern space, applying all **sed** subcommands in sequence whose addresses select that line, and writing the pattern space to standard output. It then clears the pattern space and repeats this process for each line specified in the input *File* parameter. Some of the **sed** subcommands use a hold space to save all or part of the pattern space for subsequent retrieval.

When a command includes an address (either a line number or a search pattern), only the addressed line or lines are affected by the command. Otherwise, the command is applied to all lines.

An address is either a decimal line number, a \$ (dollar sign), which addresses the last line of input, or a context address. A context address is a regular expression similar to those used in the **ed** command except for the following differences:

- You can select the character delimiter for patterns. The general form of the expression is:

```
\?pattern?
```

where ? (question mark) is a selectable character delimiter. You can select any character from the current locale except for the space or new-line character. The \ (backslash) character is required only for the first occurrence of the ? (question mark).

The default form for the pattern is the following:

```
/pattern/
```

A \ (backslash) character is not necessary.

- The **\n** sequence matches a new-line character in the pattern space, except the terminating new-line character.
- A **.** (period) matches any character except a terminating new-line character. That is, unlike the **ed** command, which cannot match a new-line character in the middle of a line, the **sed** command can match a new-line character in the pattern space.

Certain commands called *addressed* commands allow you to specify one line or a range of lines to which the command should be applied. The following rules apply to addressed commands:

- A command line without an address selects every line.
- A command line with one address, expressed in context form, selects each line that matches the address.
- A command line with two addresses separated by commas selects the entire range from the first line that matches the first address through the next line that matches the second. (If the second address is a number less than or equal to the line number first selected, only one line is selected.) Thereafter, the process is repeated, looking again for the first address.

Flags

Item	Description
-e <i>Script</i>	Uses the <i>Script</i> variable as the editing script. If you are using just one -e flag and no -f flag, the -e flag can be omitted.
-f <i>ScriptFile</i>	Uses the <i>ScriptFile</i> variable as the source of the edit script. The <i>ScriptFile</i> variable is a prepared set of editing commands applied to the <i>File</i> parameter.
-n	Suppresses all information normally written to standard output.
-u	Displays the output in an unbuffered mode. When this flag is set, the sed command displays the output instantaneously instead of buffering the output. The default is buffered mode.

Note: You can specify multiple **-e** and **-f** flags. All subcommands are added to the script in the order specified, regardless of their origin.

sed Subcommands

The **sed** command contains the following **sed** script subcommands. The number in parentheses preceding a subcommand indicates the maximum number of permissible addresses for the subcommand.

Note:

1. The *Text* variable accompanying the **a**, **c**, and **i** subcommands can continue onto more than one line, provided all lines but the last end with a **** (backslash) to quote the new-line character. Backslashes in text are treated like backslashes in the replacement string of an **s** command and can be used to protect initial blanks and tabs against the stripping that is done on every script line. The *RFile* and *WFile* variables must end the command line and must be preceded by exactly one blank. Each *WFile* variable is created before processing begins.
2. The **sed** command can process up to 999 subcommands in a pattern file.

Item	Description
(1) a <i>Text</i>	Places the <i>Text</i> variable in output before reading the next input line.
(2) b [<i>label</i>]	Branches to the : command bearing the <i>label</i> variable. If the <i>label</i> variable is empty, it branches to the end of the script.
(2) c <i>Text</i>	Deletes the pattern space. With 0 or 1 address or at the end of a 2-address range, places the <i>Text</i> variable in output and then starts the next cycle.
(2) d	Deletes the pattern space and then starts the next cycle.

Item	Description
(2) D	Deletes the initial segment of the pattern space through the first new-line character and then starts the next cycle.
(2) g	Replaces the contents of the pattern space with the contents of the hold space.
(2) G	Appends the contents of the hold space to the pattern space.
(2) h	Replaces the contents of the hold space with the contents of the pattern space.
(2) H	Appends the contents of the pattern space to the hold space.
(1) i <i>Text</i>	Writes the <i>Text</i> variable to standard output before reading the next line into the pattern space.
(2) l	Writes the pattern space to standard output showing nondisplayable characters as 4-digit hexadecimal values. Long lines are folded.
(2) l	Writes the pattern space to standard output in a visually unambiguous form. The characters <code>\\</code> , <code>\\a</code> , <code>\\b</code> , <code>\\f</code> , <code>\\r</code> , <code>\\t</code> , and <code>\\v</code> are written as the corresponding escape sequence. Non-printable characters are written as 1 three-digit octal number (with a preceding backslash character) for each byte in the character (most significant byte first). This format is also used for multibyte characters. This subcommand folds long lines. A backslash followed by a new-line character indicates the point of folding. Folding occurs at the 72nd column position. A \$ (dollar sign) marks the end of each line.
(2) n	Writes the pattern space to standard output if the default output is not suppressed. It replaces the pattern space with the next line of input.
(2) N	Appends the next line of input to the pattern space with an embedded new-line character (the current line number changes). You can use this to search for patterns that are split onto two lines.
(2) p	Writes the pattern space to standard output.
(2) P	Writes the initial segment of the pattern space through the first new-line character to standard output.
(1) q	Branches to the end of the script. It does not start a new cycle.
(2) r <i>RFile</i>	Reads the contents of the <i>RFile</i> variable. It places contents in output before reading the next input line.

Item	Description
(2) s / <i>pattern</i> / <i>replacement</i> / <i>flags</i>	Substitutes the <i>replacement</i> string for the first occurrence of the <i>pattern</i> parameter in the pattern space. Any character that is displayed after the s subcommand can substitute for the / (slash) separator except for the space or new-line character. See the Pattern Matching section of the ed command. The value of the <i>flags</i> variable must be zero or more of: g Substitutes all non-overlapping instances of the <i>pattern</i> parameter rather than just the first one. n Substitutes for the <i>n</i>-th occurrence only of the <i>pattern</i> parameter. p Writes the pattern space to standard output if a replacement was made. w <i>WFile</i> Writes the pattern space to the <i>WFile</i> variable if a replacement was made. Appends the pattern space to the <i>WFile</i> variable. If the <i>WFile</i> variable was not already created by a previous write by this sed script, the sed command creates it.
(2) t <i>label</i>	Branches to the <i>:label</i> variable in the script file if any substitutions were made since the most recent reading of an input line execution of a t subcommand. If you do not specify the <i>label</i> variable, control transfers to the end of the script.
(2) w <i>WFile</i>	Appends the pattern space to the <i>WFile</i> variable.
(2) x	Exchanges the contents of the pattern space and the hold space.
(2) y / <i>pattern1</i> / <i>pattern2</i> /	Replaces all occurrences of characters in the <i>pattern1</i> variable with the corresponding <i>pattern2</i> characters. The number of characters in the <i>pattern1</i> and <i>pattern2</i> variables must be equal. The new-line character is represented by \n .
(2) ! <i>sed-cmd</i>	Applies the specified sed subcommand only to lines not selected by the address or addresses.
(0) : <i>label</i>	Marks a branch point to be referenced by the b and t subcommands. This label can be any sequence of eight or fewer bytes.
(1) =	Writes the current line number to standard output as a line.
(2){ <i>subcmd</i> }	Groups subcommands enclosed in {} (braces).
(0)	Ignores an empty command.
(0) #	The "#" and the remainder of the line are ignored (treated as a comment), with one exception. For the first line of a script file, if the character after the # is an n, the default output is suppressed. The rest of the line after the #n is ignored.

Exit Status

This command returns the following exit values:

Ite	Description
m	
0	Successful completion.

Item Description

>0 An error occurred.

Examples

1. To perform a global change, enter:

```
sed "s/happy/enchanted/g" chap1 >chap1.new
```

This command sequence replaces each occurrence of the word **happy** found in the file **chap1** with the word **enchanted**. It puts the edited version in a separate file named **chap1.new**. The **g** character at the end of the **s** subcommand tells the **sed** command to make as many substitutions as possible on each line. Without the **g** character, the **sed** command replaces only the first occurrence of the word **happy** on a line.

The **sed** command operates as a filter. It reads text from standard input or from the files named on the command line (**chap1** in this example), modifies this text, and writes it to standard output. Unlike most editors, it does not replace the original file. This makes the **sed** command a powerful command when used in pipelines.

2. To use the **sed** command as a filter in a pipeline, enter:

```
pr chap2 | sed "s/Page *[0-9]*$/(&)/" | enq
```

This command sequence encloses the page numbers in parentheses before printing the file **chap2**. The **pr** command puts a heading and page number at the top of each page, then the **sed** command puts the page numbers in parentheses, and the **enq** command prints the edited listing.

The **sed** command pattern **/Page *[0-9]*\$/** matches page numbers that appear at the end of a line. The **s** subcommand changes this to **(&)**, where the **&** stands for the page number that was matched.

3. To display selected lines of a file, enter:

```
sed -n "/food/p" chap3
```

The **sed -n** displays each line in the file **chap3** that contains the word **food**. Normally, the **sed** command copies every line to standard output after it is edited. The **-n** flag stops the **sed** command from doing this. You then use subcommands like **p** to write specific parts of the text. Without the **-n** flag, this example displays all the lines in the file **chap3**, and it shows each line containing **food** twice.

4. To perform complex editing, enter:

```
sed -f script.sed chap4 >chap4.new
```

This command sequence creates a **sed** script file when you want to do anything complex. You can then test and modify your script before using it. You can also reuse your script to edit other files. Create the script file with an interactive text editor.

5. A sample **sed** script file:

```
:join
/\\$/ {N
s/\\n//
b join
}
```

This **sed** script joins each line that ends with a **** (backslash) to the line that follows it. First, the pattern **/\\\$/** selects a line that ends with a **** for the group of commands enclosed in **{}** (braces). The **N** subcommand then appends the next line, embedding a new-line character. The **s/\\n//** deletes the **** and embedded new-line character. Finally, **b join** branches back to the label **:join** to

check for a \ at the end of the newly joined line. Without the branch, the **sed** command writes the joined line and reads the next one before checking for a second \.

Note: The **N** subcommand causes the **sed** command to stop immediately if there are no more lines of input (that is, if the **N** subcommand reads an end-of-file character). It does not copy the pattern space to standard output before stopping. This means that if the last line of the input ends with a \, it is not copied to the output.

6. To copy an existing file (*oldfile*) to a new file (*newfile*) and replace all occurrences of the *testpattern* text string with the contents of the \$REPL shell variable, enter:

```
cat oldfile | sed -e "s/testpattern/$REPL/g" > newfile
```

7. To replace all occurrences of A with a, B with b, C with c, and all occurrences of newlines with character Z in the input file, enter:

```
$ sed -f command.file input.file
```

where *command.file* is the script file and *input.file* is the input file.

```
$cat command.file  
y/ABC\n/abcZ/
```

Alternatively, the following command can also be executed for the same function:

```
sed "y/ABC\n/abcZ/" input.file
```

sedmgr Command

Purpose

Displays and sets Stack Execution Disable flag of the system or executable files.

Syntax

```
sedmgr [-m {off | all | select | setidfiles}] [-o {on | off}] [-c {system | request | exempt}  
{file_name | file_group}] [-d {file_name | directory_name}] [-h]
```

Description

The **sedmgr** command is the manager of the Stack Execution Disable (SED) facility. You can use the command to enable and control the level of stack execution done in the system. This command can also be used to set the various flags in an executable file, controlling the stack execution disable. Any changes to the system wide mode setting will take effect only after a system reboot.

The system wide setting can only be modified by the root user. Other set and reset options on individual executable files will be successful only if the user has write permissions to the file. The SED facility is available only in the AIX 64 bit kernel operating systems.

If invoked without any parameter, the **sedmgr** command will display the current setting in regards to the stack execution disable environment.

For more information, refer to the *Stack Execution Disable Protection* section in **Login control** in the Security.

Flags

Item

-c

Description

Sets or resets the "request" and "exempt" SED flags in the header of an executable file. Also, sets or resets the SED request and exempt checking flag in the headers of all the executable files in a *file_group*. This option requires write privilege to the file, or root privilege if *file_group* is specified. The possible values are as follows:

system

If the file has the system flag in the executable's header, the operating system decides the operation for the process based on the system-wide SED flags. When the file does not specify any flags, the operating system also decides the operation for the process based on the system wide SED flags.

exempt

Sets a flag in the executable's header that indicates that this file does stack/head based execution and as a result needs exemption from the SED mechanism. The SED request checking bit is turned off.

request

Sets a flag in the executable's header that indicates that this file does not do any stack/data area based execution and as a result is SED capable. The SED exempt checking bit is turned off.

You can specify a file group that represents a group of files, such as TCB files. If the specified file name string does not identify a file, then the string is assumed to identify a *file_group*. Currently only the TCB_files file group is defined. You can set or reset the SED request and exempt flags for both 32-bit and 64-bit executable. The -c flag cannot be used with the -m, -o, and -d flags.

-d

Displays the SED request and exempt checking flag for executable files. The SED request and exempt flags are in the file header of an executable. If a directory is specified, then all executable under that directory and its subdirectories are displayed with their SED related flags. This flag requires read privilege to the *file_name* or *directory_name*. The -d flag cannot be used with the -m, -o and -c flags.

-h

Displays the syntax of the sedmgx command.

Item

-m

Description

Sets the system-wide stack execution disable mode if the processor supports SED. Any changes to the system-wide setting require a system reboot to take effect. This option will accept one of the following values:

all

Enforces stack execution disable for all files except the ones requesting (marked for) exemption.

off

Turns off the stack execution disable functionality on the system.

select

Sets the mode of operation to select the set of processes that will be enabled and monitored for stack execution disable. Only processes from files with the "request" SED flag set in their headers will be selected.

setidfiles

Sets the mode of operation so that the operating system performs SED for the files with the "request" SED flag set and enables SED for the executable files with the following characteristics:

- **setuid** files owned by root.
- setid files with primary group as "system" or "security".

The configured SED attribute is effective at the next 64-bit kernel boot time. Because the SED attribute in ODM does not affect 32-bit kernels, the SED monitoring flag is turned off in that case. If a processor does not support SED, the `sedmgr` command returns an error with the -m flag. The -m flag cannot be used with the -c and -d flags.

Item	Description
-o	This option enables SED to monitor instead of terminating the processes when exceptions occur. This option allows you to evaluate if an executable is doing any legitimate stack execution. This setting works with the system-wide mode set using the -c option. The SED Monitoring Control flag is part of the system-wide SED settings stored in ODM. Changing this setting requires root privilege. The possible values for this flag are as follows: on Turns on the monitoring for SED facility. When operating in this mode, the system will allow the process to continue operating even if an SED related exception occurs. Instead of terminating the process, the operating system logs the exception in the AIX error log subsystem. off Turns off the monitoring mode for SED facility. In this mode, the operating system terminates any process that violates and raises an exception per SED facility. The configured SED attribute is effective at the next 64-bit kernel boot time. Because the SED attribute in ODM does not affect 32-bit kernels, the SED monitoring flag is turned off in that case. If a processor does not support SED, the <code>sedmgr</code> command returns an error with the -m flag. The -o flag cannot be used with the -c and -d flags.
None	If no flag is specified, the <code>sedmgr</code> command displays the current setting in regards to the stack execution disable environment. It displays the current SED setting in the kernel <code>var</code> structure and the system-wide SED settings in ODM.

Parameters

Item	Description
<i>file_name</i>	Name of the executable file whose SED settings are changed. Requires write privilege.
<i>file_group</i>	Group of executable files whose SED settings are changed when a file name is not specified. Requires root privilege.
<i>directory_name</i>	Directory of executable files and any subdirectories of executable files whose SED checking flags are displayed with the -d flag.

Exit Status

Item	Description
0	The command completed successfully.

Item	Description
255	An error occurred.

Security

Access Control: This command should be a standard user command and have the trusted computing base attribute.

Examples

1. To change the system-wide SED Mode flag to `setidfiles` and the SED Control flag to `on`, type:

```
sedmgr -m setidfiles -o on
```

2. To change the SED checking flag to exempt for the `plans` file, type:

```
sedmgr -c exempt plans
```

3. To change the SED checking flag to select for all the executable files marked as a TCB file, type:

```
sedmgr -c request TCB_files
```

4. To display the SED checking flag of the `plans` file, type:

```
sedmgr -d plans
```

Restrictions

Auditing Events: If the auditing subsystem has been properly configured and is enabled, the `sedmgr` command generates the following audit record (event):

Event	Information
SEDMGR_Odm	System wide SED setting.
SEDMGR_File	SED setting in an executable file header.

See Setting up auditing in the **Auditing overview** section of Security for more details about how to properly select and group audit events, and how to configure audit event data collection.

Location

`/usr/sbin/sedmgr`

Files

Item	Description
<code>/usr/bin/tcbck</code>	Accessed in executable mode.
<code>/usr/bin/ldedit</code>	Accessed in executable mode.

send Command

Purpose

Sends a message.

Syntax

send [*File* ... | { **-draft** | **-nodraftfolder** | **-draftfolder** *+Folder* | **-draftmessage** *Message* }] [**-alias** *File*] [**-format** | **-noformat**] [**-nomsgid** | **-msgid**] [**-nofilter** | **-filter** *File*] [**-nopush** | **-push**] [**-forward** | **-noforward**] [**-noverbose** | **-verbose**] [**-nowatch** | **-watch**]

Description

The **send** command routes messages through the mail delivery system. If the delivery fails, the **send** command displays an error message. By default, From: and Date: fields are added to each specified message. Unless a **\$SIGNATURE** environment variable or signature: profile entry exists, the **send** command places the sender's address in the From: field.

The **send** command puts the current date in the Date: field. If the **dist** command calls the **send** command, the **send** command adds Resent- to the From:, Date:, and Message-ID: fields.

After successful delivery, the **send** command removes messages from active status by renaming them. The system renames messages by prefacing the current message number with a , (comma). Inactive files are unavailable to the Message Handler (MH) package. However, system commands can still manipulate inactive files. Until you use the **send** command again, you can retrieve an inactive file.

Flags

Item	Description
-alias <i>File</i>	Specifies a mail alias file to be searched. Three MH profile entries are required to use MH aliases: ali: -alias Aliases send: -alias Aliases whom: -alias Aliases where Aliases is the file to be searched. The default alias file is /etc/mh/MailAliases .
-draft	Uses the current draft message if no file is specified. Without this flag and when no file is specified, the send command asks the user if the current draft message is the one to use.
-draftfolder <i>+Folder</i>	Specifies the draft folder that contains the draft message to be sent. The -draftfolder <i>+Folder</i> flag followed by a <i>Message</i> parameter is the same as specifying the -draftmessage flag.

Item	Description
-draftmessage <i>Message</i>	Specifies the message to be sent. You can use one of the following message references as the value of the <i>Message</i> parameter: Number Number of the message. cur or . (period) Current message. This is the default. first First message in a folder. last Last message in a folder. next Message following the current message. prev Message preceding the current message.
-filter <i>File</i>	Uses the format instructions in the specified file to reformat copies of the message sent to the recipients listed in the Bcc : field.
-format	Puts all recipient addresses in a standard format for the delivery transport system. This flag is the default.
-forward	Adds a failure message to the draft message and returns it to the sender if the send command fails to deliver the draft. This flag is the default.
-help	Lists the command syntax, available switches (toggles), and version information. Note: For MH, the name of this flag must be fully spelled out.
-msgid	Adds a message-identification component (such as Message-ID:) to the message.
-nodraftfolder	Undoes the last occurrence of the -draftfolder +Folder flag. This flag is the default.
-nofilter	Removes the Bcc : header field from the message for recipients listed in the To : and cc : fields. The flag then sends the message with minimal headers to recipients listed in the Bcc : field. This flag is the default.
-noformat	Prevents alteration of the format of the recipient addresses.
-noforward	Prevents return of the draft message to the sender if delivery fails.
-nomsgid	Prevents addition of a message-identification component. This flag is the default.
-nopush	Runs the send command in the foreground. This flag is the default.
-noverbose	Prevents display of information during the delivery of the message to the sendmail command. This flag is the default.
-nowatch	Prevents display information during delivery by the sendmail command. This flag is the default.
-push	Runs the send command in the background. The send command does not display error messages on the terminal if delivery fails. Use the -forward flag to return messages to you that are not delivered.

Item	Description
-verbose	Displays information during the delivery of the message to the sendmail command. This information allows you to monitor the steps involved in sending mail.
-watch	Displays information during the delivery of the message by the sendmail command. This information allows you to monitor the steps involved in sending mail.

Profile Entries

The following entries are entered in the *UserMhDirectory/.mh_profile* file:

Item	Description
Draft-Folder:	Sets the default folder for drafts.
mailproc:	Specifies the program used to post failure notices.
Path:	Specifies the user's MH directory.
postproc:	Specifies the program used to post messages.
Signature:	Sets the mail signature.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

To send a draft message that is in your **\$HOME/Mail/draft** file, enter:

```
send
```

The system responds with a message similar to the following:

```
Use "/home/david/Mail/draft"?
```

If you enter yes, the draft message is sent, and you are returned to the shell prompt. In this example, the name of the **\$HOME** directory is **/home/david**.

Files

Item	Description
\$HOME/.mh_profile	Specifies the MH user profile.
/usr/bin/send	Contains the send command.

sendbug Command

Purpose

Mails a system bug report to a specified address.

Syntax

sendbug [*Address*]

Description

The **sendbug** command is a shell script to assist the user in composing and mailing bug reports in the correct format.

The **sendbug** command starts the editor specified by the **EDITOR** environment variable on a temporary copy of the bug report format outline. The default editor is vi.

Fill out the appropriate fields in the bug report format outline and exit the editor. The **sendbug** command mails the completed report to the address specified by the *Address* parameter. The default address is POSTMASTER.

Files

Item	Description
/usr/lib/bugformat	Contains the bug report outline.

sendmail Command

Purpose

Routes mail for local or network delivery.

Syntax

```
sendmail [ -ba | -bd | -bD | -bh | -bH | -bi | -bm | -bp | -bs | -bv | -bt [ -Ac File ] [ -C File ] [ -D Log File ] [ -d Value ] [ -B Type ] [ -F FullName ] [ -f Name ] [ -G ] [ -h Number ] [ -i ] [ -M x Value ] [ -n ] [ -N Dsn ] [ -OOption=Value ] [ -o Option [ Value ] ] [ -p Protocol ] [ -q [ Time ] ] [ -qG name ] [ -qI Substr ] [ -qR Substr ] [ -qS Substr ] [ -R Return ] [ -r addr ] [ -t ] [ -V Envid ] [ -v ] [ -X LogFile ] Address
```

Note: The *Address* parameter is optional for **-bd**, **-bi**, **-bp**, **-bt**, and **-q** [*Time*] flags.

Description

Notes:

- > Starting with AIX 7.2, Technology Level 5, the **sendmail** command uses the **sendmail** command version 8.18.1. The **sendmail** command is run by a new smmsp user and smmsp group instead of the root user for enhanced security.<
- In the **sendmail** command v8.7, and later, the order for name resolution is Domain Name System (DNS), Network Information Services (NIS), Network Interface Services (NIS), and local. To override this default order, specify an order in the /etc/netsvc.conf file or specify the **NSORDER** environment variable.
- > In AIX 7.3, the **sendmail** application supports Simple Authentication and Security Layer (SASL) based authentication.<

The **sendmail** command receives formatted text messages and routes the messages to one or more users. On a network, the **sendmail** command translates the message header information format to match the requirements of the destination system. The **sendmail** program determines the network of the destination system by using the syntax and content of the addresses.

The **sendmail** command delivers messages to the following users:

- Users on the local system.
- Users connected to the local system by using the TCP/IP protocol.
- Users connected to the local system by using the Basic Networking Utilities (BNU) command protocol.

Use the **sendmail** command only to deliver only pre-formatted messages and is not a user interface routine. Other commands provide user-friendly interfaces. The **sendmail** command reads standard input for message text and sends a copy of the message to all addresses listed when it reads an end of the message character. The end of the message character is either an end-of-file (Ctrl-D) control sequence or a single period on a line.

For more information on the **sendmail** command, see the following subsections:

- [sendmail Mail Filter API \(Milter\)](#)
- [sendmail mail filter flags](#)
- [sendmail mail filter timeouts](#)
- [Using the sendmail configuration files](#)
- [Restarting and refreshing the sendmail processes](#)
- [Migrating to AIX 7.2, Technology Level 4](#)
- [Defining Aliases](#)

sendmail Mail Filter API (Milter)

The **sendmail** Mail Filter API provides access to mail messages as they are being processed. This allows the third-party programs to filter the meta-information and the content. Filters that are developed by using the **sendmail** Mail Filter API use threads. It is necessary to alter the per-process limits in your filter. For example, if your filter is frequently used, use the **setrlimit** subroutine to increase the number of open file descriptors.

Specifying filters in sendmail configs

Use the key letter **X** (for external) to specify the filters. The following examples are a few of the filters:

- `Xfilter1, S=local:/var/run/f1.sock, F=R`
- `Xfilter2, S=inet6:999@localhost, F=T, T=C:10m;S:1s;R:1s;E:5m`
- `Xfilter3, S=inet:3333@localhost`

You can specify filters in your `.mc` file. The following filter is attached to a UNIX-domain socket in the `/var/run` directory:

```
INPUT_MAIL_FILTER(`filter1', `S=local:/var/run/f1.sock, F=R')
```

The following filter uses an IPv6 socket on port 999 of the local host:

```
INPUT_MAIL_FILTER(`filter2', `S=inet6:999@localhost, F=T, T=C:10m;S:1s;R:1s;E:5m')
```

The following filter uses an IPv4 socket on port 3333 of the local host:

```
INPUT_MAIL_FILTER(`filter3', `S=inet:3333@localhost')
```

sendmail mail filter flags

R

Rejects connection if filter is not available.

T

Temporarily fails connection if filter is not available.

If both F=R or F=T is specified, the **sendmail** command passes the message as if the filter is not present. The separator is a comma (,).

sendmail mail filter timeouts

You can override the default sendmail timeouts with T=x, where x can take the following values:

C

Timeout for connecting to a filter (if 0, use system timeout).

S

Timeout for sending information from the Mail Transmission Agent (MTA) to a filter.

R

Timeout for reading reply from the filter.

E

Overall timeout between sending the end-of-message to the filter and waiting for the final acknowledgment.

The separator between each entry is a semicolon(;).

The default values are T=C:0m;S:10s;R:10s;E:5m.

The InputMailFilters option determines the filters that are started and how the filters are sequenced:

```
InputMailFilters=filter1, filter2, filter3
```

This is set automatically according to the order of the **INPUT_MAIL_FILTER** commands in your .mc file. You can also reset the value by setting confINPUT_MAIL_FILTERS in your .mc file. This option calls the three filters in the order that the filters were specified.

You can define a filter without adding it to the input filter list by using MAIL_FILTER() instead of INPUT_MAIL_FILTER() in your .mc file.

Note: If InputMailFilters is not defined, no filters are used.

Using the sendmail configuration files

In AIX 7.2, Technology Level 3, and earlier, the **sendmail** command uses a single configuration file, /etc/mail/sendmail.cf to set operational parameters and to determine how the command parses addresses. Starting with AIX 7.2, Technology Level 4, the **sendmail** command supports the Mail Submission Program mode (MSP_mode) by using the /etc/mail/submit.cf configuration file. The **sendmail** command in the MSP mode does not require root privileges. Therefore, the **sendmail** command in MSP mode is more secure as compared to the previous version. The **sendmail** command uses the sendmail.cf configuration file when the **sendmail** command runs as mail server daemon in the MTA mode. For more information about the security consideration of the **sendmail** command, see <http://www.sendmail.org/~ca/email/doc8.12/SECURITY>.

Note: After the operating system is upgraded to AIX 7.2, Technology Level 5 or later, the /etc/mail directory contains the new /etc/mail/sendmail.cf and /etc/mail/submit.cf files. The existing /etc/mail/sendmail.cf and /etc/mail/submit.cf files from the earlier versions of AIX are copied to the /lpp/save.config/etc/mail directory.

The sendmail configuration files are as described:

/etc/mail/sendmail.cf

This configuration file is used when the **sendmail** command runs as mail server daemon in the MTA mode. By default, the sendmail.cf file uses the mail queue in the /var/spool/mqueue directory. On system boot, the **sendmail** MTA daemon is started in the /etc/rc.tcpip directory by default. To manually start the **sendmail** MTA daemon, enter the following command:

```
# startsrc -s sendmail -a " -bd -q30m"
```

/etc/mail/submit.cf

This configuration file is used by the **sendmail** command to operate in the MSP mode. By default, the `submit.cf` file uses the system mail queue in the `/var/spool/clientmqueue` directory. The **sendmail** command operates in the MSP mode under the following scenarios:

- When the **sendmail** command is run at command line or is called by another mail facility (such as the **mail** command) to send email.
- When the **sendmail** command is started as client-queue runner. The **sendmail** client-queue runner identifies the undelivered messages in the `/var/spool/clientmqueue` directory and submits the messages to the **sendmail** MTA daemon for delivery. To run the **sendmail** command as a queue runner in MSP mode manually, enter the following command:

```
# /usr/lib/sendmail -Ac -q 30m
```

You can also set the **sendmail** MTA daemon to start automatically whenever the system boots by editing the `/etc/rc.tcpip` file. For instructions about editing the `/etc/rc.tcpip` file, see [Starting the sendmail daemon during system boot](#).

Restarting and refreshing the sendmail processes

The configuration files that are used by the **sendmail** command are text files that you can edit by using any text editor. After modifying any of these configuration files, you must restart or refresh the MTA daemon and the MSP for the changes to take effect.

The current process ID of the **sendmail** command is stored in the `/etc/mail/sendmail.pid` file. To allow the **sendmail** command reread the newly edited configuration files, enter the following **kill** command:

```
#kill -15 `head -1 /etc/mail/sendmail.pid`
```

If the **srcmstr** command is running, you can run the **refresh** command to build the configuration database, the aliases database, and the NLS database again:

```
#refresh -s sendmail
```

If you started the **sendmail** MSP manually, and if the **sendmail** process is not controlled by the **srcmstr** command, you can stop the **sendmail** process by using the following **kill** command:

```
# kill <pid of the sendmail: Queue runner >
```

Migrating to AIX 7.2, Technology Level 4

If you are running AIX 7.2, Technology Level 3, or earlier, and if you configure the **sendmail** command, when you migrate to AIX 7.2, Technology Level 4, the **sendmail** command runs as an MTA daemon. To back up the `sendmail.cf` configuration file before starting the migration operation, enter the following command:

```
# cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.old
```

After the migration operation is complete, the previous `sendmail.cf` configuration file is transferred to the new `sendmail.cf` configuration file.

Complete the following steps after you migrate your AIX operating system to AIX 7.2, Technology Level 4, or later:

1. To restore the `sendmail.cf` backup file when you cannot find the `sendmail.cf` file after migration, enter the following command:

```
# cp /etc/mail/sendmail.cf.old /etc/mail/sendmail.cf
```

2. To restart and refresh the sendmail processes, enter the following commands:

```
# startsrc -s sendmail -a " -bd -q30m"
```

```
# refresh -s sendmail
```

These commands run the **sendmail** command as MTA.

The **sendmail** command rereads the databases and continues the operation with the `sendmail.cf` configuration file.

Defining Aliases

The **sendmail** command allows you to define aliases to use when the **sendmail** command handles the local mail. Aliases are alternative names that you can use in place of elaborate network addresses or to build distribution lists.

Define aliases in the `/etc/mail/aliases` file. This file is a text file that you can edit. The **sendmail** command uses a database version of this file. Before any changes made to the `/etc/mail/aliases` file become effective, you must build a new alias database by using the `sendmail -bi` command or the **newaliases** command.

Berkeley database (DB) support is available on AIX for Sendmail 8.11.0. The **sendmail** command continues to read the aliases in the Database Manager (DBM) format until the aliases database gets rebuilt. Once rebuilt, **sendmail** command reads the aliases in the Berkeley DB format and store them in the `/etc/mail/aliases.db` file.

Note: When defining aliases in the `/etc/mail/aliases` file, use only lowercase characters for nested aliases. Uppercase characters on the right side of an alias are converted to lowercase before storing in the aliases database. In the following example, mail that is sent to `testalias` fails as `TEST` is converted to `test` when the second line is stored.

```
TEST: user@machine
testalias: TEST
```

Every system must have a user or user alias that is designated as the alias. The default `postmaster` alias is a root file. You can assign this alias to a different user in the `/etc/mail/aliases` file. The `postmaster` alias allows other users outside your system to send mail to a known ID and to get information about mailing to users on your system. Also, users on your system can send problem notifications to the `postmaster` ID.

The **sendmail** command first opens a database in the format of hash-style aliases file. If it fails or if the NEWDB support was not compiled, the command opens a new Database Manager (NDBM) database. If that fails, the **sendmail** command reads the aliases source file into its internal symbol table.

Flags

Table 18. Flags

Item	Description
-Ac <i>File</i>	Specifies the sendmail command to choose an alternative configuration file based on the operative mode. If you specify -bm , -bs , or -t flags, the sendmail command uses the <code>submit.cf</code> configuration file. If any other flags are specified, and for compatibility with earlier versions, the sendmail command uses the <code>sendmail.cf</code> configuration file. If you do not specify the <i>File</i> variable, by default, the sendmail command uses the <code>submit.cf</code> configuration file.
-B <i>Type</i>	Sets the body type to <i>Type</i> . Current legal values are <code>7BI</code> or <code>8BITMIME</code> . Note: The -b flag is mutually exclusive.

Table 18. Flags (continued)

Item	Description
-ba	Starts the sendmail command in an ARPANET mode. All input lines to the command must end with a carriage return and a line feed (CR-LF). The sendmail command generates messages with a CR-LF at the end and looks at the From: and Sender: fields to find the name of the sender. Starting with the sendmail command version 8.18.1, the behavior is updated to expect a standard end-of-data sequence (<CR><LF> . <CR><LF>) instead of a nonstandard sequence (<LF> . <CR><LF>) and (<LF> . <LF>).
-bd	Starts the sendmail command as a daemon running in the background as a Simple Mail Transfer Protocol (SMTP) mail router.
-bD	Starts the sendmail command as a daemon running in the foreground as an SMTP mail router.
-bh	Prints the persistent host status database.
-bH	Purges the persistent host status database.
-bi	Builds the alias database from information that is defined in the /etc/mail/aliases file. Running the sendmail command with this flag is the same as running the newaliases command.
-bm	Delivers mail in the usual way by default.
-bp	Prints a listing of the mail queue. Running the sendmail command with this flag is the same as running the mailq command.
-bs	Uses the SMTP as described in RFC821 to collect mail from the standard input. This flag includes the operations of the -ba flag that are compatible with SMTP.
-bt	Starts the sendmail command in address test mode. This mode allows you to enter interactive addresses and watch as the sendmail command displays the steps it takes to parse the address. At the test-mode prompt, enter a rule set or multiple rule sets separated by commas and an address. Use this mode for debugging the address parsing rules in a new configuration file.
-bv	Starts the sendmail command with a request to verify the user IDs provided in the <i>Address</i> parameter field of the command. The sendmail command responds with a message telling which IDs can be resolved to a mailer command. It does not try to collect or deliver a message. Use this mode to validate the format of user IDs, aliases, or mailing lists.
-C File	Starts the sendmail command by using an alternative configuration file that is specified by the <i>File</i> variable. Use this flag together with -bt to test a new configuration file before installing it as the running configuration file.
-D Log File	Sends the debugging output to the specified log file. The -D option must be before the -d option.
-d Value	Sets the debugging value to the value specified by the <i>Value</i> variable. The only valid value is 21 . <i>n</i> , where <i>n</i> is any nonzero integer. This produces information regarding address parsing and is used with the -bt flag. Higher values of <i>n</i> produce more verbose information. Root permissions are required for this flag.
-F FullName	Sets the full name of the sender to the string provided in the <i>FullName</i> variable.

Table 18. Flags (continued)

Item	Description
-f <i>Name</i>	Sets the name of the from person (the envelope sender of the mail). This address might also be used in the From: header if that header is missing during initial submission. The envelope sender address is used as the recipient for delivery status notifications and might also appear in a Return-path: header. This flag must be used only by trusted users (normally root, daemon, and uucp) or if the person you are trying to become is the same as the person you are. Otherwise, a X-Authentication-Warning header is added to the message.
-G	Relay (gateway) submission of a message. For example, when the rmail command calls the sendmail command.
-h <i>Number</i>	Sets the hop count to the value specified by the <i>Number</i> variable. The hop count is the number of times that the message is processed by an SMTP router (not just the local copy of the sendmail command). The mail router increments the hop count every time that the message is processed. When it reaches a limit, the message is returned with an error message in order to prevent infinite loops in the mail system.
-i	Ignores dots alone on lines by themselves in incoming messages. This option must be set if you are reading data from a file.
-L	Sets the identifier that is used in syslog messages to the supplied tag.
-M <i>x Value</i>	Sets marco <i>x</i> to the specified <i>Value</i> .
-N <i>Dsn</i>	Set delivery status notification conditions to Data Source Name (DSN). The following conditions are the delivery status notification conditions: • never: For no notifications or for a comma-separated list of the values. • failure: For notification if delivery failed. • delay: For notification if delivery is delayed. • success: For notification when the message is successfully delivered.
-n	Prevents the sendmail command from interpreting aliases.
-O <i>Option=Value</i>	Sets the <i>Option</i> to the specified <i>Value</i> . Use for long-form option names.
-o <i>Option [Value]</i>	Sets the <i>Option</i> variable. If the option is a valued option, you must specify a value for the <i>Value</i> variable. Note: For valid values, see Options for the sendmail Command in the <code>sendmail.cf</code> file in <i>Performance Tools Guide and Reference</i> .
-p <i>Protocol</i>	Sets the sending protocol. It is recommended that you set this option. You can set <i>Protocol</i> in the form <i>Protocol:Host</i> to set both the sending protocol and the sending host. For example, -pUUCP:uunet sets the sending protocol to UUCP and the sending host to uunet. Some existing programs use -oM flag to set the r and s macros, which is equivalent to the -p flag.
-qI <i>Substr</i>	Limits process jobs to those containing <i>Substr</i> as a substring of the queue ID.
-qG <i>name</i>	Processes jobs in a queue group are called by name only.
-qR <i>Substr</i>	Limits process jobs to those containing <i>Substr</i> as a substring of one of the recipients.
-qS <i>Substr</i>	Limits process jobs to those containing <i>Substr</i> as a substring of the sender.
-q [<i>Time</i>]	Processes saved messages in the queue at the intervals that are specified by the <i>Time</i> variable. If the <i>Time</i> variable is not specified, this flag processes the queue at once.

Table 18. Flags (continued)

Item	Description
-R Return	Sets the amount of the message to be returned if the message bounces. The <i>Return</i> parameter can be <code>full</code> to return the entire message or <code>hdrs</code> to return only the headers.
-r addr	An obsolete form of -f .
-t	Sends the message to the recipients specified in the <code>To :</code> , <code>Cc :</code> , and <code>Bcc :</code> fields of the message header, as well as to any users specified on the command line.
-V Envid	Sets the original envelope ID. This is propagated across SMTP to servers that support DSNs and is returned in DSN-compliant error messages.
-v	Starts the sendmail command in the verbose mode. The sendmail command displays the messages regarding the status of transmission and the expansion of aliases.
-X LogFile	Logs all traffic in and out of the sendmail command in a <i>LogFile</i> for debugging mailer problems. Use this flag sparingly as it produces a lot of data rapidly.

You can also set or remove the **sendmail** configuration processing options and is used by the user responsible for the mail system. To set these options, use the **-o** flag on the command line or the **O** control line in the `/etc/mail/sendmail.cf` configuration file.

Exit Status

The **sendmail** command returns exit status values. These exit values are defined in the `/usr/include/sysexits.h` file. The following table summarizes the meanings of these return values:

Table 19. Exit Values

Item	Description
EX_CANTCREAT	The sendmail command cannot create a file that the user specified.
EX_CONFIG	An error is found in the format of the configuration file.
EX_DATAERR	The input data is incorrect.
EX_IOERR	An error occurred during I/O.
EX_NOHOST	The sendmail command does not recognize the specified hostname.
EX_NOINPUT	An input file (not a system file) did not exist or is not readable.
EX_NOPERM	The user does not have permission to perform the requested operation.
EX_NOUSER	The sendmail command does not recognize a specified user ID.
EX_OK	The sendmail command successfully completed.
EX_OSERR	A temporary operating system error occurred. An example of such an error is a failure to create a new process.
EX_OSFILE	A system file error occurred. For example, a system file such as <code>/etc/passwd</code> does not exist, cannot be opened, or has another type of error that prevents it from being used.
EX_PROTOCOL	The remote system returns something that is incorrect during a protocol exchange.
EX_SOFTWARE	An internal software error occurred (including bad arguments).
EX_TEMPFAIL	The sendmail command cannot create a connection to a remote system. Try the request again later.
EX_UNAVAILABLE	A required service or resource by the sendmail command is not available.

Table 19. Exit Values (continued)

Item	Description
EX_USAGE	The command syntax is not correct.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Auditing Events

Table 20. Auditing Events

Event	Information
SENDMAIL_Config	Configuration event
SENDMAIL_ToFile	File-creation event

Example

To display the sendmail version, enter the following command:

```
echo \${Z} | sendmail -d0
```

A message similar to the following message is displayed:

```
Version AIX5.2/8.11.6p2
Compiled with: LDAPMAP MAP_REGEX LOG MATCHGECOS MIME7T08 MIME8T07
              NAMED_BIND NDBM NETINET NETINET6 NETUNIX NEWDB NIS NISPLUS
              QUEUE SCANF SMTP USERDB XDEBUG

===== SYSTEM IDENTITY (after readcf) =====
  (short domain name) $w = dodgers
  (canonical domain name) $j = dodgers.usca.ibm.com
  (subdomain name) $m = usca.ibm.com
  (node name) $k = dodgers
=====

Recipient names must be specified
# oslevel -r
5200-02
#
```

Files

Table 21. Files

Item	Description
/usr/sbin/sendmail	Contains the sendmail command.
/usr/sbin/mailq/	Contains the mail queue.
/usr/sbin/newaliases	Contains the alias database.
/usr/sbin/mailstats	Contains statistics that are found in the /usr/lib/sendmail.st file.
/etc/mail/aliases	Contains the text version of the sendmail command aliases.
/etc/mail/aliases.db	Contains a Berkeley DB formatted database for aliases.

Table 21. Files (continued)

Item	Description
/etc/mail/aliases.dir	Contains a DBM formatted database for aliases.
/etc/mail/aliases.pag	Contains a DBM formatted database for aliases.
/etc/mail/sendmail.cf	Contains the text version of the sendmail configuration file.
/etc/mail/submit.cf	Contains the text version of the sendmail configuration file. If this file exists, this file is considered as the default configuration file.
/etc/sendmail.st	Contains mail routing statistics information.
/usr/lib/smdemon.cleanu	Maintains aging copies of the log file that is found in the /var/spool/mqueue directory.
/var/spool/mqueue	Contains the temporary files and the log file that is associated with the messages in the mail queue.
/usr/bin/uux	Contains the mailer command to deliver Basic Networking Utilities (BNU) mail.
/usr/bin/bellmail	Contains the mailer command to deliver local mail.

setclock Command

Purpose

Sets the time and date for a host on a network.

Syntax

/usr/sbin/setclock [[TimeServer](#)]

Description

The **/usr/sbin/setclock** command gets the time from a network time server, and if run by a user with root user authority, sets the local time and date accordingly.

The **setclock** command takes the first response from the time server, converts the calendar clock reading found there, and displays the local date and time. If the **setclock** command is run by the root user, it calls the standard workstation entry points to set the system date and time.

If no time server responds or if the network is not operational, the **setclock** command displays a message to that effect and leaves the current date and time settings of the system unchanged.

Note: Any host running the **inetd** daemon can act as a time server.

Parameter

Item	Description
<i>TimeServer</i>	The host name or address of a network host that services TIME requests. The setclock command sends an Internet TIME service request to a time server host. If the <i>TimeServer</i> name is omitted, the setclock command sends the request to the default time server. The default time server in a DOMAIN environment is specified by the name server. Otherwise the default time server is specified in the /etc/hosts file.

Examples

1. To display the date and time using the time server host specified in the **/etc/hosts** file, enter:

```
setclock
Sat Mar 11 15:31:05 1988
```

The **setclock** command displays the proper date and time.

2. To set the date and time, enter:

```
su root
setclock host1
Thu Jan 12 15:24:15 1990
```

You must use the **su** command or log in as the root user before setting the time from the time server in host1.

setea Command

Purpose

Writes or deletes a named extended attribute to a file.

Syntax

setea **-n** *Name* [**-l**] { **-v** *Value* | **-d** | **-f** *EAFile* } *FileName* ...

Description

The **setea** command writes or deletes a named extended attribute to a file. The file must be in a file system which supports named extended attributes, such as JFS2 using **v2** extended attribute format.

Note: To prevent naming collisions, JFS2 has reserved the 8-character prefix (0xf8)SYSTEM(0xF8) for system-defined extended attributes. Avoid using this prefix for naming user-defined extended attributes.

This command is not used to set ACLs. To set ACLs, use the **aclput** command.

Flags

Item	Description
-d	Specifies to delete the named extended attribute from the file.
-f <i>EAFile</i>	<i>EAFile</i> specifies a file which contains the EA value. If an extended attribute matching the specified name already exists for the <i>FileName</i> , then the value will be changed to the value specified.
-l	Specifies to write or delete the extended attribute from the symbolic link itself rather than the file to which it is pointing.
-n <i>Name</i>	Specifies the name of the extended attribute to be written.
-v <i>Value</i>	Specifies the value of the named extended attribute. If an extended attribute matching the specified name already exists for the file, then the value will be changed to the value specified. Value is treated as a character string. It should be enclosed in quotes if it contains spaces.
<i>FileName</i> ...	Specifies the file or files from which to write or delete the extended attribute.

Exit Status

Item	Description
0	Successful completion.
Positive integer	An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To create an extended attribute with a name of Approver and a value of Grover for file design.html, enter:

```
setea -n Approver -v Grover design.html
```

2. To modify an extended attribute named Approver to new value of Joon for file design.html, enter:

```
setea -n Approver -v Joon design.html
```

3. To remove an extended attribute named Approver from file design.html, enter:

```
setea -n Approver -d design.html
```

4. To create an extended attribute with a name of Approver and a value of Zach for the symbolic link design.html, enter:

```
setea -n Approver -v Zach -l design.html
```

Location

/usr/sbin

setgroups Command

Purpose

Resets a session's process group set.

Syntax

```
setgroups [ - ] [ -a GroupSet ] [ -d GroupSet ] [ -r [ Group ] ] [ GroupSet ]
```

Description

The **setgroups** command, by default, displays the user's current group set and process group set for the current shell. A user's group set is defined in the user database files. When given a flag and a *GroupSet* parameter, this command resets the process group set as listed by the *GroupSet* parameter. The *GroupSet* parameter is a comma-separated list of group names. The available groups are defined in the user database files.

You can also use the **setgroups** command to add or delete groups from the current group set. Using the **-r** flag, you can reset the real group ID. If you specify the *Groupset* parameter but no flags, the **setgroups**

command resets all the groups and makes the first group in the list the real group. The **setgroups** command does not change the security characteristics of the controlling terminal.

When you run the **setgroups** command, the system always replaces your shell with a new one. The command replaces your shell regardless of whether the command is successful or not. For this reason, the command does not return error codes.

The **setgroups -r** command is identical to the **newgrp** command.

Flags

Item	Description
-a <i>GroupSet</i>	Adds the groups specified by the <i>GroupSet</i> parameter to the current session. The number of groups in the new set must not exceed NGROUPS_MAX groups, a value defined in the limits.h file. The real group ID is not changed.
-d <i>GroupSet</i>	Removes the groups specified by the <i>GroupSet</i> parameter from the current session. If the real group is removed, the next group listed in the current set becomes the real group.
-r <i>Group</i>	Resets the real group for the current process. If you do not specify a <i>Group</i> parameter and the current real group is not the primary group, the -r flag removes the current real group and resets the real group to the original primary group. If you specify a <i>Group</i> parameter, this behaves identically to the newgrp command.
-	Re-initializes the group set of the session to its original login state.

Security

Access Control: This command should be a general user program. This command should be installed as a program in the trusted computing base (TCB). The command should be owned by the root user with the **setuid** (SUID) bit set.

Files Accessed:

Mode	Files
r	/etc/passwd
r	/etc/group

Auditing Events:

Item	Description
Event	Information
Item	Description
USER_SetGroups	realgroup, groupset

Examples

1. As user sah, you can display your current group membership and process group set, by entering:

```
setgroups
```

Output similar to the following appears:

```
sah:
  user groups = staff,payroll
  process groups = staff,payroll
```

2. To add the `finance` group to the process group of the current session, enter:

```
setgroups -a finance
```

3. To set your real group to `finance`, enter:

```
setgroups finance,staff,payroll
```

This sets `finance` as the real group. The `staff` and `payroll` groups make up the supplementary group list.

4. To delete the `payroll` group from the current process group set, enter:

```
setgroups -d payroll
```

5. To change the process group set back to your default set, enter:

```
setgroups -
```

This resets your current session to its original state just after you log in.

Files

Item	Description
<code>/usr/bin/setgroups</code>	Contains the setgroups command.
<code>/etc/group</code>	Contains basic group attributes.
<code>/etc/passwd</code>	Contains basic user attributes.

setkst Command

Purpose

Sets the entries in the kernel security tables (KST).

Syntax

```
setkst [-q] [-b | -l | -t table1, table2,...]
```

Description

The **setkst** command reads the security databases and loads the information from the databases into the kernel security tables. By default, all of the security databases are sent to the KST. Alternatively, you can specify a specific database using the **-t** flag. If only the authorization database is the only one you specified, the role and privileged command databases are updated in the KST because they are dependent on the authorization database.

The **setkst** command checks the tables before updating the KST. If any severe error in the database is found, the **setkst** command warns the user by sending message to the **stderr**, and exits without resetting the KST. If a minor error is found in the database, a warning message is displayed, and the entry is skipped.

The **setkst** command is only functional if the system is operating in enhanced Role Based Access Control (RBAC) mode. If the system is not in enhanced RBAC mode, the command displays an error message and ends.

Flags

Item	Description
-b	Loads the KST with the information that is stored in the backup binary file on the system. If information in the binary file cannot be loaded, the tables are regenerated from the security databases.
-l	Reads the <code>loglevel</code> attribute value from the <code>syslog</code> stanza in the <code>/etc/secvars.cfg</code> file and updates the <code>loglevel</code> attribute value to the kernel. The valid values for the <code>loglevel</code> attribute are as follows: <code>all</code> , <code>crit</code> , and <code>none</code> . Any invalid value for the <code>loglevel</code> attribute are ignored by the <code>setkst</code> command.
-q	Specifies quiet mode. Warning messages that occur are not displayed when the security databases are parsed.
-t <i>table1, table2</i>	Sends the specified security databases to the KST. The parameter for the -t flag is a comma-separated list of security databases. Values for this flag are as follows: auth Authorizations database role Role database cmd Privileged command database dev Privileged device database dom Domains domobj Domain objects

Security

The **setkst** command is a privileged command. Only users that have the following authorization can run the command successfully.

Item	Description
aix.security.kst.set	Required to run the command.

Files Accessed

File	Mode
/etc/security/authorizations	r
/etc/security/privcmds	r
/etc/security/privdevs	r
/etc/security/roles	r
/etc/security/domains	r
/etc/security/domobjs	r
/etc/secvars.cfg	r

Examples

1. To send all of the security databases to the KST, enter the following command:

```
setkst
```

2. To send the **role** and **privileged** command databases to the KST, enter the following command:

```
setkst -t role,cmd
```

3. To send the domain object and domain databases to the KST, enter the following command:

```
setkst -t domobj,dm
```

setmaps Command

Purpose

Sets terminal maps or code set maps.

Syntax

To use setmaps with no input or output map file designation, type the following:

```
setmaps [ -v ] [ -c | -h ]
```

To select a file from the default directory as the code set map file, type the following:

```
setmaps [ -v ] -s -i MapName
```

To select a designated file as the code set map file, type the following:

```
setmaps [ -v ] -s -I File1
```

To select a file from the default directory as the input or output terminal map file, type the following:

```
setmaps [ -v ] [ -D ] [ -k KeyName ] [ -d DirectoryPath ] { -i | -o } MapName
```

To select files from the default directory as the input or output terminal map files, type the following:

```
setmaps [ -v ] [ -D ] [ -d DirectoryPath ] -t MapName
```

To select a designated file as the input or output terminal map file, type the following:

```
setmaps [ -v ] [ -D ] [ -k KeyName ] { -I | -O } File1
```

To load the default terminal map file for later use, type the following:

```
setmaps [ -v ] [ -D ] [ -k KeyName ] [ -r ] -l File2
```

To load a designated terminal map file for later use, type the following:

```
setmaps [ -v ] [ -D ] [ -k KeyName ] [ -r ] -L File1
```

Description

Note: If this command is run without root user authority, the code set map is not loaded, only debugged.

The **setmaps** command handles terminal and code set maps. The **-s** flag must be used for code set maps. The operating system uses input and output terminal maps to convert internal data representations to the ASCII characters supported by asynchronous terminals. If you enter the **setmaps** command with no flags, it displays the names of the current input and output terminal maps.

A terminal map is a text file containing a list of rules that associate a pattern string with a replacement string. This file normally resides in the **/usr/lib/nls/termmap** directory. The operating system uses an

input map file to map input from the keyboard to an application and an output map file to map output from an application to the display.

Terminal mapping works as follows:

1. The system collects characters in a buffer until a pattern specified by a rule in the map file matches a substring in the buffer.
2. The system then constructs and returns the replacement string specified by the rule.

This processing continues with the remaining characters in the buffer.

The rules of a terminal map can test and change the state of the pattern processor. The state is identified by a single-byte character, conventionally a digit (0 through 9). The state is reset to 0, the initial state, whenever the system loads a new map or flushes the terminal input or output buffer (such as when it processes a KILL or INTR character or when a program issues an **ioctl** system call). A terminal map can use states to detect multibyte escape sequences, among other tasks. You can test for state *x* by specifying @*x* in a pattern. You can set the state to *x* by including @*x* in the replacement string.

The **setmaps** command, when using the **-s** flag, assigns a code set map to the standard input device. The operating system uses code set maps to determine the number of bytes of memory a character requires and the number of display columns it requires.

Flags

Item	Description
-c	Clears all mappings on this terminal.
-d <i>DirectoryPath</i>	Causes the <i>DirectoryPath</i> variable to be used as the path to the directory that contains the <i>MapName</i> variable. Specifying this flag and variable overrides the /usr/lib/nls/termmap directory.
-D	Produces a debug program printout of the specified map on the standard output device before loading the map. When using this to run the debug program on new maps, do not run with root user authority until the map is fully debugged to prevent the map from actually being loaded.
-h	Prints the usage information of the setmaps command (used with the -v flag for advanced users).
-i <i>MapName</i>	Selects the /usr/lib/nls/termmap/MapName.in file as the input map. When used with the -s flag, this flag selects the /usr/lib/nls/csmmap/MapName file as the terminal code set map file.
-I <i>File1</i>	Selects the contents of the <i>File1</i> variable as the input map. The file specified by the <i>File1</i> variable can be either a full path name or a path name relative to the current working directory. When used with the -s flag, this flag selects the contents of the <i>File1</i> variable as the terminal code page map file.
-k <i>KeyName</i>	Associates the contents of the <i>KeyName</i> variable with the map being selected. This key name overrides the default key, which is normally set to the value of the <i>MapName</i> variable.
-l <i>File2</i>	Loads the /usr/lib/nls/termmap/File2 file for later use. The <i>File2</i> variable includes the full path name and suffix (if any) of the map file. Note: You must have root user authority to specify this flag.
-L <i>File1</i>	Loads the specified map for later use. The <i>File1</i> variable includes the full path name and suffix (if any) of the map file. Note: You must have root user authority to specify this flag.
-o <i>MapName</i>	Selects the /usr/lib/nls/termmap/MapName.out file as the terminal output map.

Item	Description
-O <i>File1</i>	Selects the contents of the <i>File1</i> variable as the terminal output map. The <i>File1</i> variable includes the full path name and suffix (if any) of the map file.
-r	Forces reloading of the specified map, even if it is already loaded. Terminals using the old map continue to do so until they are logged off or until their maps are explicitly reset. If you do not specify this flag, a map is loaded only if it has not already been loaded into the kernel.
Note: You must have root user authority to specify this flag.	
-s	Treats any map as a code set map.
-t <i>MapName</i>	Selects the /usr/lib/nls/termmap/MapName.in file as the terminal input map and the /usr/lib/nls/termmap/MapName.out file as the terminal output map.
-v	Selects verbose output.

All maps loaded must have unique names. Use the **-k** flag to eliminate naming conflicts. Only the **-i**, **-o**, and **-t** flags implicitly add a suffix. Other flags specifying map names should include a suffix if appropriate. If a requested map name is already loaded in the kernel, that map is used even if the path information provided on the command line implies a different map.

To reset the code set map to its original state, the **/usr/lib/nls/csmmap/sbcs** code set map should be used.

Examples

1. To display the current map settings for this terminal, enter:

```
setmaps
```

2. To clear all mapping for the current terminal, enter:

```
setmaps -c
```

3. To set up mapping (both input and output maps) for an **ibm3161-C** terminal, enter:

```
setmaps -t ibm3161-C
```

4. To load the **vt220** input map into the kernel as the **fred** map, enter:

```
setmaps -k fred -i vt220
```

5. To gather debug output for a new map called **bob** in a file called **bob.dump**, enter:

```
setmaps -D -L /tmp/bob > bob.dump
```

6. To set up a code set map conforming to the **IBM-943** code page for this terminal, enter:

```
setmaps -s -i IBM-943
```

7. To set up a code set map from the file **myEUC** for this terminal, enter:

```
setmaps -s -I myEUC
```

Files

Item	Description
/usr/bin/setmaps	Contains the setmaps command.

Item	Description
<code>/usr/lib/nls/termmap/*.in</code>	Contains input map files.
<code>/usr/lib/nls/termmap/*.out</code>	Contains output map files.
<code>/usr/lib/nls/csmmap/sbcs</code>	Contains code set map for a single-byte code page.
<code>/usr/lib/nls/csmmap/IBM-943</code>	Contains code set map for the IBM-943 code page.
<code>/usr/lib/nls/csmmap/IBM-eucJP</code>	Contains code set map for the IBM-eucJP code page.

setrunmode Command

Purpose

Sets the run mode of the system.

Syntax

setrunmode { `-c` | `-o` }

Description

The **setrunmode** command sets the run mode of the system. A run mode is either the CONFIGURATION mode or the OPERATIONAL mode.

Flags

Item	Description
<code>-c</code>	Specifies the CONFIGURATION Mode.
<code>-o</code>	Specifies the OPERATIONAL mode.

Security

Only users that have the following authorization can run the command successfully:

Item	Description
aix.mls.system.mode	Required to set the run mode.

Examples

1. To set the system in the CONFIGURATION mode, enter the following command:

```
setrunmode -c
```

2. To set the system in the OPERATIONAL mode, enter the following command:

```
setrunmode -o
```

Files

Item	Description
<code>/usr/sbin/setrunmode</code>	Contains the setrunmode command.

setsecattr Command

Purpose

Sets the security attributes of a command, a device, a privileged file, a process, or a domain-assigned object.

Syntax

setsecattr [-R *load_module*]{ **-c** | **-d** | **-p** | **-f** | **-o**} *Attribute = Value* [*Attribute = Value ...*] *Name*

Description

The **setsecattr** command sets the security attributes of the command, device, or process that is specified by the *Name* parameter. The command interprets the *Name* parameter as either a command, a device, a privileged file, or a process based on whether the **-c** (command), **-d** (device), **-f** (privileged file), or **-p** (process) flag is specified.

If you configure the system to one of the following values specified by the *Name* parameter, the system performs in the order that is specified by the **secorder** attribute of the corresponding database stanza in the **/etc/nscontrol.conf** file:

- Uses databases from multiple domains
- Sets security attributes for a privileged command
- Sets security attributes for a privileged device
- Sets security attributes for a privileged file
- Sets security attributes for a domain-assigned object

Only the first matching entry is modified. Duplicate entries from the remaining domains are not modified. Use the **-R** flag to modify the entry from a specific domain. If no matching entry is found in any of the domains, a new entry for the *Name* parameter is created in the first domain. Use the **-R** flag to add the entry to a specific domain.

To set a value for an attribute, specify the attribute name and the new value with the *Attribute=Value* parameter. To clear an attribute, specify the *Attribute=* for the *Attribute=Value* pair. To make incremental changes to attributes, whose values are lists, specify the *Attribute=Value* pairs as *Attribute=+Value*, or *Attribute=-Value*. If you specify the *Attribute=+Value*, the value is added onto the existing value for the attribute. If you specify the *Attribute=-Value*, the value is removed from the existing value for the attribute.

Flags

Item	Description
-c	Specifies that the security attributes of a command on the system are to be set. If the command name that you specified using the <i>Name</i> parameter is not in the privileged command database, a command entry is created in the /etc/security/privcmds privileged command database. If an attribute is being cleared and is the only attribute set for the command, the command is removed from the privileged command database. Modifications made to the privileged command database are not used until the database is sent to the kernel security tables using the setkst command.

Item	Description
-d	Specifies that the security attributes of a device on the system are to be set. If the device name you specify using the <i>Name</i> parameter is not in the privileged device database, a device entry is created in the /etc/security/privdevs privileged device database. If an attribute is being cleared and is the only attribute set for the device, the device is removed from the privileged device database. Modifications made to the privileged device database are not used until the database is sent to the kernel security tables using the setkst command.
-f	Specifies that the security attributes of a privileged file on the system are to be set. Changes requested through the <i>Attribute=Value</i> pairs are made in the /etc/security/privfiles privileged file database. If the specified file is not in the privileged file database, a file entry is created in the database. If an attribute is being cleared and is the only attribute set for the command, the command is removed from the privileged file database.
-o	Specifies that the security attributes of an object on the system are to be set. If the object name that you specified using the <i>Name</i> parameter is not in the domain object database, an object entry is created in the /etc/security/domobjs domain object database. If an attribute is being cleared and is the only attribute set for the object, the object entry is removed from the domain object database. Modifications made to the domain object database are not used until the database is sent to the kernel security tables using the setkst command.
-p	Specifies that the numeric process identifier (PID) of an active process on the system are to be set. Changes that you specify with the <i>Attribute=Value</i> pairs immediately affects the state of the specified active process. Modifications are not saved in a database.
-R <i>load_module</i>	Specifies the loadable module to use for security attribute modification.

Parameters

Item

Attribute = Value

Description

Sets the value of a security attribute for the object. The list of valid attribute names are dependent on the object type as specified using the **-c**, **-d**, **-p**, and **-o** flags.

Use the following attributes for the privileged command database (**-c**) flag:

accessauths

Specifies access authorizations. Specifies a comma-separated list of authorization names. You can specify a total of sixteen authorization. A user with any of the authorizations that you specified can run the command. This attribute has three special additional values: `ALLOW_OWNER`, `ALLOW_GROUP`, and `ALLOW_ALL` that allows a command owner, a group, or all users to run the command without checking for access authorizations.

authprivs

Specifies authorized privileges. Specifies a list of authorizations and privilege pairs that grant additional privileges to the process. The authorization and its corresponding privileges are separated by an equal sign (=), individual privileges are separated by a plus sign (+), and authorization or privilege pairs are separated by a comma (,), as shown in the following examples:

```
auth=priv+priv+...,auth=priv+priv+...,...
```

You can specify a maximum of sixteen pairs of authorizations or privileges. Specifies roles, the users of which need to be authenticated before command can be executed successfully. Specifies a comma separated list of roles. Each role should be authenticated by different users such as no user can perform the authentication for more than one role at a time.

authroles

Specifies the user roles that need to be authenticated before the command can run successfully. If listing multiple roles, separate each role with a comma. For example:

```
authroles=so,isso
```

Each role must be authenticated by different users. For example, no one user can perform the authentication for more than one role.

innateprivs

Specifies the innate privileges. Specifies a comma-separated list of privileges that are assigned to the process when the command is run.

inheritprivs

Specifies inheritable privileges. Specifies a comma-separated list of privileges that are passed to child processes.

euid

Specifies the effective user ID to assume when the command is run.

egid

Specifies the effective group ID to assume when the command is run.

Item

Description

ruid

Specifies the real user ID to assume when the command is run. Only valid value is 0. This attribute value will be ignored if the command provides access to all users by specifying the special value ALLOW_ALL in its **accessauths** attribute.

secflags

Specifies the file security flags. Specifies a comma-separated list of security flags. Use the following values for this flag:

FSF_EPS

Causes the maximum privilege set to be loaded into the effective privilege set when the command is run.

Use the following attributes for the privileged device database (**-d**) flag:

readprivs

Specifies a comma-separated list of privileges that a user or a process must have for read access to the device. You can specify a maximum of eight privileges. The user or process must have one of the listed privileges to read from the device.

writeprivs

Specifies a comma-separated list of privileges that a user or a process must have for write access to the device. You can specify a maximum of eight privileges. The user or process must have one of the listed privileges to write to the device.

Item

Description

Use the following attributes for the privileged file (**-f**) flag:

readauths

Specify the read access authorizations. Specify a comma-separated list of authorization names. A user with any of the authorizations can read the file.

writeauths

Specify the write access authorizations. Specify a comma-separated list of authorization names. A user with any of the authorizations can read or write the file.

Use the following attributes for the privileged process (**-p**) flag:

eprivs

Specify the effective privilege set. Specify a comma-separated list of privileges that are to be active for the process. The process might remove the privileges from this set and add the privileges from the maximum privilege set to its effective privilege set.

iprivs

Specifies the inheritable privilege set. Specifies a comma-separated list of privileges that are passed to child processes' effective and maximum privilege sets. The inheritable privilege set is a subset of the limiting privilege set.

mprivs

Specify a maximum privilege set. Specify a comma-separated list of privileges that the process can add to its effective privilege set. The maximum privilege set is a superset of the effective privilege set.

lprivs

Specify the limiting privilege set. Specify a comma-separated list of privileges that make up the maximum possible privilege set for a process. The limiting privilege set is a superset of the maximum privilege set.

uprivs

Specify the used privilege set. Specify a comma-separated list of privileges that are used during the life of the process. This set is mainly used by the **tracepriv** command.

Item	Description
	Use the following attributes for the domain-assigned object database (-o) flag:
domains	Specify a comma-separated list of domains the objects belong to.
conflictsets	Specify a comma-separated list of domains that are excluded from accessing the object.
objtype	Specify the type of the object. Valid values are device, netint, netport and file.
secflags	Specify the security flags for the object. Valid values are: • FSF_DOM_ANY: This value specifies that a process can access the object if it has any of the domains given in the domains attribute. • FSF_DOM_ALL: Specifies that a process can access the object only if it has all the domains as specified in the domains attribute. This is the default value if no secflags is specified. The FSF_DOM_ANY and FSF_DOM_ALL are mutually exclusive flags.
<i>Name</i>	Specify the object to modify. The <i>Name</i> parameter is interpreted according to the flags that you specify. One name must be indicated for processing at a time.

Security

The **setsecattr** command is a privileged command. It is owned by the root user and the security group, with the mode set to 755. You must have assume a role with at least one of the following authorizations to run the command successfully. For trusted process, the auditing system will not log any object auditing events for the respective process. However, users can capture events using event auditing.

Item	Description
aix.security.cmd.set	Required to modify the attributes of a command with the -c flag.
aix.security.device.set	Required to modify the attributes of a device with the -d flag.
aix.security.file.set	Required to modify the attributes of a device with the -f flag.
aix.security.proc.set	Required to modify the attributes of a process with the -p flag.
aix.security.dobject.set	Required to modify the attributes of a process with the -o flag.

File Accessed

Item	Description
File	Mode
/etc/security/privcmds	rw
/etc/security/privdevs	rw
/etc/security/privfiles	rw
/etc/security/domobjs	rw

Examples

1. To set an authorized privilege pair for the `/usr/sbin/mount` command, enter the following command:

```
setsecattr -c authprivs=aix.fs.manage.mount=PV_FS_MOUNT /usr/sbin/mount
```

2. To incrementally add the PV_AU_WRITE and PV_DAC_W privileges to the existing set of writing privileges for the `/dev/mydev` device, enter the following command:

```
setsecattr -d writeprivs=+PV_AU_WRITE,PV_DAC_W /dev/mydev
```

3. To set a read authorization for the `/etc/security/user` file, enter the following command:

```
setsecattr -f readauths=aix.security.user.change /etc/security/user
```

4. To incrementally remove the PV_DAC_R privilege from the effective privilege set of an active process, enter the following command:

```
setsecattr -p eprivs=-PV_DAC_R 35875
```

5. To set the access authorizations for the `/usr/sbin/mount` command in LDAP, enter the following command:

```
setsecattr -R LDAP -c accessauths=aix.fs.manage.mount /usr/sbin/mount
```

6. To set the domains on the network interface **en0**, enter the following command:

```
setsecattr -o domains=INTRANET,APPLICATION conflictsets=INTERNET  
objtype=netint secflags=FSF_DOM_ANY en0
```

setsecconf Command

Purpose

Loads the system security flag settings into the kernel.

Syntax

```
setsecconf { -c | -o } [ Attribute = Value ... ]
```

Description

The **setsecconf** command loads the system security flag settings into the kernel. If you specify any attributes, the values of these attributes are stored and used when the system is restarted. This command can change the setting of the flags for the CONFIGURATION and OPERATIONAL modes of the system, but these flags can be changed only when the system is in the CONFIGURATION mode.

Flags

Item	Description
-c	Specifies the CONFIGURATION mode.
-o	Specifies the OPERATIONAL mode.

Parameters

Item	Description
<i>Attribute</i>	You can specify the following attributes: root Specifies whether the root user can log in to the system. If enabled, the root user can log in to the system. If disabled, the root user cannot log in to the system. For more information, see the information in the "Disabling the root user" topic. tnet Specifies the Advanced Security Network. If enabled, all of the data packets are labeled. tlwrite Specifies whether to enforce the write access checks on the integrity labels (TLs). If enabled, TLs are checked on write, remove, and rename operations. If disabled, TLs can be set, but are ignored on write access checks. tread Specifies whether to enforce the read access checks on the integrity labels (TLs). If enabled, TLs are checked on read operations. If disabled, TLs can be set, but are ignored on read access checks. traceauth Specifies if authorization tracing is enabled. If enabled, the authorizations used in a process are traced and logged in a process credential. The lssecattr command is used to display used authorizations. If disabled, no authorizations are traced in a system. By default, this flag is disabled. This flag is only meaningful in the operational mode. sl Specifies whether to enforce the Mandatory Access Control (MAC) flag. If enabled, MAC is enforced. If not enabled, sensitivity labels (SLs) can be configured, but not used to determine the access to files and other objects. tlib Specifies whether to recognize and enforce the Trusted Computing Base (TCB). If enabled, the TCB flag on file system objects is recognized and enforced. If disabled, the TCB on objects is ignored and all objects are treated as if they are not TCB objects. <i>Value</i> Specifies a value that is either enable or disable .

Security

The **setsecconf** command is a privileged command. Only users that have the following authorization can run the command successfully:

Item	Description
aix.mls.system.config.write	Required to set the system configuration flags.

Exit Status

The **setsecconf** command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Examples

1. To turn on the trusted network and turn off the integrity read system flags for the CONFIGURATION mode run, enter the following command:

```
setseccomp -c tnet=enable tload=disable
```

2. To turn on the integrity write system flag for the OPERATIONAL mode run, enter the following command:

```
setseccomp -o tlwrite=enable
```

Files

Item	Description
<code>/usr/sbin/setseccomp</code>	Contains the setseccomp command.

setseccomp Command

Purpose

Resets the protected state environment of a user.

Syntax

setseccomp [-] *NewEnvironment*

Description

The **setseccomp** command resets your protected state environment while you are logged in. The protected state environment is defined as a set of variables. These variables are kept in the kernel and can be modified only by a **SETUINFO** system call. The **setseccomp** command uses the variables specified by the *NewEnvironment* parameter. This parameter consists of *EnvironmentVariable=Value* definitions separated by a blank space. For information on environment variables, see **environment** File.

You cannot reset the following environment variables with the **setseccomp** command:

Item	Description
NAME	Your last authenticated user name. This corresponds to the real user ID of the current process.
TTY	The name of the terminal on which you logged in. This corresponds to the initial controlling terminal for the process. This variable cannot be set for processes initiated without a <i>full login</i> . A full login is a login initiated by the getty command.
LOGNAME	The name under which you logged in, if the current session was started from a terminal login program. If the session was not started from a terminal, this variable is not set.

If you enter the **setseccomp** command without any defined variables, it displays the current protected state. The **setseccomp** command does not change the security characteristics of the controlling terminal.

When you run the **setseccomp** command, it replaces your current shell and gives you a new one. The command replaces your shell regardless of whether it completed successfully or not. For this reason, the command does not return error codes.

Flags

Item	Description

-	Reinitializes the environment as if the user had just logged in to the system. Otherwise, the environment is not changed.

Security

Access Control: This command should be a standard user program. This command should be installed as a program in the trusted computing base (TCB). The command should be owned by the root user with the **setuid** (SUID) bit set.

Files Accessed:

Mode	File
r	/etc/environment
r	/etc/security/envIRON

Auditing Events:

Event	Information
USER_SetEnv	new environment string

Examples

1. To display the current environment variables, enter:

```
setenv
```

2. To add the PSEUDO=tom protected environment variable, enter:

```
setenv PSEUDO=tom
```

This example sets a user name for the **PSEUDO** protected environment variable.

Files

Item	Description
/usr/bin/setenv	Specifies the path to the setenv command.
/etc/environment	Contains environment information for each user.
/etc/security/envIRON	Contains privileged environment information for each user.

setsyslab Command

Purpose

Sets the minimum and maximum sensitivity labels of the system.

Syntax

setsyslab

Description

The **setsyslab** command sets the system minimum sensitivity label (SL), maximum SL, minimum integrity label (TL), and maximum TL. The values of the SL and TL are taken from the **/etc/security/enc/LabelEncodings** label encodings file.

Security

The **setsyslab** command is a privileged command. Only users that have the following authorization can run the command successfully:

Item	Description
aix.mls.system.label.write	Required to set system labels.

Files Accessed:

Item	Description
Mode	File
r	/etc/security/enc/LabelEncodings

Examples

1. To set system labels, enter the following command:

```
setsyslab
```

Files

Item	Description
/usr/sbin/setsyslab	Contains the setsyslab command.
/etc/security/enc/LabelEncodings	System default label encodings file.

settime Command

Purpose

Updates access and modification times of a file.

Syntax

settime [[MMddhhmm[yy]]] [**-f** *ReferenceFile*] File ...

Description

settime updates the argument files with the current access and modification times by default. The file is not created if it does not exist. The **settime** command silently continues its operation if the file does not exist.

Note: Any dates beyond and including the year 2038 are not valid for the **settime** command.

Flags

Item	Description

-f <i>ReferenceFile</i>	Use the corresponding time of <i>ReferenceFile</i> instead of the current time.

Parameters

Item	Description

<i>MMddhhmm</i> [<i>yy</i>]	Time is specified for the settime command in the format <i>MMddhhmm</i> or <i>MMddhhmmyy</i> , where <i>MM</i> is a two-digit representation of the month, <i>dd</i> is a two-digit representation of the day of the month, <i>hh</i> is a two-digit representation of the hour, <i>mm</i> is a two-digit representation of the minute, and <i>yy</i> is a two-digit representation of the year.

<i>File</i>	Specifies the name of a file or a space separated list of files.

Exit Status

0

The command completed successfully.

>0

An error occurred.

The return code from **settime** is the number of specified files for which the times could not be successfully modified.

Examples

1. To update the access and modification times of the file "infile" to the current time, enter:

```
settime infile
```

2. To update the access and modification times of "infile" to be the same as "reffile", enter:

```
settime -f reffile infile
```

3. To update the access and modification times of multiple files, enter:

```
settime file1 file2 file3
```

4. To update the access and modification times of a file to April 9th 2002 with time 23:59, enter:

```
settime 0409235902 infile
```

Files

Item	Description

/usr/bin/settime	Contains the settime command.

setuname Command

Purpose

Sets the node name of the system.

Syntax

setuname [**-t**] **-n** *Node*

Description

The **setuname** command is used to set the node name of the system. The **-n** option must be specified. Only users with root authority can set the node name. The change can be made temporary by using the **-t** option. The node name will be modified only on the current running kernel if a temporary change is requested. The nodename set temporarily will not persist after a reboot. Without the **-t** option the node name is changed permanently in the ODM database.

Flags

Item

-n *Node*

Description

Specifies that the node name has to be changed. This option is required. *Node* is the primary node name for the host. This can be the UUCP communications network name for the system.

-t

Temporary change. No attempt will be made to make the change permanent. The original name will be restored after reboot.

Exit Status

0

The command completed successfully.

>0

An error occurred.

Examples

1. To temporarily change the node name to "orion", enter:

```
setuname -t -n orion
```

2. To permanently change the node name to "orion", enter:

```
setuname -n orion
```

Files

Item

/usr/bin/setuname

Description

Contains the setuname command.

sh Command

Purpose

Invokes the default shell.

Syntax

Refer to the syntax of the **ksh** command. The **/usr/bin/sh** file is linked to the Korn shell.

Description

The **sh** command invokes the default shell and uses its syntax and flags. The shell linked to the **/usr/bin/sh** path is the default shell. The standard configuration of the operating system links the **/usr/bin/sh** path to the Korn shell.

Flags

Refer to the flags for the Korn shell (**ksh** command).

Files

Item	Description
/usr/bin/sh	Contains the sh command.

shconf Command

Purpose

Manages the system hang detection parameters.

Syntax

shconf -d

shconf -R -l *Name*

shconf {**-D** [**-O**] | **-E** [**-O**] } [**-H**] **-l** *Name*

shconf -l *Name* [**-a** *Attribute=Value*] ...

Description

The **shconf** command is used to display or specify the parameters of the priority problem detection and lost I/O detection.

For the priority problem, the user can specify five actions described below and for each action, the user can specify the priority level to check, the time out while no process or thread executes at a lower or equal priority, the terminal device for the warning action, and the getty action:

Item	Description
pp_cmd	Launches a command specified by the path parameter.
pp_errlog	Logs an error in error log.

Item	Description
pp_login	Launches a getty at the highest priority on the serial line specified by the terminal device parameter (term).
pp_reboot	Reboots the system.
pp_warning	Displays a warning message on the console specified by the terminal device parameter (term).

For lost I/O, the user can specify the actions listed below and **errlog**, which is automatic when lost I/O detection is enabled. There is a unique timeout which applies to all enabled actions.

Item	Description
lio_warning	Displays a warning message on the console specified by the terminal device parameter (term).
lio_reboot	Creates a system dump and reboots the system.

Note: The `shconf` command only supports the `tty` and `console` terminal types.

Flags

Item	Description
-d	Displays if priority problem detection and lost I/O detection are enabled or not.
-R	Restore the default values for a specified name of detection.
-a <i>Attribute=Value</i>	Specifies the attribute value pairs used for changing specific attribute values.
-D	Displays the default values for a specified name of detection.
-E	Displays the effective values for a specified name of detection.
-H	Displays the headers above the column output. When used together, the -O flag overrides the -H flag.
-l <i>Name</i>	Specifies the detection name.
-O	Displays all attribute names separated by colons and, on the second line, displays all the corresponding attribute values separated by colons. The attribute values are current values when the -E flag is also specified and default values when the -D flag is specified. This flag cannot be used with the -a flag.

Files

Item	Description
/usr/sbin/shconf	Contains the shconf command.

shell Command

Purpose

Executes a shell with the user's default credentials and environment.

Syntax

shell

Description

The **shell** command re-initializes a user's login session. When the command is given, the port characteristics of the process's controlling terminal are reset and all access to the port is revoked. The **shell** command then resets the process credentials and environment to the defaults established for the user and executes the user's initial program. All credentials and environment are established according to the login user ID of the invoking process.

If the **shell** command is invoked on the trusted path and the user's **tpath** attribute in the **/etc/security/user** file does not have a value of **always**, the trusted environment of the terminal is not maintained.

Note: The **shell** command does not reset the login ID of the user.

Security

Access Control: The command should be **setuid** to the root user to reset the user's process credentials, and grant execute (x) access to all users. The command should have the **trusted computing base** attribute.

Files Accessed:

Mode	File
r	/etc/passwd
r	/etc/group
r	/etc/security/audit/config
r	/etc/security/envIRON
r	/etc/security/limits
r	/etc/security/user

Auditing Events:

Event	Information
USER_Shell	portname

Examples

To re-initialize your session to your default credentials and environment after using the trusted shell (**tsh**), enter:

```
shell
```

Files

Item	Description
<code>/usr/bin/shell</code>	Contains the shell command.
<code>/etc/security/user</code>	Contains the extended attributes of users.
<code>/etc/passwd</code>	Contains user IDs.
<code>/etc/group</code>	Contains group IDs.
<code>/etc/security/audit/config</code>	Contains the audit configuration information.
<code>/etc/security/envIRON</code>	Defines the environment attributes for users.
<code>/etc/security/limits</code>	Defines process resource limits for each user.

show Command

Purpose

Shows messages.

Syntax

```
show [ +Folder ] [ -draft | Messages ] [ -header | -noheader ] [ -showproc CommandString | -noshowproc ]
```

Description

The **show** command displays the contents of messages. If standard output is not a display, the **show** command lists each message with a one-line header and two separation lines. By default, the **show** command displays the current message in the current folder.

The **show** command invokes a listing program to create the list. The default listing program is `/usr/bin/more`. You can define your own default with the `showproc :` entry in your `$HOME/.mh_profile` file. If you set the `showproc :` entry to `mhl`, the **show** command calls an internal **mhl** routine instead of the **mhl** command. You can also specify the program to perform a listing in the *CommandString* parameter of the **-showproc** flag.

The **show** command passes any flags it does not recognize to the listing program. Thus, you can specify flags for the listing program, as well as for the **show** command.

If the `Unseen-Sequence :` entry is present in your `$HOME/.mh_profile` file and the entry is not empty, the **show** command removes each of the messages shown from each sequence named by the profile entry. If several messages are specified, the last message shown becomes the current message.

Flags

Item	Description
-draft	Shows the <i>UserMhDirectory/draft</i> file if it exists.
<i>+Folder</i>	Specifies a folder. The current folder is the default.
-header	Displays a one-line description of the message being shown. The description includes the folder name and message number. If you show more than one message, this flag does not produce message headers. The -header flag is the default.

Item	Description
-help	Lists the command syntax, available switches (toggles), and version information. Note: For MH, the name of this flag must be fully spelled out.
<i>Messages</i>	Specifies the messages to show. You can specify several messages, a range of messages, or a single message. Use the following references to specify messages: Number Number of the message. Sequence A group of messages specified by the user. Recognized values include: all All messages in a folder. cur or . (period) Current message. This is the default. first First message in a folder. last Last message in a folder. next Message following the current message. prev Message preceding the current message.
-noheader	Prevents display of a one-line description of each message.
-noshowproc	Uses the /usr/bin/cat command to perform the listing. This is the default.
-showproc <i>CommandString</i>	Uses the specified command string to perform the listing.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Profile Entries

The following entries are entered in the *UserMhDirectory/.mh_profile* file:

Item	Description
Current-Folder:	Sets the default current folder.
Path:	Specifies the user's MH directory.
showproc:	Specifies the program used to show messages.
Unseen-Sequence:	Specifies the sequences used to keep track of the unseen messages.

Examples

1. To display the contents of the current message in the current folder one screen at a time, enter:

```
show
```

If the message continues for more than one screen, press the Enter key until you have read the entire message.

2. To see the contents of all the messages in the current folder, enter:

```
show all
```

If the messages continue for more than one screen, press the Enter key until you have read all the messages.

3. To see the contents of message 5 in the meetings folder, enter:

```
show +meetings 5
```

4. To see the contents of all the messages belonging to the weekly sequence in the meeting folder, enter:

```
show +meeting weekly
```

Files

Item	Description
\$HOME/.mh_profile	Specifies the MH user profile.
<i>UserMhDirectory/draft</i>	Contains the current message draft.
/usr/bin/show	Contains the show command.

showmount Command

Purpose

Displays a list of all clients that have remotely mounted file systems.

Syntax

```
/usr/bin/showmount [ -a ] [ -d ] [ -e ] [ Host ]
```

Description

The **showmount** command displays a list of all clients that have remotely mounted a file system from a specified machine in the *Host* parameter. This information is maintained by the **mountd** daemon on the *Host* parameter. This information is saved in the **/etc/rmtab** file in case the server crashes. The default value for the *Host* parameter is the value returned by the **hostname** command.

Note: If a client crashes, its entry will not be removed from the list until the client reboots and starts the **umount -a** command.

Note: The **showmount** command returns information maintained by the **mountd** daemon. Because NFS Version 4 does not use the **mountd** daemon, **showmount** will not return information about version 4 mounts.

Flags

Item	Description

-a	Prints all remote mounts in the format <i>HostName:Directory</i> , in which <i>HostName</i> is the name of the client and <i>Directory</i> is a directory pathname that has been remotely mounted.
-d	Lists only directories that have been remotely mounted by clients.
-e	Prints the list of exported directories.

Examples

1. To display a list of all remote directories that are mounted by a host, enter the following command:

```
/usr/bin/showmount -a zeus
```

In this example, the showmount command produces a list of all of the remote directories mounted by the clients on the host machine named zeus.

2. To display a list of only the directories that are mounted by a client on the host, enter the following command:

```
/usr/bin/showmount -d athena
```

In this example, the showmount command produces a list of all remote directories mounted by the client machines on the host named athena.

3. To print a list of all directories that are exported from a machine, enter the following command:

```
/usr/bin/showmount -e zeus
```

In this example, the showmount command produces a list of all remote directories that are exported by the host machine named zeus except the ones that are exported only with NFS version 4.

Files

Item	Description
/etc/rmtab	Contains information about the current state of all exported directories.
/etc/xtab	Lists currently exported directories.

shutacct Command

Purpose

Turns off processing accounting.

Syntax

```
/usr/sbin/acct/shutacct [ "Reason" ]
```

Description

The **shutacct** command turns off process accounting and calls the **acctwtmpt** command to add a record stating the reason to the **/var/adm/wtmp** file. The **shutacct** command is invoked by the **shutdown** command.

Note: It is necessary to place quotation marks around the *Reason* value in the **/var/adm/wtmp** file.

Variables

Item	Description

<i>Reason</i>	Specifies the reason for accounting system shutdown. This value is optional.

Security

Access Control: This command should grant execute (x) access only to members of the adm group.

Files

Item	Description

<i>/usr/sbin/acct</i>	The path to the accounting commands.

<i>/var/adm/wtmp</i>	The login and logout history file.

shutdown Command

Purpose

Ends system operation.

Syntax

shutdown [-d] [-F] [-h] [-i] [-k] [-l] [-m] [-p] [-r] [-t *mmddHHMM* [*yy*]] [-u] [-v] [+Time] [Message]]

Description

The **shutdown** command halts the operating system. Only a user with root user authority can run this command. During the default shutdown, users are notified (by a **wall** command) of the impending system shutdown with a message. However, shutdown is not complete until the user receives a shutdown completion message. Do not attempt to restart the system or turn off the system before the shutdown completion message is displayed; otherwise, file system damage can result.

Note: The `halt` completed message is not displayed on the tty from which shutdown is invoked if it is connected to the system through a multiport adapter.

As shutdown time approaches, warning messages are displayed on the terminals of all users on the system.

After the specified number of seconds (60 by default), the system stops the accounting and error logging processes and writes an entry to the error log. The **shutdown** command then runs the **killall** command to end any remaining processes and runs the **sync** command to flush all memory resident disk blocks. Finally, it unmounts the file systems and calls the **halt** command.

Note: Users who have files open on the node that is running the **shutdown** command, but who are not logged in to that node, are not notified about the shutdown.

If you request a complete halt to the operating system, the **shutdown** command stops all processes, unmounts all file systems, and calls the **halt** command.

The system administrator can place local customized shutdown procedures in a shell script named **/etc/rc.shutdown**. This script runs at the beginning of the shutdown if it exists. If the script runs but fails with a non-zero return code, the shutdown stops.

Attention: If you are bringing the system down to maintenance mode, you must run the **shutdown** command from the **/** (root) directory to ensure that it can cleanly unmount the file systems.

Note: By default, if issued on models having a power supply capable of software control, the **shutdown** command turns off the system.

Flags

Item	Description
-d	Brings the system down from a distributed mode to a multiuser mode.
-F	Does a fast shutdown, bypassing the messages to other users and bringing the system down as quickly as possible. The +Time [Message] options are ignored if the -F flag is specified.
-h	Halts the operating system completely; same as the -v flag.
-i	Specifies interactive mode. Displays interactive messages to guide the user through the shutdown.
-k	Allows the administrator to broadcast the shutdown warning messages <i>without</i> causing the system to shut down. When the -k flag is used, no other shutdown activity occurs except for sending messages. For example, no processes are killed, no activity is logged in /etc/shutdown.log if the -l flag is specified, and if an /etc/rc.shutdown script exists it does not run.
-l	Creates/appends the /etc/shutdown.log file that contains information about the filesystems, daemons, user login, licensing services, network interfaces being brought down. The file may be used for diagnostic and debugging purposes in the event of shutdown failures. Note: Ensure that there is enough disk space for the shutdown command to log the entries while using this flag.
-m	Brings the system down to maintenance (single user) mode.
-p	Halts the system without a power down. This is used by uninterruptible power supply (UPS). Note: The -p flag will have no effect if used in combination with flags not requiring a permanent halt. Power will still be turned off if other operands request a delayed power-on and reboot
-r	Restarts the system after being shutdown with the reboot command.

Item	Description
-t <i>mmddHHMM</i> [<i>yy</i>]	Shuts down the system immediately and then restarts the system on the date specified by <i>mmddHHMM</i> [<i>yy</i>] where mm Specifies the month. dd Specifies the day. HH Specifies the hour. MM Specifies the minute. yy Specifies the year. The shutdown -t flag cannot be used with the -v or -h option. Note: This option is only supported on systems that have a power supply which automatically turns power off at shutdown and an alarm to allow reboot at a later time. Systems without this capability may hang or may reboot immediately after shutdown.
-u	This flag is used by diagnostics to update the flash-memory and reboot.
-v	Halts the operating system completely.

Parameters

Item	Description
+Time	Specifies the time at which the shutdown command stops the system. An immediate shutdown is indicated by the word now displayed on the screen. A future time can be specified in one of two formats: +number or hour:minute . The first form brings the system down in the specified number of minutes and the second brings the system down at the time of day indicated (as a 24-hour clock). If the <i>Message</i> parameter is specified, the <i>Time</i> parameter must also be specified.
<i>Message</i>	Specifies the message

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To turn off the machine, enter:

```
shutdown
```

This shuts down the system, waiting 1 minute before stopping the user processes and the **init** process.

2. To give users more time to finish what they are doing and bring the system to maintenance mode, enter:

```
shutdown -m +2
```

This brings the system down from multiuser mode to maintenance mode after waiting 2 minutes.

Files

Item	Description
<code>/usr/sbin/shutdown</code>	Contains the shutdown command.

sisraidmgr Command

Purpose

Uses and maintains a Peripheral Component Interconnect-X (PCI-X) SCSI Redundant Array of Independent Disks (RAID) controller.

Syntax

```
sisraidmgr [ -A -l hdisk# [ -z pdisk ] [ -f ] ]
sisraidmgr [ -B -l adptr# -b ioa_opt [ -r raid_level ] ]
sisraidmgr [ -C [ -r raid_level -s stripe_size (in KB) -z pdisk_list ] ]
sisraidmgr [ -D -l adptr# [-d hdisk | -e serial_num ] ]
sisraidmgr [ -F [ -z pdisk_list ] ]
sisraidmgr [ -H [ -z pdisk_list ] ]
sisraidmgr [ -I [ -z pdisk_list ] ]
sisraidmgr [ -L -l adptr# [ -a display_opt [ -v sisarray_opt -p pdisk_opt -j jbod_opt ] ] ]
sisraidmgr [ -L -l hdisk# [ -a display_opt [ -v sisarray_opt -p pdisk_opt ] ] ]
sisraidmgr [ -L -l pdisk# [ -p pdisk_opt ] ]
sisraidmgr [ -M -l adptr# -o cmd_opt ]
sisraidmgr [ -P -z drive_list { pdisks | hdisks } ]
sisraidmgr [ -R -z pdisk_list ]
sisraidmgr [ -S -l adptr# ]
sisraidmgr [ -U -z pdisk_list ]
sisraidmgr [ -W -l adptr# -o cmd_opt ]
sisraidmgr [ -X -l adptr# -o cmd_opt ]
sisraidmgr [ -Y -l hdisk# [ -x cmd_opt ] ]
```

Description

The `sisraidmgr` command is used to create, delete, and maintain RAID arrays on a PCI-X SCSI RAID controller.



Attention: See the *PCI-X SCSI RAID Controller Reference Guide for AIX* and become familiar with the storage management concepts before you run the **sisraidmgr** command.



Attention: The *System Management Interface Tool (SMIT)* **smit pxdam** fast path is the preferred method to manage a PCI-X SCSI RAID Controller.



Attention: Service tasks require special training and must not be performed by nonservice personnel.

Flags

Item	Description
-A	Adds a device to an existing array. The performance is not optimal when you use this option because the included device does not contain parity, and the data is not restriped. -l lname The logical name of the array. -z pdisks The drives to be included. -f The option to force the include operation in the situation where the disks to be included might not be known; that is, they might be 0.
-B	Lists information about what the adapter supports. -l lname The logical name of the adapter. -b ioa_support_opt 1 Displays supported RAID levels for the lname option. This is the default option. 2 Displays supported stripe size for the lname and raid_level options. 3 Displays the minimum number of devices for the raid_level option. 4 Displays the maximum number of devices for the raid_level option. 5 Displays the minimum multiple number of devices for the raid_level option. -r raid_level Shows supported stripe sizes for this RAID level.
-C	Creates a RAID array. -r raid_level { 0, 5, or 10 (RAID 1+0) } -s stripe_size (in KB) If not specified, the default (64 KB) is used. -z pdisk_list Lists pdisks to include in the new array. For example, 'pdisk2 pdisk3 pdisk4' must be connected to the same adapter.
-D	Deletes a RAID array. -l lname The logical name of the adapter. -d hdisk The name of the array to be deleted. -e serial_num The serial number of the array to be deleted. Use this option only if the array name is unknown.
-F	Formats the pdisks for recovery. (format 522-byte formatted disks). -z drive_list A list of pdisks to format.
-H	Adds a hot spare device. -z pdisk_list A list of pdisks to be made hot spare devices.
-I	Removes a hot spare device. -z pdisk_list A list of pdisks to be removed from being hot spare devices.

Item	Description
-L	Lists advance function information.
-l lname	The device for which information is displayed. It can be a RAID adapter (<i>sisioa0</i>), a RAID array (<i>hdisk8</i>), or a physical disk (<i>pdisk5</i>).
-a display_opt	
0	Displays all configuration information for the lname option. This is the default option.
1	Displays only the logical device information for the lname option.
2	Displays only the physical device information for the lname option.
-v sisarray_opt	
0	Displays all arrays. This is the default.
1	Displays only arrays that are candidates for the Delete Array option.
2	Displays only arrays that are candidates for the Rsync Protection option.
3	Displays only arrays that are candidates for including additional devices.
4	Displays only ODM arrays that have no adapter information.
-p pdisk_opt	
0	Displays all pdisks. This is the default.
1	Displays only pdisks that are candidates for the Prepare option.
2	Displays only pdisks that are candidates for the Start RAID option.
3	Displays only pdisks that are candidates for the Add Hot Spare option.
4	Displays only pdisks that are candidates for the Remove Hot Spare option.
5	Displays only pdisks that are candidates to be added to an existing array.
6	Displays only pdisks that are candidates for the Rebuild option.
7	Displays only pdisks that are candidates for the Recovery Format option.
8	Displays only ODM pdisks that have no adapter information.
9	Displays only pdisks that are candidates for the Unprepare option (522 - 512).
10	Displays only pdisks that, if prepared, would be candidates to be added to an existing array.
-j jbod_opt	
0	Displays no JBOD hdisks. This is the default option.
1	Displays all JBOD hdisks.
2	Displays only JBOD hdisks that are candidates for the Prepare option (512 - 522).

Item	Description
-M	Maintains the rechargeable battery. -l lname The logical name of the adapter. -o cmd_option The command options follow: 0 Displays rechargeable battery information. 1 Forces a rechargeable battery error. 2 Starts caching after concurrent battery replace. 3 Queries candidates for concurrently starting batteries.
-P	Prepares devices; that is, creates array candidates physical disks. -z drive_list A list of either JBOD hdisks, pdisks, or both to become an array candidate.
-Q	Sets or Clears pdisk error suppression attributes. -z pdisk_list A list of pdisks for attributes to be applied or cleared. -o cmd_option A 1-byte hexadecimal string that specifies which error suppression bits to turn on or off.
-R	Rebuilds devices; that is, reconstructs a degraded array. -z pdisk_list A list of pdisks to be rebuilt.
-S	Displays the adapter link status. -l lname The logical name of the adapter.
-U	Creates stand-alone physical disks. -z drive_lists A list of pdisks to be formatted to stand-alone disks.
-W	Reclaims cache storage. -l lname The logical name of the adapter. -o cmd_option The command options follow: 0 Queries to determine whether a reclaim operation is needed. 1 Queries to determine whether permission for unknown data loss is needed. 2 Performs reclaim cache storage. 3 Performs reclaim cache storage and allows unknown data loss.
-X	Changes adapter assignment. -l lname The logical name of the adapter. -o cmd_option The command options follow: 0 Displays only. 1 Preferred as primary adapter. 2 No preferred operating preferences. 3 Preferred as primary adapter. This value runs the cfgmgr command.
-Y	Resynchronizes array protection. -l lname The logical name of the array.

Exit Status

This command returns the following exit values:

Item	Description
0	The sisraidmgr command completed the operation successfully.
>0	The sisraidmgr command detected an error.

Security

Privilege Control: Only the root user and members of the system group should have execute (x) access to this command.

Examples

1. Display usage information:

```
# sisraidmgr -h
```

2. Views disk array configuration on a PCI-X SCSI RAID controller named sissas0:

```
# sisraidmgr -L -l sissas0 -j3
```

3. Prepares 512 byte formatted drives (hdisk3 and hdisk4) for use in a disk array:

```
# sisraidmgr -P -z 'hdisk3 hdisk4'
```

4. Creates a RAID 0 array with stripe size of 256K on the prepared disks (pdisk2 and pdisk5):

```
# sisraidmgr -C -r 0 -s 256 -z 'pdisk2 pdisk5'
```

5. Deletes the RAID array hdisk3 on controller sissas0:

```
# sisraidmgr -D -l sissas0 -d hdisk3
```

Files

Item	Description
/usr/bin/sisraidmgr	Contains the sisraidmgr command.

sissasraidmgr Command

Purpose

Maintains and uses a Serial Attached SCSI (SAS) Redundant Array of Independent Disks (RAID) controller.

Syntax

```
sissasraidmgr -A -l hdisk# [ -z pdisk [ -f ] ]  
sissasraidmgr -B -l adptr# -b ioa_opt [ -r raid_level ]  
sissasraidmgr -C [ -r raid_level -s stripe_size (in KB) -z pdisk_list ]  
sissasraidmgr -D -l adptr# [-d hdisk | -e serial_num ]  
sissasraidmgr -E -l adptr# [-d hdisk -o cmd_opt]  
sissasraidmgr -F -z pdisk_list  
sissasraidmgr -G -l hdisk# -r raid_level [ -s stripe_size (in KB) -z pdisk_list ]
```

```

sissasraidmgr -H [ -z pdisk_list ]
sissasraidmgr -I [ -z pdisk_list ]
sissasraidmgr -J -z drive_list -o cmd_opt
sissasraidmgr -L -l adptr# [ -a display_opt [ -v sisarray_opt -p pdisk_opt -j jbod_opt ] ]
sissasraidmgr -L -l hdisk# [ -a display_opt [ -v sisarray_opt -p pdisk_opt ] ]
sissasraidmgr -L -l pdisk# [ -p pdisk_opt ]
sissasraidmgr -M -l adptr# -o cmd_opt
sissasraidmgr -P -z drive_list ( pdisks | hdisks )
sissasraidmgr -Q -z pdisks } [ -o cmd_opt ]
sissasraidmgr -R -z pdisk_list
sissasraidmgr -S -l adptr# [ -o cmd_opt ]
sissasraidmgr -T -l adptr# [ -o cmd_opt ]
sissasraidmgr -T -l device# [ -o cmd_opt ]
sissasraidmgr -U -z pdisk_list
sissasraidmgr -W -l adptr# -o cmd_opt
sissasraidmgr -X -l adptr# -o cmd_opt
sissasraidmgr -Y -l hdisk#
sissasraidmgr -Z -l adptr# -o cmd_opt

```

Description

The `sissasraidmgr` command is used to create, delete, and maintain RAID arrays on a Peripheral Component Interconnect-X (PCI-X) or PCI Express (PCIe) SAS RAID controller.



Attention: See the *Power Systems SAS RAID Controllers for AIX* reference guide and become familiar with the storage management concepts before you run the **sissasraidmgr** command.



Attention: The *System Management Interface Tool (SMIT)* **smit sasdsm** fast path is the preferred method to manage a SAS RAID controller.



Attention: Service tasks require special training and must not be performed by nonservice personnel.

Flags

Item	Description
-A	Add a device to an existing array. The performance is not optimal when using this option because the included device does not contain parity, and the data is not restriped.
-l <i>lname</i>	The logical name of the array.
-z <i>pdisks</i>	The drives to be included.
-f	The option to force the include operation in the situation where the disks to be included might not be known; that is, they might be 0.

Item	Description
-B	Lists information about what the adapter supports. -l lname The logical name of the adapter. -b ioa_support_opt 1 Displays supported RAID levels for the lname option. This is the default option. 2 Displays supported stripe size for the lname and raid_level option. 3 Displays the minimum number of devices for the raid_level option. 4 Displays the maximum number of devices for the raid_level option. 5 Displays the minimum multiple number of devices for the raid_level option. 6 Displays supported migration RAID levels for the lname option. 7 Displays supported migration stripe size for the lname and raid_level options. 8 Displays the minimum number of migration include devices for the raid_level option. 9 Displays the maximum number of migration include devices for the raid_level option. 10 Displays minimum multiple migration include devices for the raid_level option. 11 Displays the minimum percentage of the total array capacity that is allowed in one tier for the raid_level option. 12 Displays the minimum number of devices per tier for the raid_level option. -r raid_level Shows supported stripe sizes for this RAID level.
-C	Creates a RAID array. -r raid_level { 0, 5, 6, 10 (RAID 1+0), 5T2, 6T2, or 10T2 } -s stripe_size (in KB) Specifies the stripe size. If not specified, the default (64 KB) is used. -z pdisk_list Lists pdisks to include in the new array. For example, 'pdisk2 pdisk3 pdisk4' must be connected to the same adapter.
-D	Deletes a RAID array. -l lname The logical name of the adapter. -d hdisk The name of the array to be deleted. -e serial_num The serial number of the array to be deleted. Use this option only if the array name is unknown.
-E	Manages HA access characteristics of a RAID array. -l lname The logical name of the adapter. -d hdisk The name of the array. -o cmd-opt The command options follow: 1 Displays the current and preferred HA access states. 2 Sets the preference to optimized on the lname option. 3 Sets the preference to nonoptimized on the lname option. 4 Clears preferences.

Item	Description
-F	Formats the pdisks for recovery (format RAID formatted disks). -z drive_list A list of pdisks to format.
-G	Migrates the RAID array to a new RAID level. -l lname The logical name of the array. -r raid_level { 0, 5, 6, 10 (RAID 1+0), 5T2, 6T2, or 10T2} -s stripe_size (in KB) Specifies the stripe size. If not specified, the default (64 KB) is used. -z pdisk_list A list of pdisks to be included in the new array, if any.
-H	Adds a hot spare device. -z pdisk_list A list of pdisks to be made hot spare devices.
-I	Removes a hot spare device. -z pdisk_list A list of pdisks to be removed from being hot spare devices.
-J	Optimizes JBOD workload. -z drive_list A list of JBOD hdisks to optimize. -o cmd_opt The command options: 1 Optimizes for the I/O response time. 2 Optimizes for the I/O operation per second.

Item	Description
-L	Lists advance function information.
-l lname	The device for which information is displayed. It can be a RAID adapter (<i>sisioa0</i>), a RAID array (<i>hdisk8</i>), or a physical disk (<i>pdisk5</i>).
-a display_opt	
0	Displays all configuration information for the lname option. This is the default option.
1	Displays only logical device information for the lname option.
2	Displays only physical device information for the lname option.
3	Displays only the physical device information for the lname option that is not under an adapter in the secondary mode.
-v sisarray_opt	
0	Displays all arrays. This is the default.
1	Displays only arrays that are candidates for the Delete Array option.
2	Displays only arrays that are candidates for the Rsync Protection option.
3	Displays only arrays that are candidates for including additional devices.
4	Displays only ODM arrays that have no adapter information.
5	Displays only arrays that are candidates for migration to a new RAID level.
-p pdisk_opt	
0	Displays all pdisks. This is the default.
1	Displays only pdisks that are candidates for the Prepare option.
2	Displays only pdisks that are candidates for the Start RAID option.
3	Displays only pdisks that are candidates for the Add Hot Spare option.
4	Displays only pdisks that are candidates for the Remove Hot Spare option.
5	Displays only pdisks that are candidates to be added to an existing array.
6	Displays only pdisks that are candidates for the Rebuild option.
7	Displays only pdisks that are candidates for the Recovery Format option.
8	Displays only ODM pdisks that have no adapter information.
9	Displays only pdisks that are candidates for the Unprepare option.
10	Displays only pdisks that, if prepared, would be candidates to be added to an existing array.
11	Displays only pdisks under their main path (primary or only path).
12	Displays only pdisks that are candidates for including during the migration of an existing array.
-j jbod_opt	
0	Displays no JBOD hdisks. This is the default.
1	Displays all JBOD hdisks.
2	Displays only JBOD hdisks that are candidates for the Prepare option.
3	Displays all JBOD devices.

Item	Description
-M	Maintains rechargeable battery. -l lname The logical name of the adapter. -o cmd_option The command options follow: 0 Displays rechargeable battery information. 1 Forces a rechargeable battery error. 2 Starts caching after concurrent battery replacement. 3 Queries candidates for concurrently starting batteries..
-P	Prepares devices; that is, creates array candidates physical disks. -z drive_list A list of either JBOD hdisks, pdisks, or both to become an array candidate.
-Q	Sets or clears pdisk error suppression attributes. -z pdisk_list A list of pdisks for attributes to be applied or cleared. -o cmd_option A 1-byte hexadecimal string that specifies which error suppression bits to turn on or off.
-R	Rebuilds devices; that is, reconstructs a degraded array. -z pdisk_list A list of pdisks to be rebuilt. -o cmd_opt Command option for adapter: 0 Displays HA link status. This is the default. 1 Displays HA and AWC link status.
-S	Displays the adapter link status. -l lname The logical name of the adapter.
-T	Displays SAS path information for the adapter. -l lname The logical name of the adapter. -o cmd_opt The command option for the adapter follow: 0 Displays the summary path window. This is the default. 1 Displays all path information for all attached devices. 2 Graphically displays paths for all attached devices. 16 Displays I/O Adapter SAS addresses.
-T	Displays SAS path information for the attached devices. -l lname The logical name of the device (pdisk or hdisk). -o cmd_opt The command option for the adapter follow: 0 Graphically displays path information for device. 1 Displays path information data for a selected device.
-U	Creates stand-alone physical disks. -z drive_lists A list of pdisks to be formatted to stand-alone disks.

Item	Description
-W	Reclaims cache storage. -l lname The logical name of the adapter. -o cmd_option The command options follow: 0 Queries to determine whether a reclaim operation is needed. 1 Queries to determine whether permission for unknown data loss is needed. 2 Performs reclaim cache storage. 3 Performs reclaim cache storage, and allows unknown data loss.
-X	Changes adapter assignment. -l lname The logical name of the adapter. -o cmd_option The command options follow: 0 Displays only 1 Preferred as primary adapter. 2 No preferred operating preferences. 3 Preferred as primary adapter. This value runs the cfgmgr command. 4 Displays AWC preferred role information. 10 Sets the dual initiator mode to be the default. 11 Sets the dual initiator mode to the JBOD HA single path. 256 Clears HA access states. 512 Preserves HA Access states. 1024 Enables the default behavior of the IOA cache. 2048 Disables the IOA cache. Note: The clear, preserve, enable, and disable options can be paired (ORed) with options 1, 2, or 3, or they can be used as stand-alone options.
-Y	Resynchronizes array protection. -l lname The logical name of the array.
-Z	Shows the SAS controller physical resources. -l lname The logical name of the adapter. -o cmd_option The command options follow: 0 Shows the physical location. This is the default. 1 Shows physical information. Note: Enter the same options as the -L flag to filter the output.

Exit Status

This command returns the following exit values:

Item	Description
0	The sissasraidmgr command completed the operation successfully.
>0	The sissasraidmgr command detected an error.

Security

Privilege Control: Only the root user and members of the system group should have execute (x) access to this command.

Examples

1. Displays usage information:

```
# sissasraidmgr -h
```

2. Views disk array configuration on a SAS RAID controller named sissas0:

```
# sissasraidmgr -L -l sissas0 -j3
```

3. Prepares JBOD drives (hdisk3 and hdisk4) for use in a disk array:

```
# sissasraidmgr -P -z 'hdisk3 hdisk4'
```

4. Creates a RAID 0 array with a stripe size of 256 KB on the prepared disks (pdisk2 and pdisk5):

```
# sissasraidmgr -C -r 0 -s 256 -z 'pdisk2 pdisk5'
```

5. Deletes the RAID array hdisk3 on controller sissas0:

```
# sissasraidmgr -D -l sissas0 -d hdisk3
```

6. Optimizes the RAID array hdisk1 on sissas2, which is also the primary controller:

```
# sissasraidmgr -E -l sissas2 -d hdisk1 -o 2
```

7. Optimizes hdisk2 on sissas3, which is the secondary controller:

```
# sissasraidmgr -E -l sissas2 -d hdisk2 -o 3
```

8. Show SAS physical paths to a drive pdisk3:

```
# sissasraidmgr -T -l pdisk3 -o 1
```

Files

Item	Description
/usr/bin/sissasraidmgr	Contains the sissasraidmgr command.

size Command

Purpose

Displays the section sizes of the Extended Common Object File Format (XCOFF) object files.

Syntax

```
size [ -d | -o | -x ] [ -f ] [ -V ] [ -X {32 | 64 | 32_64 | d64 | any} ] [ File ... ]
```

Description

The **size** command writes to standard output the number of bytes required by all sections, along with their sum for each XCOFF file. If the **-f** flag is specified, the section name follows the section size.

Note: When no file is passed as an input to the **size** command, the **a.out** file is considered as the default.

Flags

The output is in decimal notation unless you change the output with the following flags:

Item	Description
-d	Writes in decimal notation.
-f	Writes the section name in parenthesis following the section size.
-o	Writes in octal notation.
-x	Writes in hexadecimal notation.
-X mode	Specifies the type of object file size should examine. The <i>mode</i> must be one of the following: 32 Processes only 32-bit object files 64 Processes only 64-bit object files 32_64 Processes both 32-bit and 64-bit object files d64 Examines discontinued 64-bit XCOFF files (magic number == U803XTOCMAGIC). any Processes all of the supported object files. The default is to process 32-bit object files (ignore 64-bit objects). The <i>mode</i> can also be set with the OBJECT_MODE environment variable. For example, OBJECT_MODE=64 causes size to process any 64-bit objects and ignore 32-bit objects. The -X flag overrides the OBJECT_MODE variable.
-V	Prints the version number of the size command.

Examples

1. To display the size of the **a.out** file in decimal, enter:

```
size
```

This displays the size in bytes of the executable **a.out** file. The size of each section of the object file is given, followed by the total:

```
3720 + 1752 + 4152 = 9624
```

2. To display the size of an object file in octal, enter:

```
size -o driver.o
```

This displays the size of the **driver.o** object file in octal.

3. To display the size of several object files in hexadecimal, enter:

```
size -x *.o
```

This displays in hexadecimal the size of each file ending with **.o** in the current directory.

skctl Command

Purpose

Handles alterations in the storage protection keys attributes.

Syntax

skctl [-D]

skctl [-u] <nukeys/off>] [-k on/off/default]

skctl [-v] [now/default/boot]

Description

The **skctl** command is a privileged command used on a system that supports storage protection keys. The **skctl** command can change the number of user-space storage keys, disable user-space storage keys, enable/disable kernel storage key state, and display the default, current, and next boot storage keys attributes.

Note: You must run **/usr/sbin/bosboot** command after changing the storage keys attributes, and then reboot the system for the change to take effect.

Flags

Item	Description
-u	Alters the number of user-space keys or disables user-space keys. The flag must be off or a number between 2 and the maximum number of hardware storage keys.
-k	Enables/disables kernel keys.
-v	Displays the default, current, and next boot storage keys attributes.
-D	Resets the storage protection keys attributes to default.

skulker Command

Purpose

Cleans up file systems by removing unwanted files.

Syntax

skulker

Description

Attention: Because the **skulker** command is run by a root user, and its whole purpose is to remove files, it has the potential for unexpected results. Before installing a new **skulker** command, test any additions to its file removal criteria by running the additions manually using the **xargs -p** command. After you have verified that the new **skulker** command removes only the files you want removed, you can install it.

The **skulker** command is used for periodically purging obsolete or unneeded files from file systems. Candidate files include files in the **/tmp** directory, files older than a specified age, and the following file types: ***.bak**, **a.out**, **core**, **proof**, **galley**, **...***, **ed.hup**, and files that are more than one day old.

The **skulker** command is normally invoked daily, often as part of an accounting procedure run by the **cron** command during off-peak periods. Modify the **skulker** command to suit local needs following the patterns shown in the distributed version. Local users should be made aware of the criteria for automatic file removal.

The **find** command and the **xargs** command form a powerful combination for use in the **skulker** command. Most file selection criteria can be expressed conveniently with **find** expressions. The resulting file list can be segmented and inserted into **rm** commands using the **xargs** command to reduce the overhead that would result if each file were deleted with a separate command.

slattach Command

Purpose

Attaches serial lines as network interfaces.

Syntax

```
/usr/sbin/slattach TTYName [ BaudRate DialString [ DebugLevel ] ]
```

Description

The **/usr/sbin/slattach** command assigns a TTY line to a network interface.

The **slattach** command is run by the **/etc/rc.net** file during system startup to automatically configure any Serial Line Internet Protocol (SLIP) network interfaces defined by the System Management Interface Tool (SMIT). SLIP interfaces can also be configured manually as shown in the examples section.

For a directly connected SLIP interface, broken connections are retried automatically without manual intervention. For a SLIP interface connected by modem, broken connections must be manually redialed. If a user supplies a dial string in the **slattach** command line, the user must re-enter the command and dial string to restore a broken connection.

To detach the interface, run the **ifconfig Interface down** command after terminating the **slattach** command. The *Interface* parameter is the name shown by the **netstat** command.

If configuring a slip interface from the command line, the **/usr/sbin/ifconfig** command must be invoked for the slip interface with the appropriate parameters and the slip tty line discipline must also be available in order for this command to succeed. To check if the slip tty line discipline is already loaded, run the command `strinfo -m | grep slip`. If no output is shown, the module has not yet been loaded. Load the module by issuing the command `strload -m /usr/lib/drivers/slip`.

Note:

1. After the SLIP interface has been configured with **ifconfig**, any user who has permission on the TTY may issue the **slattach** command.
2. You must configure the tty devices used by the **slattach** command before establishing a connection. You may also need to make an entry for the tty device in the BNU **/usr/lib/uucp/Devices** file.
3. Sample shell script, **/usr/sbin/slipcall**, provides a simplified interface for invoking **slattach** and connecting to remote systems. **slipcall** is useful for connecting to dial-in SLIP networks that require a user to login before activating the SLIP tty line discipline. The basic configuration of **slipcall** will connect to other operating systems with **sliplogin** configurations and derive the local and remote internet addresses and network mask assigned by the called system. It then configures the local interface with the remote system's specified values.

Parameters

Item	Description
<i>BaudRate</i>	Sets the speed of the connection. The default speed is 9600.

Item	Description
<i>DebugLevel</i>	Sets the level of debug information desired. A number from 0 through 9 may be specified. A value of 0 specifies no debug information; a value of 9 specifies the most debug information. The default value is 0.
<i>DialString</i>	Specifies a string of expect/respond sequences using the Basic Networking Utility (BNU)/AIX to AIX Copy Program (UUCP) chat syntax.
<i>TTYName</i>	Specifies a TTY line. This string is in the form <code>ttyxx</code> or <code>/dev/ttyxx</code> .

Examples

1. To attach the SLIP network interface to the `tty1` port with a direct connection, issue the following command:

```
slattach /dev/tty1
```

This command attaches `tty1` to a network interface to be used by the SLIP.

2. To attach the SLIP network interface to `tty1` using a modem connection, issue the following command:

```
slattach /dev/tty1 9600 ""AT OK \pATF1 OK \pATDT34335 CONNECT""
```

Files

Item	Description
/etc/uucp/Devices	Lists definitions of devices used for remote connections.

sleep Command

Purpose

Suspends execution for an interval.

Syntax

sleep *Seconds*

Description

The **sleep** command suspends execution of a process for at least the interval specified by the *Seconds* parameter. The amount of time specified in the *Seconds* parameter can range from 1 to **MAXINT** (2,147,483,647) seconds.

Exit Status

This command returns the following exit values:

Item	Description
0	The execution was successfully suspended for at least <i>Seconds</i> seconds, or a SIGALRM signal was received.
>0	An error occurred.

Examples

1. To run a command after a certain amount of time has passed, enter:

```
(
echo "SYSTEM SHUTDOWN IN 10 MINUTES!" | wall
sleep 300; echo "SYSTEM SHUTDOWN IN 5 MINUTES!" | wall
sleep 240; echo "SYSTEM SHUTDOWN IN 1 MINUTE!" | wall
sleep 60; shutdown
)&
```

This command sequence warns all users 10 minutes, 5 minutes, and 1 minute before the system is shut down.

2. To run a command at regular intervals, enter:

```
while true
do
date
sleep 60
done
```

This shell procedure displays the date and time once a minute. To stop it, press the Interrupt key sequence.

slibclean Command

Purpose

Removes any currently unused modules in kernel and library memory.

Syntax

slibclean

Description

The **slibclean** command unloads all object files with load and use counts of 0. It can also be used to remove object files that are no longer used from both the shared library region and in the shared library and kernel text regions by removing object files that are no longer required.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Files

Item	Description
/usr/sbin/slibclean	Contains the slibclean command.

sliplogin Command

Purpose

Converts a standard-input terminal line into a Serial Line Internet Protocol (SLIP) link to a remote host.

Syntax

sliplogin [*LoginName*]

Description

The **sliplogin** command configures a standard-input terminal line into a Serial Line Internet Protocol (SLIP) link to a remote host; that is, the command attaches a serial line network interface.

Note: User requires root authority to attach a network interface.

The **sliplogin** command searches the **/etc/slip.hosts** file for a *loginname* entry that matches the value of the *LoginName* parameter. If a matching entry is found, **sliplogin** configures the line appropriately for SLIP (that is, for 8-bit transparent input/output) and converts it to SLIP line discipline. Then, **sliplogin** invokes the applicable login shell script which initializes the SLIP interface with the local and remote Internet Protocol (IP) addresses, netmask, and optional arguments associated with the *loginname* entry in the **/etc/slip.hosts** file.

The usual initialization script file is **/etc/slip.login**. However, in order to accommodate special initialization needs of a particular host, a script file named **/etc/slip.login.userlogin** (where *userlogin* corresponds to the *loginname* entry in the **/etc/slip.hosts** file) can be created. The **sliplogin** command uses the **/etc/slip.login.userlogin** script file when it exists, instead of the **/etc/slip.login** script file.

To deinitialize the SLIP interface, the **sliplogin** command uses either the **/etc/slip.logout** script file or the **/etc/slip.logout.userlogin** script file, if one of them exists, with preference given to the latter. The **/etc/slip.logout** script file is given the same arguments as the **/etc/slip.login** script file; the **/etc/slip.logout.userlogin** script file is given the same arguments as the **/etc/slip.login.userlogin** script file. In its default form, the **/etc/slip.logout** script file deletes all routes through the network interface for the specified SLIP unit. Additional processes to be done when the SLIP interface is disconnected can be added to either logout script file.

Note:

1. The interface automatically deactivates when the remote connection terminates or when the **sliplogin** command dies.
2. Use the **slattach** command to access a remote system that has a SLIP link configured. Use the sample shell script file **/usr/sbin/slipcall** to invoke the **slattach** command with the proper parameters needed to call a remote system and configure the local interface with the appropriate values assigned by the remote system.
3. When using **sliplogin** as a user's login shell on a tty device, then this tty port used needs to be enabled for login. (This differs from the configuration when using **slattach** instead of **sliplogin** as a SLIP server process.

/etc/slip.hosts File

The **/etc/slip.hosts** file is the configuration file containing the names of preconfigured sliplogin users and the IP addresses to be assigned to the local and remote interface when the user logs in. **sliplogin** searches this file for matching *LoginName* entries. This file has the following format:

- Comments (lines starting with a #) and blank lines are ignored.
- Other lines must start with a *loginname* argument, and the fields should contain whatever is appropriate for the **slip.login** file that is executed for that name.
- Arguments are separated by white space and follow normal sh(1) quoting conventions. However, the *loginname* argument cannot be quoted. Usually lines have the following form:

```
loginname local_address remote_address netmask opt_args
```

where *local_address* and *remote_address* are the IP host names or addresses of the local and remote ends of the SLIP line, and *netmask* is the appropriate IP netmask. These arguments are passed directly to the **ifconfig** command. *Opt_args* are optional arguments used to configure the line.

- This implementation of **sliplogin** allows the **/etc/slip.hosts** file to contain multiple entries for a single SLIP user with differing addresses. This enables multiple SLIP interfaces to be activated by the **sliplogin** command for the same user name. When user entries are retrieved from the **/etc/slip.hosts** file, only entry addresses meeting the following criteria are selected.

The entry is ignored if a **slip.hosts** entry specifies a local address which is already in use on another non-SLIP interface on the local system.

The entry is ignored if the remote address specified in an **/etc/slip.hosts** entry is already in use on any other interface.

/etc/slip.login File

The **/etc/slip.login** or **/etc/slip.login.userlogin** file is the setup script invoked by the **sliplogin** command to initialize the user's network interface. The **/etc/slip.login.userlogin** file is invoked if it exists, where the value of the *LoginName* parameter of the **sliplogin** command corresponds to a loginname entry in the **/etc/slip.hosts** file. If this file cannot be accessed, the **/etc/slip.login** file is invoked instead. The login script file contains the following parameters:

Item	Description
<i>slipunit</i>	Specifies the unit number of SLIP interface assigned to this line. For example, 0 for sl0 (sl0 is s, lowercase L, zero.)
<i>speed</i>	Specifies the speed of the line.
<i>args</i>	Specifies the arguments from the /etc/slip.hosts file entries, in order, starting with <i>loginname</i> .

/etc/slip.logout File

The **/etc/slip.logout** or **/etc/slip.logout.userlogin** file is the setup script invoked by **sliplogin** to deinitialize the user's network interface. The **/etc/slip.logout.userlogin** file is invoked if it exists, where the value of the *LoginName* parameter of **sliplogin** corresponds to a loginname entry in the **/etc/slip.hosts** file. If this file cannot be accessed, the **/etc/slip.logout** file is invoked instead.

Flags

Item	Description
</dev/ttyx	Redirects the command to the tttyx device if the user is already logged into a ttty device and wants to configure their terminal as a SLIP line.

Parameters

Item	Description
<i>LoginName</i>	Specifies the desired login name. The default is the current login name.

Examples

The normal use of the **sliplogin** command is to create an **/etc/passwd** entry for each legal, remote SLIP site with **sliplogin** as the shell for the entry. For example,

```
foo:!:2010:1:slip line to foo:/tmp:/usr/sbin/sliplogin
```

An entry must then be added to the **/etc/slip.hosts** file. The entry should resemble the following example:

```
foo 1.1.1.1 1.1.1.2 0xffffffff00 normal
```

where *loginname* = *foo*, *local_address* = 1.1.1.1, *remote_address* = 1.1.1.2, *netmask* = 0xffffffff00, and *opt_args* = *normal*. (The optional argument *normal* indicates which SLIP mode to activate.)

Diagnostics

The **sliplogin** command logs various information to the system log daemon (**syslogd**). The messages are listed here, grouped by severity levels.

Error Severity	
Message	Description
ioctl (TCGETS): <i>reason</i>	The ioctl subroutine failed to get the line parameters for the reason indicated.
ioctl (TCSETS): <i>reason</i>	The ioctl subroutine failed to set the line parameters for the reason indicated.
ioctl (TIOCGTD): <i>reason</i>	The ioctl subroutine failed to get the current tty discipline for the reason indicated.
/etc/slip.hosts: <i>reason</i>	The /etc/slip.hosts file could not be opened for the reason indicated.
Check of flags for interface xxx failed. Errno is reason.	An attempt to check the status of the indicated interface to avert possible addressing conflicts failed for the reason indicated in the errno global variable.
Access denied for user - no /etc/slip.login[<i>userlogin</i>] file.	No /etc/slip.login or /etc/slip.login.userlogin script file could be found.
Access denied for user - no /etc/slip.hosts entries available.	No loginname entry in the /etc/slip.hosts file matched the <i>LoginName</i> value specified in the command.
Access denied - getlogin returned 0.	The user issuing the sliplogin command does not have a password entry in the /etc/passwd file.
Logout script failed: exit status xxx from /etc/slip.logout[<i>userlogin</i>]	An attempt to run the /etc/slip.logout or /etc/slip.logout.userlogin script file failed with the indicated exit status.
No SLIP interface for ttyx. Errno is reason.	No SLIP interface could be located for the ttyx device for the reason indicated in the errno global variable. Try either running the ifconfig slx up command or using SMIT to add a network interface for the tty device.
Open /dev/null: <i>reason</i>	An attempt to open the /dev/null device failed for the reason indicated.
/etc/slip.logout file not found	The /etc/slip.logout file could not be located.
sliplogin: cannot add SLIP discipline to ttyx	No SLIP interface exists for the ttyx device. Try either running the ifconfig slx up command or using SMIT to add a network interface for the tty device.
SLIP discipline removal from tty failed. Errno is reason.	An attempt to remove the SLIP discipline from the tty device failed for the reason indicated in the errno global variable.

Error Severity (<i>continued</i>)	
Message	Description
tcgetattr: <i>reason</i>	An attempt to read the current attributes of the tty device failed for the reason indicated.
<i>userlogin</i> login failed: exit status <i>xxx</i> from <code>/etc/slip.login[<i>userlogin</i>]</code>	A system call to execute the /etc/slip.login or /etc/slip.login.userlogin script file failed with the indicated exit status.

Information Severity	
Message	Description
Attaching SLIP unit <i>xxx</i> for <i>userlogin</i> on <i>ttyx</i> .	The sliplogin command found a loginname entry in the /etc/slip.hosts file that matched the <i>LoginName</i> value specified in the command, invoked the applicable /etc/slip.login or /etc/slip.login.userlogin file, and is now attaching the indicated network interface.
Closed <i>userlogin</i> SLIP unit <i>xxx</i> (signal)	The indicated SLIP unit for the indicated <i>userlogin</i> was closed because the sliplogin command terminated due to a signal.

Notice Severity	
Message	Description
Attaching SLIP unit <i>xxx</i> for <i>userlogin</i> .	The indicated SLIP unit has been successfully attached for the indicated <i>userlogin</i> .

Files

Item	Description
/etc/slip.hosts	The configuration file that contains the names of preconfigured sliplogin users and the IP addresses to be assigned to the local and remote interface when the user logs in.
/etc/slip.login or /etc/slip.login.userlogin	The setup script invoked by the sliplogin command to initialize the user's network interface.
/etc/slip.logout or /etc/slip.logout.userlogin	The setup script invoked by the sliplogin command to deinitialize the user's network interface.

slocal Command

Purpose

Processes incoming mail.

Syntax

slocal [**-verbose** | **-noverbose**] [**-debug**]

Description

The **slocal** command performs a set of actions each time a message is sent to the user. The **slocal** command is not started by the user. The **slocal** command is called by the **sendmail** command.

The **sendmail** command starts the **slocal** command upon encountering the following line in the **\$HOME/.forward** files:

```
/usr/lib/mh/slocal
```

For each incoming message, the **slocal** command performs the actions specified in the **.maildelivery** file. If the **slocal** command cannot find the **\$HOME/.maildelivery** file, the **slocal** command uses the **/etc/mh/maildelivery** default file. If the delivery request fails, the **slocal** command delivers the message to the **/usr/mail/\$USER** file.

Flags

Item	Description
-debug	Provides information for debugging.
-help	Lists the command syntax, available switches (toggles), and version information. Note: For Message Handler (MH), the name of this flag must be fully spelled out.
-noverbose	Does not display information as the system executes commands in the .maildelivery file. This flag is the default.
-verbose	Displays information as the system executes commands in the .maildelivery file.

Files

Item	Description
/usr/lib/mh/mtstailor	Contains MH command definitions.
/etc/mh//.maildelivery	Contains the default MH instructions for local mail delivery.
\$HOME/.maildelivery	Provides the user with MH instructions for local mail delivery.
\$HOME/.forward	Contains either the line that starts the slocal command or a path to forward mail.
/etc/mh/mh_profile	Contains parameters that customize the MH package.

slp_srvreg Command

Purpose

Manages a service location protocol (SLP) service agent.

Syntax

slp_srvreg -t *servicetype* **-u** *URL* [**-a** *attribute*] [**-l** *lifetime*] [**-s** *scopes*] [**-T** *IPAddress*] [**-p** *port*] [**-U**] [**-v**] [**-b** *debuglevel*] [**-6**]

slp_srvreg -d *URL* [**-s** *scopes*] [**-T** *IPAddress*] [**-p** *port*] [**-v**] [**-b** *debuglevel*] [**-6**]

slp_srvreg -D [**-v**] [**-b** *debuglevel*] [**-p** *port*]

slp_srvreg -k [**-v**] [**-b** *debuglevel*]

slp_srvreg -h

Description

The **slp_srvreg** command manages the service location protocol (SLP) service agent. The **slp_srvreg** command is used to register services for a specified URL with an attribute list in a given scope. The *servicetype* specified with the **-t** flag will override any service type expressed in the URL with the **scheme** service.

To register a service, use the **slp_srvreg** command with the **-u** flag to specify the URL to register.

To deregister a service, use the **slp_srvreg** command with the **-d** flag to specify the URL to deregister.

For both registration and deregistration, use the **-T** flag to specify an IP address to which the registration request will be sent to. If you specify the IP address of the local host (e.g. 127.0.0.1) or if you do not use the **-T** flag, the registration of the service URL is processed locally.

You must specify the **slp_srvreg** command with the **-D** flag to run **slp_srvreg** as a daemon. The **slp_srvreg** command with the **-k** flag kills the **slp_srvreg**.

Restriction: Do not run more than one **slp_srvreg** daemon on the same machine.

Use the **-p** flag to make the **slp_srvreg** agent running as daemon listen on a user specified port instead of the default port number 427. When registering or de-registering with a port specified in the **-p** flag of the **slp_srvreg**, only the service agents or directory agents listening on this port will accept the registration or deregistration.

Requirement: The **-t** and **-u** flags are mandatory for registration.

SLP clients must not expect the SLP service agent to return attribute values using the same case as used during the registration. For example, if a client registers a service with *attribute=true*, a query for the attribute might respond with *attribute=TRUE*. Any client seeking this information must handle the attribute in a case-insensitive manner.

Note: When the command **slp_srvreg -D -b debuglevel** is used with a debuglevel greater than zero, then **slp_srvreg** is not run as a daemon.

Flags

Item	Description
-a attribute	Specifies a comma-separated list of attributes for the services to be registered.
-d URL	Specifies the URL for the service to be deregistered.
-D	Specifies to run as a daemon.
-k	Kills the slp_srvreg daemon.
-l lifetime	Specifies the time after which the service registration needs be renewed. The value of the <i>lifetime</i> attribute is specified in number of seconds.
-p port	Specifies the port to listen to when running as a daemon. If you do not specify the -p flag, the default port 427 is used. If the slp_srvreg daemon is listening on a port other than the standard port, the user agent uses this flag to send the new registration data to the correct listener.
-s scopes	Specifies the scopes of the services to be registered.
-t servicetype	Specifies the service type of the service URL.
-T IPAddress	Specifies the IP address that the service registration needs to be sent to.
-u URL	Specifies the URL for the service to be registered.

Item	Description
-U	Replaces an existing registration.
-v	Specifies verbose output.
-b level	Specifies the debuglevel (from 0 to 7). A three-bit mask is used: • - 0b001 = 1 is to see the important debug information (errors and main program steps) • - 0b010 = 2 is to see the detailed debug information (detailed program steps) • - 0b100 = 4 is to see the start and stop traces of all functions.
-6	Specifies that IPv6 must be used to resolve any hostname used in URL; if omitted, IPv4 is used to resolve the host names.
-h	Display the help: Command Usage.

Examples

1. To run the command as a daemon on the default SLP port 427, enter the following command:

```
# slp_srvreg -D
```

2. To register the service with the `service:pop3://mail.ibm.com` URL and the `user=Tom, Richard` attributes for two days, enter the following command:

```
# slp_srvreg -v -a "user=Tom, Richard" -u "service:pop3://mail.ibm.com"
-t "service:pop3" -l 172800
```

3. To register the service with the `service:pop3://mail.ibm.com` URL and the `user=Tom, Richard` attributes for two days for the local host, enter the following command:

```
# slp_srvreg -a "user=Tom, Richard" -u "service:pop3://mail.ibm.com"
-t "service:pop3" -l 172800 -T 127.0.0.1
```

4. To register the service with the `service:pop3://mail.ibm.com` URL and the `user=Tom, Richard` attributes for two days for the local host, enter the following command:

```
# slp_srvreg -a "user=Tom, Richard" -u "service:pop3://mail.ibm.com"
-t "service:pop3" -l 172800 -T 127.0.0.1
```

5. To deregister the service with the `service:pop3://mail.ibm.com` URL with important and detailed debug traces (0b011 = 3), enter the following command:

```
# slp_srvreg -d "service:pop3://mail.ibm.com" -t "service:pop3" -b 5
```

6. To kill the **slp_srvreg** daemon, enter the following command:

```
# slp_srvreg -k
```

smbcd Daemon

Purpose

Processes General Security Services API (GSSAPI) authentication requests for Server Message Block (SMB) client file system.

Syntax

```
/usr/sbin/smbcd
```

Description

The SMB client file system in the AIX operating system requires Kerberos-based GSSAPI to start the user-authenticated session by using the SMB protocol version 2.1. In the AIX operating system, the GSSAPI is provided by a Userspace Library in the IBM Network Authentication Service (NAS) version 1.16.1.0, or later fileset. This fileset is included in AIX Expansion Pack.

The **smbcd** daemon authenticates the SMB client file system to the required Kerberos-based GSSAPI and later deletes the established authentication context for the SMB client file system based on the type of request. The SMB client file system sends requests to the SMB server to access the remote shares (files and directories) during the mount or unmount operations and to reauthenticate an existing session at regular intervals or after a session expiry.

When the `smbc.rte` fileset is installed, the **smbcd** daemon is configured to operate with the System Resource Controller (SRC) master program. The SRC commands can stop and start the **smbcd** daemon. The **smbcd** daemon is started automatically when the logical partition starts. If the **smbcd** daemon is killed, the SRC master program restarts the **smbcd** daemon automatically. If the **smbcd** daemon is not running, you cannot authenticate or reauthenticate the session with the SMB server. All further attempts to access files in the SMB client file system in an unauthenticated session fails. You can start the **smbcd** daemon directly by running the **startsrc** command.

When the **smbcd** daemon starts, the **smbcd** daemon parses the `/etc/smbc/smbctune.conf` file and updates the kernel with the latest values of the tunable parameters from the file. Thus, the tunable parameter values are preserved between system restart operations.

You can determine the number of concurrent authentications that can be performed by using the **smbc_max_concurrent_mount** tunable parameter in the **smbctune** command. You can query the **smbcd** daemon status to determine the basic information such as the process ID, state, and subsystem.

Files

/var/adm/smbc/unixsock

UNIX socket file that is used for inter-process communication between the **smbcd** daemon and the `smbclient` kernel extension.

/var/adm/smbc/smbcgssd_krb5cc

Directory that contains the cache files of Kerberos credentials for various users.

/var/locks/LCK..smbcd

Lock file for the **smbcd** daemon. The lock operation ensures that the **smbcd** daemon does not run multiple times on repeated invocation.

smbcstat Command

Purpose

Displays statistics information for the mount operation of the Server Message Block (SMB) client file system.

Syntax

```
/usr/sbin/smbcstat [-l] [-s <remote share> | -d <mount point>]
```

Description

The **smbcstat** command displays statistics information for the mount operation of the SMB client file system. You can view statistics information for a specific mounted remote directory, a specific mount point, or all mount points in the logical partition. If you do not specify any options, the **smbcstat** command lists statistics information for all SMB mount points in the logical partition.

Flags

-d mount point

Lists statistics information for the SMB mount point that is specified as an argument to this option.

-l

Lists statistics information for all SMB mount points.

-s remote share

Lists mount statistics information for the remote mount point that is specified as an argument to this option.

Exit status

0

Indicates success.

>0

Indicates error.

Example

- To list statistics information for all the mount points in the SMB client file system, enter the following command:

```
smbcstat -l
```

For SMB protocol version 2.1, the command displays an output similar to the following example:

```
Share Name      : PERFLOGS
Protocol Used   : smb2.1
Port            : 445
Mount Name      : /mnt2
Server Name     : 9.126.144.132
User Name       : braz21pa2usr1
Domain Name     : SMB_30_2.DEV
Signing State   : required
Group Id        : 0
User Id         : 0
Mode            : 0
SPN              : cifs/llm141.in.ibm.com
```

```
-----
Total Number of SMB Client Filesystem Mounts : 1
```

For SMB protocol version 3.0.2, the command displays an output similar to the following example:

```

Share Name      : PERFLOGS
Protocol Used   : smb3.0.2
Port           : 445
Mount Name      : /mnt1
Server Name     : 9.126.144.132
User Name       : braz21pa2usr1
Domain Name     : SMB_30_2.DEV
Signing State   : required
Group Id        : 0
User Id         : 0
Mode           : 0
Secure Negotiate: True
Encryption state: True
SPN             : cifs/llm141.in.ibm.com
-----
Total Number of SMB Client Filesystem Mounts : 1

```

smbctune Command

Purpose

Sets and displays the tunable parameters for the mount operations of the Server Message Block (SMB) client file system.

Syntax

```
smbctune [ -l | -f ] | [ [ -p ] -s smbc_tunable_parameter1=value smbc_tunable_parameter2=value ... ]
```

Description

You can run this command with the **-s** flag to set new values for the tunable parameters by specifying the tunable parameters as an argument. When you set new values for the tunable parameters, the **smbctune** command updates the kernel with new values of the tunable parameters. If you use **-p** flag along with the **-s** flag, the **smbctune** command updates the `/etc/smbc/smbctune.conf` file with new values to make tunable parameters persistent. This file is used when logical partition starts and initializes the SMB client subsystem.

When you do not specify the **pver**, **signing**, **secure_negotiate**, and **encryption** parameters with the **mount** command by using the **-o** flag, default values are used from the tunable parameter values in the kernel that are read from the `smbctune.conf` file or set by using the **smbctune** command.

Flags

-f

Displays the tunable parameter values that are specified in the `/etc/smbc/smbctune.conf` file.

-l

Displays the tunable parameter values that are specified in the kernel.

-p

Makes the tunable parameter values persistent by writing values to `/etc/smbc/smbctune.conf` file. Should be used only along with **-s**.

-s *smbc_tunable_parameter=value*

Sets specified values to specified tunable parameters. You can update multiple tunable parameters in a single command. You must not specify any negative values.

smbc_max_concurrent_mount

Specifies the maximum number of concurrent authentications that the **smbcd** daemon can perform at a time. This value indicates the threshold limit for the number of concurrent mount operations that can be processed at a time. When you set a new value to this tunable parameter, the **smbctune** command refreshes the **smbcd** daemon so that the corresponding value in the

userspace daemon changes accordingly. Valid values for this tunable parameter can be in the range 0-INT_MAX-1. The default value is 0.

smbc_request_timeout

Specifies the amount of time, in seconds, to wait before a request from the SMB client system to the SMB server times out. When this tunable parameter is set to 0, the requests do not expire on the SMB client system. Valid values for this tunable parameter can be in the range 0-INT_MAX-1. The default value for this tunable parameter is 0.

smbc_max_connections

Specifies the maximum number of SMB client connections that can exist with any number of SMB servers. When this tunable parameter is set to 0, the SMB client system can establish unlimited number of connections with the SMB servers. Valid values for this tunable parameter can be in the range 0-INT_MAX-1. This tunable parameter is set to 0 by default.

smbc_lookup_cache_size

Specifies the cache memory size of the SMB client lookup file. This lookup file keeps N file identifiers open so that these identifiers can be reused for lookup operations. The contents of this lookup file is similar to a Least Recently Used (LRU) cache memory that is implemented for each mount operation. The size of cache memory is same for each mount operation of the SMB client systems. The size of the cache memory cannot be a negative value. Valid value for this tunable parameter can be in the range 0-100. The default value is 32 cache entries. You can set a higher value if less number of SMB client systems are connecting to the SMB server.

smbc_krb5_lifetime

Specifies the token expiry time. The SMB client system and the SMB server communicate with each other by using tokens. This token is valid for a specific period. Based on the default values of Kerberos authentication, the token is valid for 10 hours. After that time period, the authenticated session between the SMB client system and SMB server expires and you cannot perform any other operation on the mounted remote share. Token expiry time is saved both in the cache memory of the SMB client credentials and in the token. The duration of the token can be renewed by reauthenticating to the SMB server. You can set this tunable parameter to a value less than 10 hours so that you can reauthenticate to the SMB server before 10 hours. Valid values for this tunable parameter can be in the range 0-INT_MAX-1. The default value is 0.

smbc_krb5_renew_till

Specifies the maximum time in seconds that a token can be reauthenticated to renew the token duration. After this period, the existing token context is deleted, and a new authenticated token context is created and used. Valid values for this tunable parameter can be in the range 0-INT_MAX-1. The default value is 0. Kerberos authentication follows its own default values.

smbc_file_lease_enable

Enables the leasing function in the SMB client file system. By default, this tunable parameter is enabled and is applicable for each SMB client file system.

The SMB client file system can use the file leasing function only when the file leasing function is enabled in the SMB client and in the SMB server systems, and when the SMB server supports the file leasing function. The file leasing function is preferred over the oplock function. The valid values are 1 and 0. You can change the value of this tunable parameter only when SMB client mount points are not available on the SMB client system.

smbc_oplock_enable

Enables the opportunistic locks (oplock) function for the SMB client system. By default, this tunable parameter is enabled and is applicable for each SMB client system.

The SMB client file system can use the oplock function only when the oplock function is enabled in the SMB server and in the SMB client system, and when the file leasing function is disabled in the SMB server or SMB client system. The valid values are 1 and 0. You can change the value of this tunable parameter only when SMB client mount points are not available on the SMB client system.

Note: When the SMB server or the SMB client system disables the file leasing function and the oplock function, the SMB client file system uses direct I/O operations.

smbc_protocol_version

Specifies the version of the SMB protocol that is used to communicate with the SMB server. The valid values are 2.1, 3.0.2, and auto. The default value is auto. For the value, auto, the SMB client protocol version 2.1 or version 3.0.2 is used based on the specified version of the SMB server.

smbc_signing

Specifies whether the file system of SMB client requires digital signature for communication. The valid values are enabled and required. The default value is enabled.

smbc_secure_negotiate

Specifies whether file system of SMB client requires secure dialect negotiation capability. Valid values are desired, required, and disabled. Default value is desired.

smbc_encryption

Specifies whether SMB client file system requires encryption. Valid values are desired, required, and disabled. Default value is desired.

The following table shows few tunable parameters in the `smbctune.conf` file and their peer options for the **mount** command.

Tunable parameter of the <code>smbctune.conf</code> file	Corresponding mount option	Valid tunable parameter values (<code>smbctune.conf</code> file)	Default tunable parameter values (<code>smbctune.conf</code> file)
<code>smbc_protocol_version</code>	<code>pver</code>	2.1, 3.0.2, auto	auto
<code>smbc_signing</code>	<code>signing</code>	enabled, required	enabled
<code>smbc_secure_negotiate</code>	<code>secure_negotiate</code>	desired, required, disabled	desired
<code>smbc_encryption</code>	<code>encryption</code>	desired, required, disabled	desired

Exit status

0

Indicates success.

>0

Indicates error.

Example

- To set the value of the **smbc_max_concurrent_mount** tunable parameter to 20, enter the following command:

```
/usr/sbin/smbctune -s smbc_max_concurrent_mount=20
```

- To update the **smbc_max_concurrent_mount** tunable parameter value, enter the following command:

```
/usr/sbin/smbctune -p -s smbc_max_concurrent_mount=20
```

Note: The command also updates the `smbctune.conf` file, which makes the updates persistent.

smdemon.cleau Command

Purpose

Cleans up the **sendmail** queue for periodic housekeeping.

Syntax

`/usr/lib/smdemon.cleanu`

Description

The **smdemon.cleanu** command is a shell procedure that cleans up the **sendmail** command queue and maintains the `/var/spool/mqueue/log` file.

To enable the **smdemon.cleanu** command, you must remove the comment statement by deleting the `#` character from the beginning of the **smdemon.cleanu** line in the `/var/spool/cron/crontabs/root` file. If the `/var/spool/mqueue` directory does not exist, do not change the `/var/spool/cron/crontabs/root` file.

Be careful that the average size of a log file for each **smdemon.cleanu** session multiplied by the number of log files does not use more space than you need. You can arrange the number of log files to suit your needs.

Note: The **smdemon.cleanu** command is not usually entered on the command line. The command is executed by the **cron** daemon.

Examples

To run the **smdemon.cleanu** procedure automatically, edit the `/var/spool/cron/crontabs/root` file and delete the `#` (comment character) from the beginning of the **smdemon.cleanu** line as follows:

```
# ulimit 5000; /usr/lib/smdemon.cleanu > /dev/null
```

Files

Item	Description
<code>/var/spool/cron/crontabs/root</code>	Schedules when the smdemon.cleanu command will run.
<code>/var/spool/mqueue</code>	Contains the log file and temporary files associated with the message in the mail queue.

smit Command

Purpose

Performs system management.

Syntax

```
smit [ -C | -M ] [ -D ] [ -f ] [ -h ] [ -l File ] [ -o PathName ] [ -p Entity/ValueString ] [ -r RunMode ] [ -s File ]  
[ -t ] [ -v ] [ [ -m | -n | -d ] FastPath ] [ -X ] [ -x ]
```

Description

The **smit** command invokes the System Management Interface Tool (SMIT). SMIT is an interactive interface application designed to simplify system management tasks. The **smit** command displays a hierarchy of menus that can lead to interactive dialogues. SMIT builds and runs commands as directed by the user. Because SMIT runs commands, you need the authority to execute the commands that SMIT runs.

SMIT creates two files, the **smit.script** file and the **smit.log** file. Invoking the **smit** command with the **-s** *PathName* flag saves the **smit.script** file in the file specified by the *PathName* parameter. If the **-s** flag is not specified, the script information is saved in the `$HOME/smit.script` file. Invoking the **smit** command with the **-l** *PathName* flag saves the **smit.log** file in the file specified by the *PathName* parameter. If the

-l flag is not specified, the log information is recorded in the **\$HOME/smit.log** file. You must have write permission for the directory in which you have requested the **smit** file to be written or the **smit.script** file and **smit.log** file are not created. SMIT does not overwrite the **smit.log** file or the **smit.script** file. The files are appended when possible.

The **smit.script** file automatically records the commands with the command flags and parameters used. The **smit.script** file can be used as an executable shell script to duplicate system configuration. SMIT creates the **smit.log** file, which contains additional detailed information that can be used by programmers in extending the SMIT system. The **smit.log** file is affected by the **-D**, **-l**, **-t**, and **-v** flags.

The **smit** command takes you to the top level of the menu hierarchy if you do not use the *FastPath* parameter. To enter the menu at lower levels, use the *FastPath* parameter. All commands run by SMIT can be used as *FastPaths*. The *FastPath* parameter will assist you as you become familiar with the commands. For example, you can enter: **smit chuser** to go directly to the dialog from which you can change user characteristics.

Note: User access to SMIT panels may be controlled with the **smitacl.user** or **smitacl.group** commands.

SMIT requires access to the following files:

Item	Description
sm_menu_opt	SMIT database
sm_name_hdr	SMIT database
sm_cmd_hdr	SMIT database
sm_cmd_opt	SMIT database
smit.log	SMIT log file
smit.script	SMIT script file
/usr/lpp/msg/.../smit.cat	Message Catalog

Note: If any of these files are corrupt, or exist on an NFS server and that server goes down, SMIT may fail to respond.

Flags

Item	Description
-C	Starts SMIT using an ASCII (also called Curses) interface.
-D	Sets the debug mode; sets -t and -v flags.
-d FastPath	Identifies that the <i>FastPath</i> is the name of a dialogue.
-f	Allows standard in and standard out from SMIT to be redirected.
-h	Displays the command usage message.
-l File	Redirects the smit.log file to the specified <i>File</i> .
-M	Starts SMIT using a windows (also called Motif) interface.
-m FastPath	Identifies that the <i>FastPath</i> is the name of a menu.
-n FastPath	Identifies that the <i>FastPath</i> is the name of a selector.
-o PathName	Specifies a directory <i>PathName</i> of an alternate repository for SMIT objects. The default directory is /etc/objrepos .

Item	Description
-p <i>Entity/ValueString</i>	This flag only applies to the smit windows version. Allows nameselects and dialogs to be filled in from the command line. Also allows you to operate on multiple entities simultaneously. You can set the environment variables ENTITY_SEP and VALUE_SEP to override the default comma and semicolon separators. You can enter <i>Entity/ValueString</i> in any of the following formats: <pre>"Entity1:Val1,Val2... ; Entity2:Val1,Val2... ; ..."</pre> or <pre>"Val1,Val2... ; Val1,Val2... ; ..."</pre>
-r <i>RunMode</i>	This flag only applies to smit windows version. Specifies the mode to run msmit in. You can enter the following values for <i>RunMode</i>: 1 Exit msmit when done is clicked in the output window. 2 Exit msmit when ok is clicked in a dialog. Print the dialog options upon exit. Do not run the command. 3 Run msmit silently, print the dialog options. Do not run the command. 4 Exit msmit when ok is clicked in the dialog. Print the commands upon exit. Do not run the command.
-s <i>File</i>	Redirects the smit.script file to the specified <i>File</i> .
-t	Records detailed trace information in the smit.log file.
-v	Records the command strings for intermediate and target task commands run by SMIT, and also records their output in the smit.log file.
-x	Does not run any command_to_execute , but still logs them for later execution.
-X	Does not run any command_to_discover , command_to_list , command_to_classify or command_to_execute .

Examples

1. To display the main menu in the overall system management hierarchy, enter:

```
smit
```

2. To change the characteristics of a user, enter:

```
smit chuser
```

The **chuser** command is an example of a *FastPath* parameter. The **smit** command and the *FastPath* parameter **chuser** takes you directly to the dialog, Change User Attributes, which guides you through changing the characteristics of a user.

3. To make the **smit.script** file executable for duplicate configuration, enter:

```
chmod +x smit.script
```

Then, to duplicate your configuration, enter:

```
smit.script
```

The **smit.script** file can be edited to create slight variations in the configuration commands, or to use only subsets of the commands. The **smit.script** file should be renamed or copied to prevent SMIT from modifying it.

Note: SMIT runs commands under the Korn shell (**/usr/bin/ksh**). Some command strings in the **smit.script** file may require this environment to run correctly.

Files

Item	Description
/usr/bin/smit	Contains the smit command.
/etc/objrepos	Specifies the default directory for the SMIT database.
smit.log	Specifies detailed information of your session, with time stamps.
smit.script	Specifies only the target task commands run by SMIT, with time stamps.

smitty Command

Purpose

Provides a Curses-based text interface to perform system management.

Syntax

```
smitty [ -C ] [ -D ] [ -f ] [ -h ] [ -l File ] [ -o PathName ] [ -s File ] [ -t ] [ -v ] [ [ -m | -n | -d ] FastPath ] [ -X ] [ -x ]
```

Description

The **smitty** command invokes the System Management Interface Tool (SMIT). SMIT is an interactive interface application designed to simplify system management tasks. The **smitty** command displays a hierarchy of menus that can lead to interactive dialogues. SMIT builds and runs commands as directed by the user. Because SMIT runs commands, you need the authority to execute the commands that SMIT runs.

Note: The **smitty** command is identical to **smit -C**.

SMIT creates two files, the **smit.script** file and the **smit.log** file. Invoking the **smitty** command with the **-s** *PathName* flag saves the **smit.script** file in the file specified by the *PathName* parameter. If the **-s** flag is not specified, the script information is saved in the **\$HOME/smit.script** file. Invoking the **smitty** command with the **-l** *PathName* flag saves the **smit.log** file in the file specified by the *PathName* parameter. If the **-l** flag is not specified, the log information is recorded in the **\$HOME/smit.log** file. You must have write permission for the directory in which you have requested the **smit** files to be written or the **smit.script** file and **smit.log** file are not created. SMIT does not overwrite the **smit.log** file or the **smit.script** file. The files are appended when possible.

The **smit.script** file automatically records the commands with the command flags and parameters used. The **smit.script** file can be used as an executable shell script to duplicate system configuration. SMIT creates the **smit.log** file, which contains additional detailed information that can be used by programmers in extending the SMIT system. The **smit.log** file is affected by the **-D**, **-l**, **-t**, and **-v** flags.

The **smitty** command takes you to the top level of the menu hierarchy if you do not use the *FastPath* parameter. To enter the menu at lower levels, use the *FastPath* parameter. All commands run by SMIT can be used as *FastPaths*. The *FastPath* parameter will assist you as you become familiar with the commands. For example, you can enter: **smitty chuser** to go directly to the dialog from which you can change user characteristics.

SMIT requires access to the following files:

Item	Description
sm_menu_opt	SMIT database
sm_name_hdr	SMIT database
sm_cmd_hdr	SMIT database
sm_cmd_opt	SMIT database
smit.log	SMIT log file
smit.script	SMIT script file
/usr/lpp/msg/.../smit.cat	Message Catalog

Note: If any of these files are corrupt, or exist on an NFS server and that server goes down, SMIT may fail to respond.

Flags

Item	Description
-C	Starts SMIT using a Curses-based text interface. This is the default for the smitty command.
-D	Sets the debug mode; sets -t and -v flags.
-d <i>FastPath</i>	Identifies that the <i>FastPath</i> is the name of a dialogue.
-f	Allows standard in and standard out from SMIT to be redirected.
-h	Displays the command usage message.
-l <i>File</i>	Redirects the smit.log file to the specified <i>File</i> .
-m <i>FastPath</i>	Identifies that the <i>FastPath</i> is the name of a menu.
-n <i>FastPath</i>	Identifies that the <i>FastPath</i> is the name of a selector.
-o <i>PathName</i>	Specifies a directory <i>PathName</i> of an alternate repository for SMIT objects. The default directory is /etc/objrepos .
-s <i>File</i>	Redirects the smit.script file to the specified <i>File</i> .
-t	Records detailed trace information in the smit.log file.
-v	Records the command strings for intermediate and target task commands run by SMIT, and also records their output in the smit.log file.
-x	Does not run any command_to_execute , but still logs them for later execution.
-X	Does not run any command_to_discover , command_to_list , command_to_classify or command_to_execute .

Examples

1. To display the main menu in the overall system management hierarchy, enter:

```
smitty
```

2. To change the characteristics of a user, enter:

```
smitty chuser
```

The **chuser** command is an example of a *FastPath* parameter. The **smitty** command and the *FastPath* parameter **chuser** takes you directly to the dialog, Change User Attributes, which guides you through changing the characteristics of a user.

Note: The **smitty chuser** command should be used to modify only local users.

3. To make the **smit.script** file executable for duplicate configuration, enter:

```
chmod +x smit.script
```

Then, to duplicate your configuration, enter:

```
smit.script
```

The **smit.script** file can be edited to create slight variations in the configuration commands, or to use only subsets of the commands. The **smit.script** file should be renamed or copied to prevent SMIT from modifying it.

Note: SMIT runs commands under the Korn shell (**/usr/bin/ksh**). Some command strings in the **smit.script** file may require this environment to run correctly.

Files

Item	Description
/usr/bin/smitty	Contains the smitty command.
/etc/objrepos	Specifies the default directory for the SMIT database.
smit.log	Specifies detailed information of your session, with time stamps.
smit.script	Specifies only the target task commands run by SMIT, with time stamps.

smrsh Command

Purpose

Restricted shell for sendmail.

Syntax

smrsh **-c** *command*

Description

The **smrsh** command is intended as a replacement for the **sh** command in the **prog** mailer in **sendmail** configuration files. The **smrsh** command limits the programs that can be run using the **sendmail** command syntax. This improves overall system security. **smrsh** limits the set of programs that a programmer can execute, even if **sendmail** runs a program without going through an alias or forward file.

The **smrsh** command requires that programs be in the **/var/adm/sm.bin** directory. This allows system administrators to choose which programs can be run by the **smrsh** command. The **smrsh** command also rejects any commands with the following characters on the command line to prevent end-run attacks: **, , < , > , | , ; , & , \$, \r (<RETURN>)**, or **\n (<NEWLINE>)** on the command line to prevent end run attacks.

- ,
- <
- >
- |
- ;

- &
- \$
- \r (<RETURN>)
- or \n (<NEWLINE>)

Initial pathnames on programs are stripped, so forwarding to **/usr/ucb/vacation**, **/usr/bin/vacation**, **/home/server/mydir/bin/vacation**, and **vacation** all actually forward to **/var/adm/sm.bin/vacation**. System administrators should be conservative about populating **/var/adm/sm.bin**. Reasonable additions are utilities such as **vacation(1)** and **procmail**. Never include any shell or shell-like programs (for example, **perl**) in the **sm.bin** directory. This does not allow the execution of arbitrary programs, but does not restrict the use of shell or perl scripts in the **sm.bin** directory (using the **#!** syntax).

Flags

-c *command*

Runs the program specified by *command*.

Location

/usr/sbin/smrsh

Default location of smrsh command.

Files

/var/adm/sm.bin

Directory for restricted programs.

smtctl Command

Purpose

The **smtctl** command controls the enabling and disabling of processor simultaneous multithreading mode.

Syntax

smtctl [**-m off** | **on** [**-w boot** | **now**]]

smtctl [**-t #SMT** [**-w boot** | **now**]]

smtctl [**-m suspend** [**-w boot**]]

smtctl [**-m limit** [**-t #SMT**] [**-w boot**]]

smtctl [**-m recommended** [**-w boot** | **now**]]

Description

This command is provided for privileged users and applications to control utilization of processors with simultaneous multithreading support. The simultaneous multithreading mode allows processors to have thread level parallelism at the instruction level. This mode can be enabled or disabled for all processors either immediately or on subsequent boots of the system. This command controls the simultaneous multithreading options.

Each individual Simultaneous Multi-threading (SMT) thread of a physical processor core is treated as an independent logical processor by AIX. The AIX operating system limits the combination of physical processor cores assigned and SMT modes in order to maintain symmetry across all of the physical processor cores assigned to AIX. Due to this limitation, the number of logical processors is less than or equal to 1024 for AIX 7.1 and 256 for AIX 6.1.

The POWER8 processors are capable of SMT-8 which means up to 128 cores can be used in SMT-8 mode which yields 1024 logical processors. A lower SMT mode must be used for AIX users to be able to use more than 128 POWER8 cores.

Number of threads

When booting a P8 Logical Partition (LPAR), the default number of SMT threads is 4. To increase the default number of SMT threads dynamically, enter:

```
smtctl -m on
smtctl -t 8
```

The change to SMT-8 is effective immediately and reboot is not required. If you want the setting to persist after rebooting, then you must rebuild the boot image with the **bosboot** command. The default SMT-4 is intended for better performance for existing applications that are not designed or compiled for more than 4 threads.

Number of cores

If you have allocated more than 128 cores to an LPAR, by default it uses 128 cores. This is to ensure that the AIX limit of maximum 1024 logical processors is not exceeded if SMT-8 is enabled (128 cores * SMT8 = 1024 total). If you want an LPAR to use more than 128 cores, then you need to run a sequence of following AIX commands to establish a limit to the number of SMT threads that are available per core.

```
smtctl -m limit -t 4
bosboot -a
shutdown -Fr
```

Upon rebooting, AIX negotiates with the firmware to allow up to 256 cores because the operating system's limit of 1024 processors will not be exceeded with the specified limit of 4 SMT threads. You can exceed 256 cores if you run the **smtctl** command as stated above, but with a limit of 2 instead of 4. The following command suspends SMT capability allowing more cores.

```
smtctl -m suspend
bosboot -a
shutdown -Fr
```

Flags

Item	Description
-m off	Set the simultaneous multithreading mode to disabled. This option cannot be used with the -t flag.
-m on	Set the simultaneous multithreading mode to enabled. By using the -m flag, the maximum number of threads supported per processor is enabled. This option cannot be used with the -t flag.
-t #SMT	Set the number of the simultaneous threads per processor. The value may be set to one to disable simultaneous multi-threading. The value may be set to two for systems that support 2-way simultaneous multi-threading and the value may be set to four, for the systems that support 4-way simultaneous multi-threading.
-w boot	Makes the simultaneous multithreading mode change effective on next and subsequent reboots if you run the bosboot command before the next system reboot.
-w now	Makes the simultaneous multithreading mode change immediately but will not persist across reboot. If the -w boot or the -w now option is specified, the mode change is made immediately and will persist subsequent reboots if you run the bosboot command before the next system reboot.

Item	Description
-m limit	Limits the number of simultaneous multithreading threads to two, or the specified value if the -t flag is used and enables more processor nodes, if available, effective at the next reboot (running bosboot is required to rebuild the boot image). This limit cannot be dynamically changed during run time, and you must reboot to change the operating state.
-m suspend	Suspends the simultaneous multithreading capability, and enables more processor nodes, if available, effective at the next reboot (running bosboot is required to rebuild the boot image). This limit cannot be dynamically changed during run time, and you must reboot to change the operating state.
-m recommended	Sets the number of threads to a value that provides the best performance for the most common types of workloads that are based on the underlying physical processor type. This setting takes place immediately. You can also specify -w to start using the new value.

If no options are specified then the following simultaneous multithreading settings will be reported:

Item	Description
SMT Capability	Indicator that the physical or virtual processors are capable of simultaneous multithreading.
SMT Mode	Current runtime simultaneous multithreading mode of disabled or enabled.
SMT Boot Mode	Current boot time simultaneous multithreading mode of disabled or enabled.
SMT Threads	Number of simultaneous multithreading threads per physical or virtual processor.
SMT Bound	Indicator that the simultaneous multithreading threads are bound on the same physical or virtual processor.
SMT Thread Capability	Maximum number of simultaneous multi-threading threads per physical or virtual processor supported by the system.

Exit Status

Item	Description
0	Successfully completed the requested operation.
>0	An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To enable simultaneous multithreading for the current boot cycle, enter:

```
smtctl -m on -w now
```

The system displays a message similar to the following:

```
smtctl: SMT is now enabled.
```

2. To enable a 2-way simultaneous multithreading on a system that supports up to 4 way, enter:

```
smtctl -t 2 -w now
```

The system displays a message similar to the following:

```
smtctl: SMT is now enabled.
```

3. To view the current simultaneous multithreading mode settings and processor information, enter:

```
smtctl
```

The system displays a message similar to the following:

```
This system is SMT capable.

This system supports up to 4 SMT threads per processor
SMT is currently enabled.

SMT boot mode is set to disabled.

proc0 has 2 SMT threads
Bind processor 0 is bound with proc0
Bind processor 1 is bound with proc0

proc2 has 2 SMT threads
Bind processor 2 is bound with proc2
Bind processor 3 is bound with proc2
```

4. To disable simultaneous multithreading for the current boot cycle and for all subsequent boots, enter:

```
smtctl -m off
```

The system displays a message similar to the following:

```
smtctl: SMT is now disabled. It will persist across reboots if
you run the bosboot command before the next reboot.
```

Another method to disable simultaneous multi-threading for the current boot cycle and for subsequent boots, enter:

```
smtctl -t 1
```

Note: The boot image must be remade with the **bosboot** command before the next reboot.

Location

/usr/sbin/smtctl

Files

Item	Description
/usr/sbin/smtctl	Contains the smtctl command.

snap Command

Purpose

Gathers system configuration information.

Syntax

```
snap [-@] [-a] [-z "product_name=prd_name,..." | "class=myclass,..."] ALL [-M Timeout] [-A] [-b] [-B]
[-c] [-C] [-D] [-f] [-F] [-g] [-G] [-i] [-k] [-l] [-L] [-n] [-N] [-p] [-r] [-R] [-s] [-S] [-t] [-T
Filename] [-u user1,...] [-w] [-X] [-Y] [-o OutputDevice] [-d Dir] [-v Component] [-O FileSplitSize] [-P
Files] [script1 script2 ... | All | file:filepath] [-U]
```

```
snap -e [-m Nodelist] [-d Dir]
```

```
snap -z ADD ["product_name=prod_name" "class=myclass" "command_path=/tmp/myprod_myscript
-a"]
```

```
snap -z DELETE ["product_name=prod_name"]
```

Description

The **snap** command gathers system configuration information and compresses the information into a **pax** file. The file may then be written to a device such as tape or DVD, or transmitted to a remote system. The information gathered with the **snap** command might be required to identify and resolve system problems.

Note: Root user authority is required to execute the **snap** command. Use the **snap -o /dev/cd0** command to copy the compressed image to DVD. Use the **snap -o /dev/rmt0** command to copy the image to tape.

Use the **snap -o /dev/rfd0** command to copy the compressed image to diskette. Use the **snap -o /dev/rmt0** command to copy the image to tape.

At least 8 MB of temporary disk space is required to collect all system information, including contents of the error log. If you do not gather all system information with the **snap -a** command, less disk space may be required (depending on the options selected).

Note: If you intend to use a tape to send a snap image to IBM for software support, the tape must be one of the following formats:

- 8 mm, 2.3 GB capacity
- 8 mm, 5.0 GB capacity
- 4 mm, 4.0 GB capacity

Using other formats prevents or delays IBM software support from being able to examine the contents.

The **snap -g** command gathers general system information, including the following:

- Error report
- Copy of the customized Object Data Manager (ODM) database
- Trace file
- User environment
- Amount of physical memory and paging space
- Device and attribute information
- Security user information
- Configuration and tuning parameter information of the system

The output of the **snap -g** command is written to the **/tmp/ibmsupt/general/general.snap** file.

The **snap** command checks for available space in the **/tmp/ibmsupt** directory, the default directory for **snap** command output. You can write the output to another directory by using the **-d** flag. If there is not enough space to hold the **snap** command output, you must expand the file system.

Each execution of the **snap** command appends information to previously created files. Use the **-r** flag to remove previously gathered and saved information.

Flags

Item	Description
-@	Gathers the workload partition information.
-a	Gathers all system configuration information except HACMP specific data. To gather HACMP specific data, run the snap -e option. Collection of registered debug data scripts for external products gets executed and their data is also included as part of system configuration and it can be limited for selected products by specifying their names with the -z flag. The -a option requires at least 8 MB of temporary disk space.
-A	Gathers asynchronous (TTY) information.
-b	Gathers SSA information.
-B	Bypasses collection of SSA adapter dumps. The -B flag only works when the -b flag is also specified; otherwise, the -B flag is ignored.
-c	Creates a compressed pax image (snap.pax.gz). Note: Information that is not gathered with this option must be copied to the snap directory tree before using the -c flag. If a test case is needed to demonstrate the system problem, copy the test case to the /tmp/ibmsupt/testcase directory before compressing the pax file. Any directories that are defined by the user must be saved in the /tmp/ibmsupt/other directory for the snap command to compress them.
-C	Retrieves all the files in the fwddump_dir directory. The files are placed in the "general" subdirectory. The -C snap option behaves the same as -P* .
-D	Gathers dump and /unix information. The primary dump device is used. Note: 1. If bosboot -k was used to specify the running kernel to be other than /unix, the incorrect kernel is gathered. Make sure that /unix is, or is linked to, the kernel in use when the dump was taken. 2. If the dump file is copied to the host machine, the snap command does not collect the dump image in the /tmp/ibmsupt/dump directory. Instead, it creates a link in the dump directory to the actual dump image.
-d <i>AbsolutePath</i>	Identifies the optional snap command output directory (/tmp/ibmsupt is the default). You must specify the absolute path.
-e	Gathers HACMP specific information. Note: HACMP specific data is collected from all nodes belonging to the cluster. This flag cannot be used with any other flags except -m and -d.
-f	Gathers file system information.
-F	Gathers flash adapter information.
-g	Gathers the output of the lspp -hac command, which is required to recreate exact operating system environments. Writes output to the /tmp/ibmsupt/general/lspp.hac file. Also collects general system information and writes the output to the /tmp/ibmsupt/general/general.snap file.
-G	Includes predefined Object Data Manager (ODM) files in general information collected with the -g flag.
-i	Gathers installation debug vital product data (VPD) information.

Item	Description
-k	Gathers kernel information
-l	Gathers programming language information.
-L	Gathers LVM information.
-m <i>Nodelist</i>	Node name list (separated by commas) to gather HACMP information. Note: Currently this flag is only valid with the -e flag.
-M <i>Timeout</i>	Specifies the maximum time out value in seconds, that the snap frame work waits before it kills one registered external product debug data command. Default time out value is 300 seconds.
-n	Gathers Network File System (NFS) information.
-N	Suppresses the check for free space required.
-o <i>OutputDevice</i>	Copies the compressed image onto the specified device.
-O <i>FileSplitSize</i>	Used to enable splitting of the snap output files into smaller files. The size of these files is specified as a parameter to the -O option and must be specified in megabytes. This flag can only be used when the -c flag is specified.
-p	Gathers printer information.
-P <i>Files</i>	Retrieves the named <i>Files</i> from the <code>fwdump_dir</code> directory. If -P * is specified, all the files in the directory are gathered. The files are placed in the <code>general</code> subdirectory. The -C snap option behaves the same as -P* .
-r	Removes snap command output from the <code>/tmp/ibmsupt</code> directory.
-R	Gathers SCSI RAID information.
-s	Gathers Systems Network Architecture (SNA) information.
-S	Includes security files in general information collected with the -g flag.
-t	Gathers Transmission control protocol information.
-T <i>Filename</i>	Gathers all the log files for a multi-CPU trace. Only the base file, trcfile , is captured with the -g flag.
-u <i>user1,user2...</i>	Specifies comma separated user names whose shell and System Management Interface Tool (SMIT) history is to be collected.
-v <i>Component</i>	Displays the output of the commands executed by the snap command. Use this flag to view the specified name or group of files. Note: Press the Ctrl-C key sequence to interrupt the snap command. A prompt will return with the following options: press the Enter key to return to current operation; press the S key to stop the current operation; press the Q key to quit the snap command completely.
-w	Gathers WLM information.
-X	Gathers X.25 (Packet-based Communication Protocol) information.
-Y	Gathers InfiniBand information and saves it in the <code>/tmp/ibmsupt/IB</code> directory.

Item	Description
-z	Facilitates debug data collection for external products. • The ADD keyword allows external products to register their debug data collection script with the snap framework. • The DELETE keyword allows external products to deregister their debug data collection script with the snap framework. When a product name is specified as parameter to the product_name attribute, a registered debug data collection command is executed. To collect data for more than one product specify the required product names in the product_name attribute. When a class name is specified as parameter to the class attribute, registered debug command of all the products in that class are executed. To collect data for more than one class specify the required class names in the class attribute. When ALL is specified as parameter, registered debug data collection command of all the products in all classes is executed. When any script gets executed, system appends the product name to the list pointed by SNAPDEBUGDATA environment variable.
-U	Collects Live kernel update information and save it in the /tmp/ibmsupt/liveupdate directory.

Parameters

Arguments

Names of third-party scripts to be executed are specified as parameters to snap. A parameter can be a single word or a list of words enclosed in quotes. When parameters are enclosed in quotes, the first parameter in the list represents the name of the script and the subsequent words represent the arguments to pass to the script.

When **All** is specified as a parameter, all the scripts in the script repository are executed. No script parameters may be passed in this case.

If the **file:** keyword is used and is immediately followed by a path to a file, that file is read to get the scripts to execute. Each line in the file represents a script and optional parameters to the script .

snap Scripts

A third-party script must be executable in **/usr/lib/ras/snapscripts**, and must follow the guidelines described below. When called during pass 1, a script must return its size estimation to snap. In pass 2, it collects the data and saves it as specified by snap.

The script must read and utilize the following environment variables, **SNAPDIR**, **PASSNO**, **SCRIPTSIZE** and **SCRIPTLOG**.

The scripts or commands can also use **SNAPDEBUGDATA** variable to learn about the debug data collected by snap script. This variable has comma separated name of the products for which the **snap** command collects the data during execution.

All output files must be written to **\$SNAPDIR**. This is the directory where the script should be saving its output. The **PASSNO** variable contains the snap phase during which the script is called. During the first pass, the script should calculate a size estimation for the data it will write during the second pass. It will then write that numeric estimation to the file pointed to by **\$SCRIPTSIZE**. The value saved to the file should be in decimal. snap passes the path to a log file where all debug data for the script should be saved. Standard out and standard error should not be redirected by the script, because snap will save standard out and standard error to **\$SNAPDIR/ScriptName.out** and **\$SNAPDIR/ScriptName.err**, respectively.

The following example shows a snap script:

```
#!/usr/bin/ksh

if [ "$PASSNO" = 1 ]
then
    (( size=99999 ))
    ....
    # this is where code to do the size estimation should go.
    ....
    echo $size > $SCRIPTSIZE
else if [ "$PASSNO" = 2 ]
then
    # debug information should go to $SCRIPTLOG
    echo "Debug Data" >> $SCRIPTLOG

    # .....where the work to collect the data takes place
    # ...

    # The data collected should be written to $SNAPDIR
    touch $SNAPDIR/foo_output1
    touch $SNAPDIR/foo_output2
fi
fi
```

Note: To collect information about virtual SCSI devices, run the **snap client_collect,all** command. If you need to collect data from the Virtual I/O server, see the **snap** command page on the Virtual I/O server, which uses different syntax from the **snap** command on AIX.

The following scripts can be run when you run the **snap** command with the **-a** or **-g** flags:

- To run with the **-a** flag: svCollect, client_collect, lsvirt
- To run with the **-g** flag: svCollect, client_collect

Splitting of snap Output

If it is split, snap output might look like the following:

```
% ls -l
total 112048
-rw----- 1 root    system    3145728 Feb 01 05:59 snap.hastings.020122055925.pax.gzaaa
-rw----- 1 root    system    3145728 Feb 01 05:59 snap.hastings.020122055925.pax.gzaab
-rw----- 1 root    system    3145728 Feb 01 05:59 snap.hastings.020122055925.pax.gzaac
-rw----- 1 root    system    3145728 Feb 01 05:59 snap.hastings.020122055925.pax.gzaad
-rw----- 1 root    system    3145728 Feb 01 05:59 snap.hastings.020122055925.pax.gzaae
-rw----- 1 root    system    3145728 Feb 01 05:59 snap.hastings.020122055925.pax.gzaaf
-rw----- 1 root    system    3145728 Feb 01 05:59 snap.fvt-zz4-
lp3.020122055925.pax.gzaag
-rw----- 1 root    system    3145728 Feb 01 05:59 snap.fvt-zz4-
lp3.020122055925.pax.gzaah
-rw----- 1 root    system    3145728 Feb 01 05:59 snap.fvt-zz4-
lp3.020122055925.pax.gzaai
-rw----- 1 root    system    3145728 Feb 01 05:59 snap.fvt-zz4-
lp3.020122055925.pax.gzaaj
-rw----- 1 root    system    3145728 Feb 01 05:59 snap.fvt-zz4-
lp3.020122055925.pax.gzaak
-rw----- 1 root    system    3145728 Feb 01 05:59 snap.fvt-zz4-
lp3.020122055925.pax.gzaal
-rw----- 1 root    system    2234613 Feb 01 05:59 snap.fvt-zz4-
lp3.020122055925.pax.gzaam
```

Executing Third Party Scripts

An external product debug data collection command or script is a standalone executable. The script is registered with the snap framework before it can be used to collect user defined debug data. These scripts can be de-registered as per user discretion.

Following is the ODM class defined in the system.

```
#define DEFAULTSIZE 256
#define DATA_VALUESIZE 1024
```

```
class snap_config {
    char product_name[DEFAULTSIZE]; key
    char class[DEFAULTSIZE];key
    char command_path[DATA_VALUESIZE];
    vchar sc_reserved1[DATA_VALUESIZE];
    vchar sc_reserved2[DATA_VALUESIZE];
}
```

product_name

Specify the name of the product. The same name is used for deregistration of the product debug data collection script.

class

Class can be a storage, a network or a database. You can choose appropriate class based on the product or define your own class. Class helps in the classification of the products. Users can contact IBM service personnel to add any other class in the **snap** documentation.

command_path

Path of the command or executable along with its options. **sc_reserved1** and **sc_reserved2** are reserved.

Registration of Third Party Debug Script with Snap framework

Registration can be done in two ways:

1. You can explicitly run **odmadd** command to add the entry. In such case:

- You must copy the script or executable to `/usr/lib/ras/snapscript/bin/<productname>` directory.

Points to remember:

a. You need to enter the command before executing the **odmadd**:

```
export ODMDIR=/usr/lib/objrepos
```

b. After the **odmadd** command completes, you can restore the old value of the **ODMDIR** command.

c. You can continue running the **snap** command. For example, the content of *myfile* is given below:

```
product_name=myprod
class=myclass
command_path=/usr/lib/ras/snapscripts/bin/prod_name/myscript1.sh -t 10
export ODMDIR=/usr/lib/objrepos
odmadd myfile
```

Note: Users making direct entry to ODM must take care of duplicate entries as the **snap** command processes only one entry for a particular product name. So, the **odmdelete** command must be executed before the **odmadd** command is invoked.

2. Use the **ADD** keyword with the **-z** flag.

Note:

- 1. If the debug binary is changed or updated, the user must re-register the component to update the snap repository with the latest binary.
- 2. Combination of multiple commands as a part of **command_path** variable is not supported. For example, the following format is not supported:

```
command_path=<path>/ls|<path>/grep myfile
```

- 3. Special characters like `'`, `<`, `|` are not supported as values to the **command_path** variable.

Deregistration of Third party debug scripts from Snap framework

Deregistration can be done in two ways:

1. Use the **odmdelete** command to deregister the product. For example,

```
export ODMDIR=/usr/lib/objrepos
odmdelete -o snap_config -q product_name=productname
```

2. Use the **DELETE** keyword with the **-z** flag. For example,

```
Snap -z DELETE product_name=productname
```

Examples

1. Enter the following command to gather all system configuration information:

```
snap -a
```

The output of this command is written to the **/tmp/ibmsupt** directory.

2. Enter the following command to create a **pax** image of all files contained in the **/tmp/ibmsupt** directory:

```
snap -c
```

3. Enter the following command to gather general system configuration information, including the output of the **lspp -hac** command:

```
snap -g -o /dev/rfd0
```

Output is written to the **/tmp/ibmsupt/general/lspp.hac** and **/tmp/ibmsupt/general/general.snap** files. This command also writes the system information to a removable diskette.

4. Enter the following command to gather HACMP specific information from nodes node1 and node2 belonging to a single cluster:

```
snap -e -m node1,node2
```

Output is written to the **/tmp/ibmsupt/hacmp** directory.

5. To run the scripts foo1, foo2 and foo3. where foo1 takes no argument, foo2 takes three arguments and foo3 takes one argument, type the following:

```
snap foo1 "foo2 -x -y 3" "foo3 6"
```

Output is written to **/tmp/ibmsupt/snapscripts/foo1**, **/tmp/ibmsupt/snapscripts/foo2** and **/tmp/ibmsupt/snapscripts/foo3** assuming the destination directory is the default, **/tmp/ibmsupt**.

6. To specify the **All** parameter to run all the scripts, type:

```
snap All
```

Note: No parameters are passed in this case.

7. To specify the path to a file containing the name and optional parameter list of scripts to execute, type:

```
snap file:/tmp/scriptnames
```

A sample input file to execute the scripts from example 5:

```
foo1
foo2 -x -y 3
foo6
```

8. If splitting of the snap output into 4MB files is desired, type:

```
snap -a -c -O 4
```

9. To submit only the HACMP snap -e data from nodes node1 and node2, enter the following command:

```
snap -e -m node1,node2
snap -c
```

Submit the <pax.gz> file to IBM according to the instructions of the service representative.

10. To submit all of the snap data from nodes node1 and node2, enter the following commands:

```
snap -e -m node1,node2
snap -a
snap -c
```

Submit the <pax.gz> file to IBM according to the instructions of the service representative.

11. To register a debug script present in the /usr/lpp/abc/debug_abc directory of product **abc**, in class **storage** enter the following command:

```
snap -z ADD "product_name=abc" "class=storage" "command_path=/usr/lpp/abc/debug_abc -a"
```

12. To deregister a debug script of product **abc**, enter the following command:

```
snap -z DELETE "product_name=abc"
```

13. To gather debug data of multiple products, enter the following command:

```
snap -z "product_name=abc, product_name=def"
```

Files

Item	Description
/usr/sbin/snap	Contains the snap command.
/tmp/ibmsupt	Contains snap command output.
/tmp/ibmsupt/general/lslpp.hac	Contains the output of the lslpp -hac command, which is required to recreate exact operating system environments.
/tmp/ibmsupt/general/general.snap	Contains general system information that is collected with the snap -g command.
/tmp/ibmsupt/testcase	Contains the test case that demonstrates your system problem.
/tmp/ibmsupt/other	Contains user-defined directory.

snapcore Command

Purpose

Gathers the **core** file.

Syntax

```
snapcore[ -d Dir] [-r] core [program]
```

Description

The **snapcore** command gathers the **core** file, program, and libraries used by the program and compresses the information into a **pax** file. The file can then be downloaded to disk or tape, or transmitted to a remote system. The information gathered with the **snapcore** command is required to identify and resolve a problem with the application.

The **snapcore** command checks for available space in the **/tmp/snapcore** directory, the default directory for **snapcore** command output. You can write the output to another directory by using the **-d** flag. If there is not enough space to hold the **snapcore** command output, you must expand the file system.

Each execution of the **snapcore** command creates a new archive file. The archive file is named **snapcore_\$pid.pax**. Use the **-r** flag to remove the previously created archive file. This command uses **\$pid** (pid of the **snapcore** command) to create a unique name file and preserve any previously created archives.

Specify the full path name for core and program. If the program name is not specified, **snapcore** reads the program name from the **core** file and searches for the location in directories contained in the **PATH** variable.

Flags

Item	Description
-dDir	Identifies the optional snapcore command output directory (/tmp/snapcore is the default).
-r	Removes snapcore command output from the /tmp/snapcore directory.

Examples

1. To gather the **core** file, enter the following:

- a. `snapcore <core file name> <program name>`
- b. `snapcore <core file name>`

Directories contained in the **PATH** variable are searched to find the program file. The **pax** file is created in **/tmp/snapcore** directory.

2. To clean the previously created core archive and create a new one, enter the following:

```
snapcore -r<core file name> <program name>
```

The **pax** file is created in **/tmp/snapcore** directory.

3. To create the **core** file archive in an alternate directory, enter the following:

```
snapcore -d<dir name> <core file name> <program name>
```

The **pax** file is created in **<dirname>/tmp/snapcore** directory.

4. To clean the **/tmp/snapcore** directory, enter the following:

```
snapcore -r
```

Files

Item	Description
/usr/sbin/snapcore	Contains the snapcore command.
/tmp/snapcore	Contains core file archive.

snapshot Command

Purpose

Modify, create, or view properties of enhanced journaled file system (JFS2) snapshots.

Syntax

To create an external snapshot

snapshot -o snapfrom=snappedFS snapshotLV

snapshot -o snapfrom=snappedFS -o size=Size

To create an internal snapshot

snapshot -o snapfrom=snappedFS -n snapshotName

To delete an external snapshot

snapshot -d snapshotLV

To delete an internal snapshot

snapshot -d -n snapshotName snappedFS

To query a JFS2 file system

snapshot -q [-c fieldSeparator] snappedFS

To query an external snapshot

snapshot -q [-c fieldSeparator] snapshotLV

To query an internal snapshot

snapshot -q -n snapshotName [-c fieldSeparator] snappedFS

To modify an external snapshot

snapshot -o size=Size snapshotLV

Notes:

- The **snapshot** command does not support modifying internal snapshots. The size of an internal snapshot is limited by the amount of free space available in the file system itself.
- >| A logical volume that is automatically created for the external snapshot follows the same mirror pool strictness policy of the logical volume where the snappedFS file system is created. |<
- >| A logical volume that is automatically created for the external snapshot follows the same encryption policy of the logical volume where the snappedFS file system is created.
 - The **snapshot** command copies the passphrase and key server authentication methods from the snappedFS logical volume to the external snapshot logical volume. The snappedFS logical volume and the snapshot logical volume share the same key server authentication method. If you remove the key server authentication method from one logical volume, the other logical volume cannot access the key server authentication method as well.
 - The **snapshot** command adds new Platform keystore (PKS) authentication method to the external snapshot logical volume if snappedFS logical volume has PKS authentication method configured and if PKS key slots are available.

|<

Description

This command provides an interface to JFS2 snapshots.

The maximum number of external snapshots per file system is 15, while the maximum number of internal snapshots per file system is 64.

You cannot have both internal snapshot and external snapshot of a file system at the same time.

Flags

Item	Description
-c <i>fieldSeparator</i>	Specifies the output from the snapshot query to be displayed in colon format. The <i>fieldSeparator</i> is the character to use to separate the fields of the display.
-d	Deletes the snapshot and any previous snapshots. If the snapshot is an external snapshot, the logical volume containing the snapshot is also deleted unless you specify the -s flag. For an external snapshot, the <i>snapshotLV</i> parameter specifies the snapshot to delete. For an internal snapshot, the <i>snappedFS</i> parameter specifies the file system containing the snapshot to delete. The -n flag specifies the name of the snapshot to delete.
-n <i>snapshotName</i>	Specifies the access point for the internal snapshot under the <i>snappedFS/.snapshot/snapshotName</i> . If you specify the -n flag when creating a snapshot, the file system specified by the <i>snappedFS</i> parameter must be enabled for internal snapshots. Otherwise, an error message is displayed and no snapshot is created. To enable a file system to use internal snapshots, specify the isnapshot option when you create the file system with the mkfs command (-o isnapshot={yes}) or the crfs command (-a isnapshot = {yes}).
-o snapfrom=snappedFS	Creates a snapshot of the file system specified by the <i>snappedFS</i> parameter. If the -n flag is specified, an internal snapshot is created. If the <i>snapshotLV</i> parameter is specified, the logic volume must already exist and must be in the same volume group as the file system specified by the <i>snappedFS</i> parameter. If the specified logic volume is already in use as a snapshot or a file system known to the /etc/filesystems file, the command issues an error message and fails. If the -n flag and the <i>snapshotLV</i> parameter are not specified, a new logical volume is created for the external snapshot.
-o size=Size	Specifies the size of a new logical volume for an external snapshot when you specify this flag with the -o snapfrom=snappedFS flag. Otherwise, this flag increases the size of the external snapshot specified by the <i>snapshotLV</i> field to the value of <i>Size</i> . This flag is ignored if any other flag is specified. If the value of the <i>Size</i> attribute is followed by a character M , the value is considered to be in megabytes. If the value is followed by a character G , the value is considered to be in gigabytes. If the value is followed by a character T , the value is considered to be in terabytes. If the value is followed by a character P , the value is considered to be in petabytes. If the value of the <i>Size</i> attribute is not followed by any character, the value is considered to be in 512-byte blocks.

Item	Description
-q	Displays information about the specified snapshot or snapshots. Specifies the following flags and options to determine the query as needed: - Specify the -n flag to display information about the named internal snapshot belonging to the file system that is specified by the <i>snappedFS</i> parameter is displayed. The information includes the file system that the snapshot belongs to, and the time when the snapshot is taken. - Specify the <i>snapshotLV</i> parameter to display information about the external snapshot. The information includes the file system that the snapshot belongs to, the time when the snapshot is taken, the size of the snapshot storage object, and the remaining free space. - Specify the <i>snappedFS</i> parameter to display information about all of the snapshots for the file system specified by the <i>snappedFS</i> parameter. For external snapshots, the information includes each of the snapshots and their storage objects, the time when the snapshot is taken, the size of the snapshot storage objects, and the remaining free space. For internal snapshots, the information includes each of the snapshots and the time when the snapshot is taken.
-s	Retains the specified logical volume for the specified snapshot when the external snapshot is deleted.

Parameters

Item	Description
<i>fieldSeparator</i>	The character to use to separate the fields of the display.
<i>snappedFS</i>	The JFS2 file system to act on for snapshot creation, deletion, or query.
<i>snapshotLV</i>	The logical volume of the external snapshot.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To create a snapshot for the **/home/janet/sb** file system on the **/dev/snapsb** logical volume, enter the following:

```
snapshot -o snapfrom=/home/janet/sb /dev/snapsb
```

This command creates a snapshot for the **/home/janet/sb** file system on the **/dev/snapsb** logical volume, which already exists.

2. To create a snapshot for the **/home/janet/sb** file system, enter the following:

```
snapshot -o snapfrom=/home/janet/sb -o size=16M
```

This command creates a 16-megabyte logical volume and creates a snapshot for the **/home/janet/sb** file system on the newly created logical volume.

3. To view information about all of the snapshots for the **/home/janet/sb** file system, enter the following:

```
snapshot -q /home/janet/sb
```

This command displays each snapshot for the **/home/janet/sb** file system along with the time when the snapshot was taken, the size of the snapshot storage object, and the remaining free space.

4. To increase the size of the snapshot on the **/dev/snapsb** device, enter the following:

```
snapshot -o size=64M /dev/snapsb
```

This command increases the **/dev/snapsb** device to 64 megabytes along with the snapshot contained on the device.

5. To delete the snapshot on the **/dev/snapsb** device, enter the following:

```
snapshot -d /dev/snapsb
```

This command deletes the snapshot contained on the **/dev/snapsb** device and removes the **/dev/snapsb** logical volume .

snapsplit Command

Purpose

To split a snap output file into multiple smaller files of arbitrary or specified size.

Syntax

```
snapsplit [ -s size ] [ -H machinename ] [ -f filename ]
```

```
snapsplit -u -T timestamp [ -H machinename ]
```

Description

The `snapsplit` command is used to split a snap output file into smaller files. This command is useful for dealing with very large snap files. It breaks the file down into files of a specific size that are multiples of 1 megabyte. Furthermore, it will combine these files into the original file when called with the `-u` option.

The output files are named as following: `snap.machinename.timestamp.pax.gzxxx`. *Machinename* is the host name and *timestamp* is in the format MMDDYYHHMMSS. In addition, xxx represents the extension for the split files that is crucial when putting these files back together. The extensions from the start of the files go in the following order: aaa, aab, aac, aad, aae ..., aaz, aba, abb, abc, abd, ..., abz, aca, acb, acc, ...

When performing `ls` on these files, the first file listed would represent the top of the original file and the last file, the end of the original file.

Note that this command should only be used for snap files that are paxed and compressed. When executed on local system where snap output was gathered, the `-H` option need not be used. That flag is provided for the case where user has moved a complete snap file to a remote system and wishes to split it. Any machine name may be selected, but it is recommended, to use the machine name where data was collected.

Flags

Item	Description
<code>-f filename</code>	Input <code>snapsplit</code> file. It must be a compressed pax file. The default is <code>snap.pax.gz</code> .

Item	Description
-H <i>machinename</i>	Name of the host machine. If none is specified, the default is the current host. Care must be exercised to name snap files for the appropriate system.
-s <i>size</i>	Specifies the size of snap output in multiples of 1 MB. The last file will be smaller or equal to this size. <i>Size</i> should be entered in megabytes. The default size is 1 MB.
-T <i>timestamp</i>	Timestamp of the snapsplit files to use in restoring the original snap output. It is in the format MMDDYYHHMMSS, where MM for month, DD for day, YY for year, HH for hours, MM is for minutes and SS is for seconds.
-u	Flag used for rejoining snapsplit files. Used with the -T flag.

Examples

1. To split the default snap file (snap.pax.gz must be in the current directory), enter the following command:

```
snapsplit
```

The output of this command is written to current directory.

2. To split file snap.somefile.pax.gz from system doe, enter the following command:

```
snapsplit -H doe -f snap.somefile.pax.Z
```

Note: The resulting files will be named snap.doe.MMDDYYHHMMSS.pax.gz.

3. To restore a file for which the snap files (snap.sue.102303141211.xxx) are for system sue, and timestamp 102303141211, type:

```
snapsplit -u -T 102303141211 -H sue
```



Attention: If any one of the snap files is missing or has been renamed, the snap file created will be corrupted.

4. To restore a snap file from files with time stamp 102603084512, and which are for the current system, type:

```
snapsplit -u -T 102603084512
```

5. To gather general system configuration information, including the output of the `ls1pp -hBc` command, type the following:

```
snap -g -o /dev/rfd0
```

Output is written to the /tmp/ibmsupt/general/ls1pp.hBc and /tmp/ibmsupt/general/general.snap files. This command also writes the system information to a removable diskette.

Files

Item	Description
/usr/sbin/ snapsplit	Contains the snapsplit command.

snmpd Daemon

Purpose

Starts the Simple Network Management Protocol (SNMP) agent as a background process.

Syntax

Refer to the syntax for either the [snmpdv1](#) daemon or the **snmpdv3** daemon.

Description

/usr/sbin/snmpd is a symbolic link to either the encrypted or non-encrypted version of the **snmpdv3** daemon which supports SNMP version 3.

Note: The encrypted version of the SNMP version 3 agent is available from the AIX Expansion Pack.

Files

Item	Description
/usr/sbin/snmpd	Contains a symbolic link to either /usr/sbin/snmpdv1 , /usr/sbin/snmpdv3e , or /usr/sbin/snmpdv3ne .
/usr/sbin/snmpdv1	Contains the SNMP version 1 agent.
/usr/sbin/snmpdv3e	Contains the encrypted version of the SNMP version 3 agent.
/usr/sbin/snmpdv3ne	Contains the non-encrypted version of the SNMP version 3 agent.

snmpdv1 Daemon

Purpose

Starts the Simple Network Management Protocol (SNMP) version 1 agent as a background process.

Syntax

snmpd [**-c** *ConfigFile*] [**-d** *Level*] [**-f** *LogFile*] [**-S**]

Description

The **snmpd** command starts the SNMP daemon. This command may only be issued by a user with root privileges or by a member of the system group.

The SNMP daemon is a server that supports the standard Simple Network Management Protocol (SNMP) documented by RFC 1157 and the Management Information Base (MIB) as defined in RFC 1155 and RFC 1213. The SNMP daemon provides the following three functions:

- Receiving and authenticating SNMP requests from network monitors.
- Processing requests and returning results to the originating monitor.
- Sending trap notification to all hosts listed in the configuration file.

The SNMP daemon server keeps log messages in a file specified by the *LogFile* variable if the **-f** flag is used or in a log file specified in the configuration file. When the size of the log file exceeds the predefined maximum log file size, the **snmpd** command will rotate the log file by moving the old log file to another file as follows:

- LogFile.3 is deleted.
- LogFile.2 is moved to LogFile.3.
- LogFile.1 is moved to LogFile.2.
- LogFile.0 is moved to LogFile.1.
- LogFile is moved to LogFile.0.
- Logging continues in LogFile.

If logging is not directed from the **snmpd** command line with the **-f** flag, logging can be directed from the configuration file.

Supported set variables are:

- **sysContact**
- **sysName**
- **sysLocation**
- **ifAdminStatus**
- **atPhysAddress**
- **atNetAddress**
- **ipForwarding**
- **ipDefaultTTL**
- **ipRouteDest**
- **ipRouteNextHop**
- **ipRouteType**
- **ipNetToMediaPhysAddress**
- **ipNetToMediaNetAddress**
- **ipNetToMediaType**
- **snmpEnableAuthenTraps**
- **smuxPstatus**
- **smuxTstatus**

See "Understanding SNMP Daemon Support for SET Request Processing" in *AIX Version 6.1 Communications Programming Concepts* for more information on the supported set variables.

The following commands should be issued before the SNMP daemon is started:

- **ifconfig lo0 loopback**
- **startsrc -s inetd**

These commands are normally executed during system startup when the **/etc/rc.net** and **/etc/rc.tcpip** shell scripts are called. (The **snmpd** command can be placed in the **/etc/rc.tcpip** shell script.)

The **snmpd** daemon should be controlled using the System Resource Controller (SRC). Entering **snmpd** at the command line is not recommended.

Manipulating the snmpd Daemon with the System Resource Controller

The **snmpd** daemon is a subsystem controlled by the System Resource Controller (SRC). The **snmpd** daemon is a member of the **tcpip** system group. The **snmpd** daemon is enabled by default and can be manipulated by SRC commands.

Use the following SRC commands to manipulate the **snmpd** daemon:

Item	Description
<u>startsrc</u>	Starts a subsystem, group of subsystems, or a subserver. Issuing the startsrc command causes the snmpd command to generate a <i>coldStart</i> trap.
<u>stopsrc</u>	Stops a subsystem, group of subsystems, or a subserver.
<u>refresh</u>	Causes a subsystem or group of subsystems to reread the appropriate configuration file. Issuing a refresh command causes the snmpd daemon to generate a <i>warmStart</i> trap.
<u>traceson</u>	Enables tracing of a subsystem, group of subsystems, or a subserver. If the user issuing the traceson command is not the root user, the debugging level will not exceed level 2.
<u>tracesoff</u>	Disables tracing of a subsystem, group of subsystems, or a subserver.

Item	Description
<u>lssrc</u>	Gets the status of a subsystem, group of subsystems, or a subserver. If the user issuing the long status form of the lssrc command is not the root user, no community name information is displayed.

Flags

Item	Description
-c <i>ConfigFile</i>	Specifies full path and file name of the configuration file for the snmpd daemon. This file is read when the snmpd daemon starts up and when a refresh or kill -1 signal is issued. If the -c flag is not specified, the default configuration file is /etc/snmpd.conf . See the snmpd.conf file for information on this file format.
-d <i>Level</i>	Specifies the level of tracing the snmpd command produces. The <i>Level</i> value can be one of: 0 - All notices, exceptions, and fatal messages 1 - Level 0 plus debug messages 2 - Level 1 plus a hexadecimal dump of incoming and outgoing packets 3 - Level 2 plus an English version of the request and response packets If the -d flag is not specified, the debugging level is set to 0.
-f <i>LogFile</i>	Specifies the full path and file name into which snmpd tracing information is logged. If the -f flag is not specified, no information will be logged. See the snmpd.conf file for more information on setting logging parameters.
-s	Enable the security option if it's specified. It will prevent the local non-root user from changing the value of MIB variable(s) on the local host.

Examples

1. To start the **snmpd** daemon, enter a command similar to the following:

```
startsrc -s snmpd -a "-f /tmp/snmpd.log"
```

This command starts the **snmpd** daemon and logs information to the **/tmp/snmpd.log** file at debug level 0.

2. To stop the **snmpd** daemon normally, enter:

```
stopsrc -s snmpd
```

This command stops the daemon. The **-s** flag specifies the subsystem that follows to be stopped.

3. To get short status from the **snmpd** daemon, enter:

```
lssrc -s snmpd
```

This command returns the name of the daemon, the process ID of the daemon, and the state of the daemon (active or inactive).

4. To get a long status from the **snmpd** daemon, enter:

```
lssrc -ls snmpd
```

If you are the root user, this long form of the status report lists the configured community names and associated access privileges and views for **snmp** requests. The long form also lists the community names associated with the hosts for trap notification, logging configuration parameters, **snmpd** specific configuration parameters and **smux** configuration parameters.

5. To enable tracing for the **snmpd** daemon, enter the following:

```
traceson -s snmpd
```

This command enables **snmpd** debugging if the **snmpd** daemon is configured for logging.

6. To view the contents of the DHCP Server database files **/etc/dhcpsd.ar** and **/etc/dhcpsd.cr**, enter:

```
lssrc -l -s dhcpsd
```

Files

Item	Description
<u>/etc/services</u>	Contains port assignments for required services. The following entries must be present in the /etc/services file if the entries are not already present: <pre> snmp 161/udp snmp-trap 162/udp smux 199/tcp </pre> Requirements: • The snmp port must be 161 as required by RFC 1157. • The snmp-trap port must be 162 as required by RFC 1157. • The smux port must be 199. • The /etc/services file is shipped with these entries already in place. • If the /etc/services file is being served from a server, these entries must be present in the server's /etc/services file.
<u>/etc/snmpd.conf</u>	Specifies the configuration parameters for the snmpd agent.
<u>/etc/mib.defs</u>	Defines the Management Information Base (MIB) variables the SNMP agent should recognize and handle.

snmpdv3 Daemon

Purpose

Starts the Simple Network Management Protocol (SNMP) version 3 agent as a background process.

Syntax

```
snmpd [ -d level ] [ -i interval ] [ -p port ] [ -S ] [ -c community ]
```

Description

The **snmpd** command starts the Simple Network Management Protocol (SNMP) daemon. This command may only be issued by a user with root privileges or by a member of the system group.

The SNMP daemon is a server that supports all the SNMPv1, SNMPv2c, and SNMPv3 protocols documented by RFCs 1157, RFD 1905, and RFC 2572. It also behaves as a SMUX server as defined by RFC 1227 and as a Distributed Protocol Interface (DPI) version 2.0 agent as defined by RFC 1592. The SNMP daemon provides the following three functions:

- Receiving and authenticating SNMP requests from network monitors.
- Processing requests and returning results to the originating monitor.
- Sending trap notification to all hosts listed in the configuration file.

The SNMP daemon server stores log messages in a file specified by the *LogFile* variable if the **-f** flag is used or stores log messages in a log file specified in the configuration file. The maximum value for number of log files is 4. When the size of the log file exceeds the predefined maximum log file size, the **snmpd** command moves the old log file to another file as follows:

- LogFile.3 is deleted.
- LogFile.2 is moved to LogFile.3.
- LogFile.1 is moved to LogFile.2.
- LogFile.0 is moved to LogFile.1.
- LogFile is moved to LogFile.0.
- Logging continues in LogFile.

The following commands should be issued before the SNMP daemon is started:

- **ifconfig lo0 loopback**
- **startsrc -s inetd**

These commands are normally executed during system startup when the **/etc/rc.net** and **/etc/rc.tcpip** shell scripts are called. (The **snmpd** command can be placed in the **/etc/rc.tcpip** shell script.)

The **snmpdv3** daemon should be controlled using the System Resource Controller (SRC). Entering **snmpd** at the command line is not recommended.

Manipulating the snmpd Daemon with the System Resource Controller

The **snmpdv3** daemon is a subsystem controlled by the System Resource Controller (SRC). The **snmpdv3** daemon is a member of the **tcpip** system group. The **snmpdv3** daemon is enabled by default and can be manipulated by SRC commands.

Use the following SRC commands to manipulate the **snmpd** daemon:

Item	Description
startsrc	Starts a subsystem, group of subsystems, or a subserver. Issuing the startsrc command causes the snmpdv3 command to generate a <i>coldStart</i> trap.
stopsrc	Stops a subsystem, group of subsystems, or a subserver.
lssrc	Gets the status of a subsystem, group of subsystems, or a subserver.

Flags

Item

-d *level*

Description

Specifies the level of tracing to be started. The valid values for level are 0-255. If the **-d** parameter is not specified, then the default level of 0 is used, meaning no tracing will be done. If the **-d** parameter is specified without a level, then a level of 31 is used, meaning all SNMP requests/responses/traps and DPI activity will be traced.

There are 8 levels of tracing provided. Each level selected has a corresponding number. The sum of the numbers associated with each level of tracing selected is the value which should be specified as level. The numbers for the trace levels are:

0

No tracing. This is the default.

1

Trace SNMP responses, requests, and traps.

2

Trace DPI level 1 and DPI level 2.

3

Same as level 1 plus level 2 plus internal trace.

4

Same as trace level 3 plus extended trace.

-i *interval*

Specifies the interval (in minutes) at which dynamic configuration changes to the SNMP agent should be written out to the **/etc/snmpdv3.conf** configuration file. Valid values are 0-10. The default value is 5. This parameter is only relevant when the **/etc/snmpdv3.conf** file is used for SNMPv3 configuration.

-p *port*

Listens for SNMP packets on this port. The default is port 161.

-s

Prevents non-root users from changing the MIB values.

-c *community*

Accepts the requests with the community name that the *community* parameter specifies.

Examples

1. To start the **snmpd** daemon, enter a command similar to the following:

```
startsrc -s snmpd
```

This command starts the **snmpd** daemon at debug level 0.

2. To stop the **snmpd** daemon normally, enter:

```
stopsrc -s snmpd
```

This command stops the daemon. The **-s** flag specifies the subsystem that follows to be stopped.

3. To get status from the **snmpd** daemon, enter:

```
lssrc -s snmpd
```

This command returns the name of the daemon, the process ID of the daemon, and the state of the daemon (active or inactive).

Files

Item	Description
/etc/services	Contains port assignments for required services. The following entries must be present in the /etc/services file if the entries are not already present: snmp 161/udp snmp-trap 162/udp smux 199/tcp
/etc/snmpdv3.conf	Specifies the configuration parameters for the snmpdv3 agent.
/etc/snmpd.boots	Specifies the engineID and the engineBoots for the snmpdv3 agent.
/etc/mib.defs	Defines the Management Information Base (MIB) variable the SNMP agent should recognize and handle.

snmpevent Command

Purpose

Sends ERRM events to an SNMP agent.

Syntax

```
snmpevent [-a host-name] [-c community] [-h]
```

Description

The `snmpevent` script sends a Simple Network Management Protocol (SNMP) trap of an event response resource manager (ERRM) event to a host running an SNMP agent. The agent formats the trap information into an SNMP trap and sends it to the SNMP manager defined in its configuration file. This script is meant to be called by the predefined ERRM response Generate SNMP trap. Event or rearm event information is captured and posted by ERRM in environment variables that are generated when an ERRM event or a rearm event occurs.

The `snmpevent` script can also be used as a template to create other user-defined actions. See the *RSCF Administration Guide* to understand how an event response resource runs an action command.

The following message template is sent as a trap when an event or a rearm event occurs and `snmpevent` is the defined response:

```
[ERRM_COND_SEVERITY] [ERRM_TYPE] occurred:  
Condition: [ERRM_COND_NAME]  
Node: [ERRM_NODE_NAME]  
Resource: [ERRM_RSRC_NAME]  
Resource Class: [ERRM_RSRC_CLASS_NAME]  
Resource Attribute: [ERRM_ATTR_NAME]
```

```
Attribute Type: [ERRM_DATA_TYPE]
Attribute Value: [ERRM_VALUE]
```

The environment variables have the following definitions:

ERRM_COND_SEVERITY

Specifies the significance of the condition resource that caused the event or rearm event. The valid values are: Critical, Warning, or Informational.

ERRM_TYPE

Specifies the type of event that occurred. The valid values are: event or rearm event.

ERRM_COND_NAME

Specifies the name of the condition resource with the attribute value that changed to cause this event or rearm event.

ERRM_NODE_NAME

Specifies the host name on which this event or rearm event occurred.

ERRM_RSRC_NAME

Specifies the name of the resource with the attribute that changed to cause this event or rearm event.

ERRM_RSRC_CLASS_NAME

Specifies the name of the resource class to which the resource that caused this event or rearm event belongs.

ERRM_ATTR_NAME

Specifies the name of the resource attribute that changed to cause this event or rearm event.

ERRM_DATA_TYPE

Specifies the data type of the resource attribute.

ERRM_VALUE

Specifies the value of the resource attribute that changed to cause this event or rearm event.

The `snmpevent` command captures these environment variable values and formats a generic message that is sent as a trap via a call to the `snmptrap` command.

Flags

-a *host-name*

Specifies the host name of the SNMP agent to which the AIX subagent will connect. By default, the subagent will connect to the SNMP agent running on the local node.

-c

Specifies the SNMP community to be used. This can be any string the SNMP agent will accept. The default is `public`.

-h

Writes this script's usage statement to standard output.

Parameters

log_file

Specifies the name of the file where event information is logged. An absolute path for the *log_file* parameter should be specified.

The *log_file* is treated as a circular log and has a fixed size of 64KB. When *log_file* is full, new entries are written over the oldest existing entries.

If *log_file* already exists, event information is appended to it. If *log_file* does not exist, it is created so that event information can be written to it.

Exit Status

0

The script has run successfully.

1

An error occurred when the script was run.

Restrictions

This script must be run on the node where the ERRM is running.

Standard Output

When the -h flag is specified, this script's usage statement is written to standard output.

Examples

1. Suppose the command `/opt/rsct/bin/snmpevent` is an action in the critical-notification response, which is associated with the CSM predefined condition `NodeChanged`. This can be done with the `mkcondresp` command followed by the `startcondresp` command. The `/etc/snmpdv3.conf` file should be configured to where the trap will be sent. In this example, if you want the trap sent to `9.117.16.246`, write the `/etc/snmpdv3.conf` file as follows:

```
VACM_GROUP group1 SNMPv1 public -
VACM_VIEW defaultView          internet          - included
-VACM_ACCESS group1 - - noAuthNoPriv SNMPv1 defaultView - defaultView -
NOTIFY notify1 traptag trap -
#TARGET_ADDRESS Target1 UDP 127.0.0.1          traptag trapparms1 - - -
TARGET_ADDRESS Target1 UDP 9.117.16.246        traptag trapparms1 - - -
TARGET_PARAMETERS trapparms1 SNMPv1 SNMPv1 public noAuthNoPriv -
COMMUNITY public public noAuthNoPriv 0.0.0.0 0.0.0.0 -
DEFAULT_SECURITY no-access - -
logging file=/usr/tmp/snmpdv3.log enabled
logging size=0 level=0
smux 1.3.6.1.4.1.2.3.1.2.1.2 gated_password # gated
snmpd smuxtimeout=200 #muxatmd
smux 1.3.6.1.4.1.2.3.1.2.3.1.1 muxatmd_password #muxatmd
```

Then, restart the `snmpd` daemon by first killing the `snmpd` daemon that is currently running and then starting it again:

```
# ps -ef | grep snmpd
  root  4570 12956   1 08:24:32 pts/0  0:00 grep snmpd
  root 13810     1   0 08:11:04 -    0:00 snmpd
# kill -9 13810
# snmpd
```

Next, change the `LParID` property of node `c175n08` to 12:

```
# chnode c175n08 LParID=12
```

Now, on the node `9.117.16.158` (the node with the SNMP manager that was specified in the `/etc/snmpdv3.conf` file), the SNMP manager should record something like this:

```
2002-07-15 09:09:25 c174tr1.ppd.pok.ibm.com [9.114.78.17] TRAP, SNMP v1,
community public
    enterprises.ibm Enterprise Specific Trap (1) Uptime: 0:01:45.00
    enterprises.ibm.ibmProd.191.1.6.1.0 = "Informational Event
occurred. Condition=NodeChanged Node=c174tr1.ppd.pok.ibm.com
Resource=c175n08.ppd.pok.ibm.com Resource Class=Node Resource
Attribute=Changed Attributes Attribute Type=CT_CHAR_PTR_ARRAY Attribute
Val={LParID} "
```

The output varies based on SNMP managers.

Location

/opt/rsct/bin/snmpevent

snmpinfo Command

Purpose

Requests or modifies values of Management Information Base (MIB) variables managed by a Simple Network Management Protocol (SNMP) agent.

Syntax

The get or next Option

```
snmpinfo [ -m get | next ] [ -v ] [ -c Community ] [ -d Level ] [ -h HostName ]  
[ -o ObjectsFile ] ... [ -t Tries ] [ -w Waittime ] Variable Instance ...
```

The set Option

```
snmpinfo -m set [ -v ] [ -c Community ] [ -d Level ] [ -h HostName ] [ -o ObjectsFile ] ... [ -t Tries ] [ -w  
Waittime ] Variable . Instance = Value ...
```

The dump Option

```
snmpinfo -m dump [ -v ] [ -c Community ] [ -d Level ] [ -h HostName ] [ -o ObjectsFile ] ... [ -t Tries ] [ -w  
Waittime ] [ Variable Instance ] ...
```

Description

The **snmpinfo** command requests or modifies values for one or more MIB variables for an SNMP agent. This command may only be issued by a user with root privileges or by a member of the system group.

If the you specify the **get** option, the **snmpinfo** command requests information about one or more MIB variables from an SNMP agent.

If you specify the **next** option, the **snmpinfo** command requests information from an SNMP agent about the instances following the specified instances. The **next** option makes it possible to obtain MIB values without knowledge of the instance qualifiers.

If you specify the **set** option, the **snmpinfo** command modifies values for one or more MIB variables for an SNMP agent. Only a few MIB variables are designated read-write. The agent that manages the MIB database may take various actions as a side effect of modifying MIB variables. For example, setting the **ifAdminStatus** MIB variable to 2 will normally shut down a network interface. The action taken is determined by the implementation of the SNMP agent that manages the database.

If you specify the **dump** option, the **snmpinfo** command can be used to traverse the entire MIB tree of a given agent. If a group is passed in as the *Variable* parameter, the **snmpinfo** command will traverse that specified path of the MIB tree.

The **snmpinfo** command has a debug facility that will dump debug information for transmitted and received packets. The facility is enabled with the **-d** flag.

Parameters

Item	Description
<i>Value</i>	Specifies the value to which the MIB <i>Variable</i> parameter is to be set. A value must be specified for each variable. If a value is not specified, the request packet will be invalid.
<i>Variable</i>	Specifies the name in text format or numeric format of a specific MIB variable as defined in the /etc/mib.defs file. If the option to the -m flag is next or dump , the <i>Variable</i> parameter may be specified as a MIB group.

Item	Description
<i>Instance</i>	Specifies the instance qualifier for the MIB <i>Variable</i> parameter. The <i>Instance</i> parameter is required if the option to the -m flag is get or set . The <i>Instance</i> parameter is optional if the option to the -m flag is next or dump .

Note:

1. There should be no blank spaces in the *Variable.Instance* parameter sequence.
2. If the *Instance* parameter is not specified, do not place a . (dot) after the *Variable* parameter.

For further information, consult RFC 1213, which defines the Management Information Base (MIB) for network management, and RFC 1157, which defines the SNMP protocol for creating requests for MIB information and formatting responses.

Flags

Item	Description
-c <i>Community</i>	Specifies the community name to be used to query the SNMP agent. If the -c flag is not specified, the default community name is public .
-d <i>Level</i>	Specifies the level of I/O debug information. The <i>Level</i> value can be one of: 0 No debug information. 1 Port bindings and the number of bytes transmitted and received. 2 Level 1 plus a hexadecimal dump of incoming and outgoing packets. 3 Level 2 plus an English version of the request and response packets. If the -d flag is not specified, the default debug level is 0.
-h <i>HostName</i>	Specifies the host name of the SNMP agent to be queried. The host name can be an IPv4 address, an IPv6 address, or a host name. If the -h flag is not specified, the default host name is the host name of the machine on which the user is currently logged in.
-m <i>Option</i>	Specifies the mode by which to access the MIB variables. The <i>Option</i> value can be one of: get Requests information about the specified MIB variables. next Requests the instances following the specified instances. set Modifies the specified write access MIB variables. dump Dumps the specified section of the MIB tree. Note: 1. The option name can be specified by the minimum number of characters required to make it unique. 2. If the -m flag is not specified, the default mode is get.

Item	Description
-o <i>ObjectsFile</i>	Specifies the name of the objects definition file that defines the MIB objects the snmpinfo command can request. If the -o flag is not specified, the default objects definition file name is /etc/mib.defs . See the mosy command for information on creating this file. More than one <i>ObjectsFile</i> can be referenced with the restriction that files containing parent definitions be specified before files containing child definitions.
-t <i>Tries</i>	Specifies the number of times the snmpinfo command transmits the SNMP request to the SNMP agent before terminating with the message no SNMP response. If the -t flag is not specified, the default number of tries is 3.
-v	Specifies that the output from the snmpinfo command be displayed in verbose mode. If the -v flag is not specified, the information will not be displayed in verbose mode.
-w	Specifies the wait time in seconds for the response from the snmpd agent. If the -w flag is not specified, the default wait time is 15 seconds.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Limitation

When the **snmpdv3** daemon encounters a **SMI-v2** data type MIB while processing a SNMPv1 protocol request from **snmpinfo** manager, it will skip the MIB until it encounters a **SMI-v1** data type MIB.

Work around

The **clsnmp** manager should be configured with **SNMPv2** type requests or **SNMPv3** type requests to dump all of the MIB variables with the **snmpdv3** daemon.

Examples

1. To get the values for the MIB variable **ifDescr.1**, for the interface associated with **ifIndex.1** and **SysDescr**, enter:

```
snmpinfo -m get -v sysDescr.0 ifDescr.1
```

In this example, the **-m get** flag specifies that the **snmpinfo** command should retrieve the value of MIB variables **ifDescr.1**, (the interface description for the interface associated with the **ifIndex.1**), and **sysDescr.0** (the system description of the local host).

2. To get the value for the MIB variable following the **ipAdEntIfIndex** MIB variable for the host specified by IP address 192.100.154.1, enter:

```
snmpinfo -m next -v 1.3.6.1.2.1.4.20.1.2.192.100.154.1
```

In this example, the **-m next** flag specifies that the **snmpinfo** command should retrieve the information for the MIB variable **ifAdEntIfIndex.192.100.154.1**.

3. To get the value of the first MIB variable in the system group, enter:

```
snmpinfo -m next -v -h giants system
```

In this example, the **-m next** flag specifies that the **snmpinfo** command should retrieve the information for the MIB variable following the system group, which is **sysDescr.0**; the **-v** flag

indicates verbose mode; the **-h** flag indicates that the agent to be queried is **giants**; the group to retrieve information from is **system**.

4. To set the value of a MIB variable, enter a command similar to the following:

```
snmpinfo -m set -v -h giants -c monitor -t 2 ifAdminStatus.1=2
```

In this example, the MIB **ifAdminStatus** variable is set to 2, or down, for the interface associated with **ifIndex.1** on the host known as **giants**. The **-c** flag specifies the community for the host. The **-t 2** flag specifies that the **snmpinfo** command will transmit the SNMP request to the SNMP agent 2 times before terminating if no response is received from the SNMP agent.

5. To dump a group of the MIB tree in verbose mode, enter a command similar to the following:

```
snmpinfo -m dump -v interfaces
```

In this example the **interfaces** group is dumped in verbose mode.

6. To dump the entire MIB tree, enter:

```
snmpinfo -m dump
```

7. To get the values for the **sysName.0** MIB variable, enter:

```
snmpinfo -m get -v -h 2000:1:1:1:209:6bff:feae:6d67 sysName.0
```

In this example, the **-m get** flag specifies that the **snmpinfo** command should retrieve the value of the **sysName.0** MIB variables. The **-v** flag indicates verbose mode. The **-h** flag indicates that the agent to be queried is an IPv6 address.

Files

Item	Description
/etc/mib.defs	Defines the Management Information Base (MIB) variables the SNMP agent should recognize and handle.

snmpmibd Daemon

Purpose

Starts the **snmpmibd** Distributed Protocol Interface (DPI) version 2 sub-agent daemon as a background process.

Syntax

```
snmpmibd [ -f file ] [ -d [level] ] [ -h hostname ] [ -c community ]
```

Description

The **snmpmibd** command starts the **snmpmibd** Distributed Protocol Interface (DPI) version 2 (**dpi2**) sub-agent. This command may only be issued by a user with root privileges or by a member of the system group.

The **snmpmibd** daemon complies with the standard Simple Network Management Protocol (SNMP) DPI version 2.0 defined by RFC 1592. It acts as a **dpi2** sub-agent to communicate with the **dpi2** agent through **dpiPortForTCP.0** (1.3.6.1.4.1.2.2.1.1.1.0) which is defined in RFC 1592 section 3.1.

The Management Information Base (MIB) is defined by RFC 1155(SMIv1) and RFC 2578(SMIv2).

The specific MIB variables that the **snmpmibd** command is managing are defined by the following RFCs:

RFC 1213

MIB-II

RFC 1229

Extension to the Generic-Interface MIB

RFC 1231

IEEE 802.5 Token Ring MIB

RFC 1398

Ethernet-like Interface Types MIB

RFC 1512

FDDI MIB

RFC 4022

MIB for the Transmission Control Protocol (TCP)

RFC 4113

MIB for the User Datagram Protocol (UDP)

RFC 4292

IP Forwarding Table MIB

RFC 4293

Management Information Base for the Internet Protocol (IP)

Note: The "**system**" and "**snmp**" groups defined in RFC1213 are not implemented by **snmpdmibd** daemon. Instead they are implemented by **snmpdv3** agent.

For the **RFC 4292**, read-only access is provided to the variables.

For the **RFC 4293**, read and write access is provided to the **ipv6IpForwarding** variable and the **ipv6IpDefaultHopLimit** variable. Read-only access is provided to the other MIB variables. Both the server and the agent must use the **SNMP v2c** protocol or later, because some variables defined in this RFC cannot be accessed using the **SNMP v1** protocol.

The **snmpmibd** daemon is normally executed during system startup when **/etc/rc.tcpip** shell script is called.

The **snmpmibd** daemon should be controlled using the System Resource Controller (SRC). Entering **snmpmibd** at the command line is not recommended.

Use the following SRC commands to manipulate the **snmpmibd** daemon:

startsrc

Starts a subsystem, group of subsystems, or a subserver.

stopsrc

Stops a subsystem, group of subsystems, or a subserver.

refresh

Causes a subsystem or group of subsystems to reread the appropriate configuration file.

lssrc

Gets the status of a subsystem, group of subsystems, or a subserver. If the user issuing the long status form of the **lssrc** command is not the root user, no community name information is displayed.

Flags**Item**

-c *community*

Description

Uses specified community name. If **-c** flag is not specified, the default community name is **public**.

Item	Description
-d [<i>level</i>]	Specifies tracing/debug level. The levels are: 8 DPI level 1 16 DPI level 2 32 Internal level 1 64 Internal level 2 128 Internal level 3 Adds the numbers for multiple trace levels. If -d flag is specified and the <i>level</i> is not specified, the default level is 56. If -d flag is not specified, the default level is 0.
-f <i>file</i>	A non-default configuration file. If the -f flag is not specified, the default configuration file is /etc/snmpmibd.conf . See /etc/snmpmibd.conf file for information on this file format.
-h <i>hostname</i>	Sends request to specified host. The value of the <i>hostname</i> attribute can be an IPv4 address, an IPv6 address, or a host name. If the -h flag is not specified, the default destination host is loopback (127.0.0.1).

Examples

1. To start the **snmpmibd** daemon, enter a command similar to the following:

```
startsrc -s snmpmibd -a "-f /tmp/snmpmibd.conf"
```

This command starts the **snmpmibd** daemon and reads the configuration file from **/tmp/snmpmibd.conf**.

2. To stop the **snmpmibd** daemon normally, enter:

```
stopsrc -s snmpmibd
```

This command returns the name of the daemon, the process ID of the daemon, and the state of the daemon (active or inactive).

3. To get long status from the **snmpmibd** daemon, enter:

```
lssrc -ls snmpmibd
```

If you are the root user, this long form of the status report lists the configuration parameters in **/etc/snmpmibd.conf**.

Files

Item	Description
/etc/snmpmibd.conf	Defines the configuration parameters for snmpmibd command.
/etc/mib.defs	Defines the Management Information Base (MIB) variables the SNMP agent and manager should recognize and handle.

snmptrap Command

Purpose

Generate a notification (trap) to report an event to the SNMP manager with the specified message.

Syntax

snmptrap [**-a** *host*] [**-h** *targethost*] [**-c** *community*] [**-o** *oid*] [**-d**] **-m** *message*

Description

Generate a notification (trap) to report an event to the SNMP manager with the specified message.

Flags

Item	Description
-a <i>host</i>	Specifies to connect to the SNMP agent on the specified host. If the -a flag is not specified, the default host is the local host. <i>host</i> can be an IPv4 address, an IPv6 address, or a host name.
-c <i>community</i>	Specifies community name to use. This community must have been set in /etc/snmpdv3.conf for SNMP version 3 or in /etc/snmpd.conf for SNMP version 1 and have the read access privilege at least to the SNMP agent running on the specified host or local host. If the -c flag is not specified, the default community name is "public".
-o <i>oid</i>	Specifies the event that generates the trap message. The <i>oid</i> specified, it will be used in the trap packet. If the parameter is not specified, the default OID is used in the trap packet. This specified OID is not validated for its correctness.
-d	Enables the debug facility
-h <i>targethost</i>	Specifies the target network manger host to which the trap message will be sent. The target host can be an IPv4 address, an IPv6 address, or a host name. The -h flag is different from the -a flag. The -a flag specifies a host where the AIX SNMP agent (snmp) must be running and the SNMP agent forwards this trap to network mangers. However, the -h flag does not require the AIX SNMP agent to forward the trap message to network managers, and it sends the trap directly to the network manager. If there are no -h and -a flags, the trap will be sent to the AIX SNMP agent on the local host.
-m <i>message</i>	Defines the message that the snmptrap command will send. <i>message</i> specifies the information the trap will hold. This information is in the text format. The -m flag must be the last flag specified.

Exit Status

- 0**
Trap information was sent out correctly.
- 1**
This indicates something was wrong during the process.

Examples

1. To send a trap with the message 'hello world' to the SNMP agent running on the local host, enter the following:

```
snmptrap -m hello world
```

Note: The community, public, must have read access to the SNMP agent running on the local host. For details, please refer to SNMP configuration documentation.

2. To send a trap with the community name, community1, and the message 'hello world' to the SNMP agent running on a remote host blah, enter the following:

```
snmptrap -c community1 -h blah -m hello world
```

Note: The community 'community1' must have read access to the SNMP agent running on the host 'blah'. For details, please refer to the SNMP configuration documentation.

3. To send a trap to the network manager running on a Linux platform and where the host name is nehcyg, type the following:

```
snmptrap -h nehcyg -m hello world
```

4. To send a trap to the network manager running on a Linux platform where the host name is *nehcyg*, and with the OID 1.3.6.1.4.1.2.6.191.1.6.1.0, enter the following:

```
snmptrap -h nehcyg -o 1.3.6.1.4.1.2.6.191.1.6.1.0 -m hello world
```

5. To send a trap with the community1 community name, and the message hello world to the SNMP agent that is running on an IPv6 address, enter the following command:

```
snmptrap -c community1 -h 2000:1:1:1:209:6bff:feae:6d67 -m hello world
```

Note: The community1 community must have read access to the SNMP agent that is running on the IPv6 address. For more information, see [SNMP for network management](#).

6. To send a trap to the network manager that runs on an IPv6 address, and with the OID 1.3.6.1.4.1.2.6.191.1.6.1.0, enter the following command:

```
snmptrap -h 2000:1:1:1:209:6bff:feae:6d67 -o 1.3.6.1.4.1.2.6.191.1.6.1.0 -m hello world
```

Files

Item	Description
/etc/snmpdv3.conf	Contains the configuration file for the SNMP version 3 agent.
/etc/snmpd.conf	Contains the configuration file for the SNMP version 1 agent.

snmpv3_ssw Command

Purpose

Switch the symbolic links among the non-encrypted **snmpdv3** agent, encrypted **snmpdv3** agent and **snmpdv1** agent.

Syntax

snmpv3_ssw [**-e** | **-n** | **-1**]

Description

Switch the symbolic links among the non-encrypted snmpdv3 agent, encrypted snmpdv3 agent and snmpdv1 agent, and then start the newly chosen SNMP agent. A user can choose which version of SNMP agent to run.

For example, if the current running SNMP agent is the encrypted **snmpdv3** agent, the actual SNMP agent executable which is running on the machine is **"/usr/sbin/snmpdv3e"**. The symbolic links on the machine are:

- /usr/sbin/snmpd --> /usr/sbin/snmpdv3e
- /usr/sbin/clsnmp --> /usr/sbin/clsnmpe

If a user chooses to switch to the non-encrypted snmpdv3 agent, after user runs the **/usr/sbin/snmpv3_ssw** command with the **-n** option, the actual snmp agent which is running on the machine **"/usr/sbin/snmpdv3ne"**. The symbolic links on the machine will be changed to:

- /usr/sbin/snmpd --> /usr/sbin/snmpdv3ne
- /usr/sbin/clsnmp --> /usr/sbin/clsnmpne

Flags

Item	Description
-e	Switch to the encrypted version of snmpdv3 agent.
-n	Switch to the non-encrypted version of snmpdv3 agent.
-1	Switch to the snmpdv1 agent.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To switch to the encrypted version of **snmpdv3** agent, enter:

```
/usr/sbin/snmp3_ssw -e
```

sno Command

Purpose

Provides a SNOBOL interpreter.

Syntax

sno [*File ...*]

Description

The **sno** command provides a SNOBOL compiler and interpreter, with some differences from standard SNOBOL. It reads the named files and the standard input and compiles all input through a statement containing the **end** label. The rest is available to the **syspit** pseudo-variable.

The **sno** command differs from standard SNOBOL in the following ways:

- There are no unanchored searches. To get the same effect, use lines similar to the following:

Item	Description
a ** b	Produces an unanchored search for <i>b</i> .
a ** b = x c	Produces an unanchored assignment.

- There is no back referencing.

```
x = "abc"
```

Item	Description
a ** x	Produces an unanchored search for abc .

- Function declaration is done at compile time by the use of the (non-unique) **define** label. Execution of a function call begins at the statement following the **define** label. Functions cannot be defined at run time, and the use of the name **define** is preempted. There is no provision for automatic variables other than parameters. For example:

```
define f()  
define f(a, b, c)
```

- All labels except **define** (even **end**), must have a nonempty statement.
- Labels, functions, and variables must all have distinct names. In particular, the nonempty statement on **end** cannot merely name a label.
- If **start** is a label in the program, program execution begins there. If not, execution begins with the first executable statement. The **define** label is not an executable statement.
- There are no built-in functions.
- Parentheses for arithmetic are not needed. Normal precedence applies. Because of this, the arithmetic operators \ (backslash) and * (asterisk) must be set off by spaces.
- The right side of assignments must be nonempty.
- Either ' (single quotation mark) or " (double quotation mark) can be used for literal quotation marks.
- The pseudo-variable **syspt** is not available.

Examples

To run the file `test.s` through the **sno** command and direct the output into the file `output`, enter:

```
sno < test.s > output
```

Files

Item	Description
<code>/usr/bin/sno</code>	Contains the sno command.

sntp Command for NTPv4

Purpose

The **sntp** command queries a Network Time Protocol (NTP) server and displays the offset time of the system clock with respect to the server clock.

Syntax

```
sntp [{-help -?}] [{-4 -6}] [-a keynum][-b bcaddress][-B bctimeout][-c][-d][-D debug-level][-g delay][-K kodfile][-k keyfile][-l logfile][-M steplimit][-o ntpver][-r][-S][-s][-u uctimeout][-wait][-version][address(es)]
```

Description

The **sntp** program is a Simple Network Time Protocol (SNTP) client that is used to query a Network Time Protocol (NTP) server. The **sntp** command displays the offset time of the system clock with respect to the server clock. If the root user of the system runs the **sntp** command, the **sntp** command corrects the system-offset time. The **sntp** command can be executed as an interactive command or from a script in a cron job. The **sntp** command implements the SNTP protocol that is defined in RFC 5905, including the full on-wire protocol. The **sntp** command does not provide the sanity checks, access controls, security functions, and mitigation algorithms as defined in the full NTP version 4 (NTPv4) specification that are also defined in RFC 5905.

By default, the **sntp** command displays the local date and time to the standard output in a format similar to the following format:

```
2011-08-04 00:40:36.642222 (+0000) +0.006611 +/- 0.041061 psp-os1 149.20.68.26 s3 no-leap
```

Where +0.006611 +/- 0.041061 indicates the time offset and error bound of the system clock with respect to the server clock, in seconds. The hostname, host IP address, stratum of the server, and leap indicator status are also displayed by the **sntp** command.

Flags

Table 22. Flags

Item	Description
-?, --help	Displays usage information.
-4, --ipv4	Forces DNS resolution of hostnames to the IP version 4 (IPv4) namespace.
-6, --ipv6	Forces DNS resolution of hostnames to the IP version 6 (IPv6) namespace.
-a <i>keynum</i>, --authentication <i>keynum</i>	Enables authentication with the key ID that is specified by the <i>keynum</i> variable. The key ID is a number that is specified in the key file along with an authentication password. For more information about the key file, see the -k option.
-b <i>bcaddress</i>, --broadcast <i>bcaddress</i>	Listens for the NTP packets that are sent to the broadcast or multicast address that is specified by the <i>bcaddress</i> variable. The broadcast or multicast address is a DNS name or IP address. The default maximum wait time to listen for broadcast or multicast NTP messages is 68 seconds. The wait time to listen for broadcast or multicast NTP messages can be modified by using the -B option.
-B <i>bctimeout</i>, --bctimeout <i>bctimeout</i>	Sets the wait time to listen for the broadcast or multicast NTP messages. The wait time is specified by the <i>bctimeout</i> variable. The default wait time is 68 seconds, as the ntpd command transmits broadcast or multicast NTP messages every 64 seconds.

Table 22. Flags (continued)

Item	Description
-c, --concurrent	Concurrently queries all the addresses that are returned for hostname resolution. Requests from an NTP client to a single server can be sent only once every two seconds. By default, all the addresses that are resolved from a single hostname are assumed to be for a single instance of the ntpd command. The sntp command sends queries to the addresses that are resolved from a single hostname, one after another, waiting for two seconds between the queries. The sntp command can send concurrent queries because the multiple addresses that are returned for a hostname are on different machines.
-d, --debug-level	Increases the debug verbosity level by one. The -d option can be specified multiple times. You can also set the debug verbosity level to a specific level by using the -D option.
-D debug-level, --set-debug-level debug-level	Sets the debug verbosity level to the level specified by the <i>debug-level</i> variable. The default debug verbosity level is zero. You can also use the -d option to increase the debug verbosity level by one.
-g delay, --gap delay	Sets the delay time between two outgoing queries. The delay time is specified by the <i>delay</i> variable, in milliseconds. The default delay time is 50 ms. The sntp command sends queries to all the provided hostnames or addresses in short succession. By default, the sntp command terminates once the first valid response is received for the queries sent. With the multiple number of time sources, only one valid response for the queries that are sent by the sntp command is used. To limit the number of queries whose responses are not used, each query is separated from the preceding one by the specified delay time. The delay between two queries gives time to receive a response for the earlier query. A larger delay time between two queries reduces the query load on the time sources by increasing the time to receive a valid response if the first time source attempted is slow or unreachable.
-k keyfile, --keyfile keyfile	Specifies the key file name that must be used with the -a option. The key file name is specified with the <i>keyfile</i> variable. For more information about the format of the key file, see the ntp-keygen Command for NTPv4 topic.
-K kodfile, --kod kodfile	Specifies the file path that must be used for the persistent history of kiss-of-death (KoD) or rate-limiting responses that are received from the servers. The file path is specified by the <i>kodfile</i> variable. The default file path is <code>/var/db/ntp-kod</code> . If the file specified by the <i>kodfile</i> variable does not exist, a warning message is displayed. However, a new file is not created.
-l logfile, --filelog logfile	Specifies the file name to which a copy of the status message is appended. The status message also appears on the terminal.

Table 22. Flags (continued)

Item	Description
-M <i>steplimit</i> , --steplimit <i>steplimit</i>	Specifies the offset limit in milliseconds to decide whether to correct the time by slewing the clock or by setting the clock to the correct time. If the offset is less than the offset limit specified by the <i>steplimit</i> variable, the time is corrected by slewing the clock by using the adjtime subroutine. If the offset is equal to or greater than the offset limit specified by the <i>steplimit</i> variable, the time is corrected by setting the clock to the correct time.
-o <i>ntpver</i> , --ntpversion <i>ntpver</i>	Specifies the NTP protocol version number to include in the requests. The NTP protocol version number is specified by the <i>ntpver</i> variable. The default value is 4.
-r , --userreservedport	Directs the sntp command to use the reserved NTP port 123 to query the NTP server. When the -r option is not used, by default the sntp command uses a User Datagram Protocol (UDP) source port number that the operating system selects. The -r option requires a root user to run the sntp command. This feature helps to identify connectivity failures due to port-based firewalling that affects the ntpd command. The ntpd command always uses the NTP source port 123.
-s , --slew	Enables offset correction by slewing with the help of adjtime subroutine. This option changes the rate of the clock for a period long enough to accomplish the required offset correction. The -s option requires a root user to run the sntp command.
-S , --step	Enables offset correction by stepping where the clock is directly set to the correct time. The -S option requires a root user to run the sntp command.
-u <i>uctimeout</i> , --uctimeout <i>uctimeout</i>	Specifies the maximum time to wait for a unicast response before the sntp command terminates, in seconds. The maximum time to wait is specified by the <i>uctimeout</i> variable.
--wait	Causes the sntp command to wait for the responses of all the pending queries. When -s or -S options are not specified, the sntp command terminates after the first valid response is received. However, when --wait option is specified, the sntp command waits for all the pending queries to respond.
--version	Displays the version number of the sntp program and the date and time when it was compiled.

Parameters

Table 23. Parameters

Item	Description
<i>address</i>	NTP server address.

Exit status

Table 24. Exit Status

Item	Description
Zero	Successful completion.
Nonzero	An error occurred.

Security

Access Control

You must be a part of the system group to run this command.

Files

/usr/sbin/ntp4/sntp4

Contains the **sntp** daemon for NTPv4.

/usr/sbin/sntp-->/usr/sbin/ntp4/sntp4

The default symbolic link to the NTP version 4 binary file from the `/usr/sbin` directory.

Example

To get the time offset of the system clock relative to the server (9.41.254.24) clock, enter the following command:

```
sntp 9.41.254.24
```

The following output appears:

```
2009 Feb 25 12:28:38.00620 - 0.00679 +/- 0.31077 secs
```

sodebug Command

Purpose

Sets or unsets the socket debug flag (**SO_DEBUG** socket option) and trace level on sockets.

Syntax

```
sodebug [ -h ] [ -l [ level ] ] [ -p pid | -s sockaddr [ -t type ] ]
```

Description

The `sodebug` command sets, unsets, or lists the socket debug flag and trace level on active sockets.

If the socket debug flag (also known as the **SO_DEBUG** socket option) is set for a socket, the events on this socket can be traced using the `trace` command.

You can use the `-l` option to set the socket debug flag on sockets that already exist on a system. The `-l` option also sets the trace level for a given socket.

If the `sodebug` command is run without any options, the socket debug flag status and trace level for each active socket displays.

The `trace` and `trtp` commands collect information based on the trace level.

The following table describes the information collected based on the trace level for trace hook ID 25A (TCPDBG):

	min	normal	detail
tcp_debug data (td_time, td_act, td_ostate, td_tcb, family and td_req)		X	X
tcpcb header		X	X
Address of tcpcb		X	X
All tcpcb fields			X
Address of socket		X	X
All socket fields			X

You can also set or unset the socket debug flag and the trace level as described below:

1. The following command enables the socket debug flag for all sockets that are subsequently created on the system:

```
no -o sodebug=1
```

2. You can specify **|DEBUG[=level]** in the wait/nwait field of a service in inetd.conf to turn on socket debugging for a specific service. You can set the trace level to **min**, **normal**, or **detail**. If no level is specified, the default level is **normal**.
3. You can set socket debugging on or off for all subsequent sockets created by a process using the **sodebug_env** parameter of the no command and specifying **export SODEBUG=level** in a process environment. You can set the trace level to **min**, **normal**, or **detail**.

Flags

Item	Description
-h	Displays help for the sodebug command.
-l [level]	Specifies the trace level. Valid values for level are none , min , normal , and detail . If no level is specified, the default trace level is normal .
-p pid	Specifies the process ID of a process.
-s sockaddr	Specifies a socket by the socket address, the address of the socket's inpcb, or the address of the socket's tcpcb.
-t type	Specifies the type of address that is specified by the -s sockaddr option. Valid values are socket , inpcb , and tcpcb . The default value is socket .

Security

You must have root authority to run the **sodebug** command.

Examples

1. To list the debug flag and socket trace level for socket f100090002d0a800, type:

```
sodebug -s f100090002d0a800
```

The output is similar to the following example:

```
socket address : f100090002d0a800 , sodebug flag : 0 , trace level : none(0)
```

2. To set the trace level to normal and set the debug flag to 1, type:

```
sodebug -s f100090002d0a800 -l normal
```

The output is similar to the following example:

```
Setting new values for trace level and debug flag  
socket address : f100090002d0a800 , sodebug flag : 1 , trace level : normal(3)
```

soelim Command

Purpose

Processes **.so** requests in **nroff** command files.

Syntax

soelim [*File* ... | -]

Description

The **soelim** command reads specified files or standard input and performs inclusion specified by the **nroff** command and **troff** command requests of the form **.so filename** when the request appears at the beginning of input lines. Any combination of ASCII spaces and ASCII tab characters can follow the **.so** request and precede the file name. No characters should follow the file name.

The **soelim** command is useful because commands, such as the **tbl** command, do not normally perform file inclusions during processing.

When the - (minus sign) flag is specified, a file name corresponding to standard input is included.

Flag

Item	Description

-	Indicates a file name corresponding to standard input.

Note: Inclusion can be suppressed by using a ' (single quotation mark) instead of a . (period), as follows:

Parameter

Item	Description

<i>File</i>	Specifies files that the command performs inclusion on. The default is standard input.

```
'so /usr/share/lib/tmac/tmac.s
```

Example

Following is a sample usage of the **soelim** command:

```
soelim exum?.n | tbl | nroff -ms -Tlp | col -Tlp | pg
```

In this example, you use the **soelim** command to preprocess the file inclusion (**.so**) requests. The output is then passed to the **tbl** command. This makes it easier to place tables in separate files that can be included in forming a large document.

soestat Command

Purpose

Displays Serial over Ethernet (SoE) device driver information and statistics information of various SoE device drivers.

Syntax

```
soestat [-a] | [-d sa_device_name [-p tty_device]] [-r] [-v]
```

Description

The **soestat** command symbolically displays the driver information and statistics information of various SoE adapter (sa) serial devices and the teletype (tty) devices by using the SoE device driver.

Flags

- a**
Displays statistics information for all sa serial devices and tty devices.
- d *sa_device_name***
Displays statistics information for a specific sa device and all tty devices that are configured for the specified sa device.
- p *tty_device***
Displays statistics information for a specific sa device and specific tty device.
- r**
Resets all statistic information to their initial values.
- v**
Verbose mode. Displays additional debug information.

Location

/usr/sbin/soestat

Files

- /usr/sbin/soestat**
Contains the **soestat** command.
- /usr/lib/drivers/soedd**
Contains the SoE kernel extension.

Security

You can run this command with all flags except the **-r** flag, which resets the statistic information. You must be either a root, or have aix.device.monitor.tty authority to use the **-r** flag.

Examples

1. To display information about all SoE devices, enter:

```
# soestat -a
```


2. To display information about the sa0 SoE Adapter only, enter:

```
# soestat -d sa0
```

3. To display information about tty1 tty (teletype) device, which is connected to the sa0 SoE adapter, enter:

```
# soestat -d sa0 -p tty1
```

4. To reset all statistic information, enter:

```
# soestat -ar
```

5. To reset information of the tty1 tty (teletype) device, which is connected to the sa0 SoE Adapter, enter:

```
# soestat -r -d sa0 -p tty1
```

sort Command

Purpose

Sorts files, merges files that are already sorted, and checks files to determine if they have been sorted.

Syntax

```
sort [ -A ] [ -b ] [ -c ] [ -d ] [ -f ] [ -i ] [ -m ] [ -n ] [ -r ] [ -u ] [ -o OutFile ] [ -t Character ] [ -T Directory ]  
[ -y [ Kilobytes ] ] [ -z RecordSize ] [ [ + [ FSkip ] [ .CSkip ] ] [ b ] [ d ] [ f ] [ i ] [ n ] [ r ] ] [ - [ FSkip ] [ .CSkip ] ] [ b ]  
[ d ] [ f ] [ i ] [ n ] [ r ] ] ... [ -k KeyDefinition ] ... [ File ... ]
```

Description

The **sort** command sorts lines in the files specified by the *File* parameter and writes the result to standard output. If the *File* parameter specifies more than one file, the **sort** command concatenates the files and sorts them as one file. A **-** (minus sign) in place of a file name specifies standard input. If you do not specify any file names, the command sorts standard input. An output file can be specified with the **-o** flag.

If no flags are specified, the **sort** command sorts entire lines of the input file based upon the collation order of the current locale.

Sort Keys

A sort key is a portion of an input line that is specified by a field number and a column number. Fields are parts of input lines that are separated by field separators. The default field separator is a sequence of one or more consecutive blank characters. However, these blank characters are considered to be a part of the following field for sorting purposes. You can specify the **-b** option to ignore these leading blank characters. A different field separator can be specified using the **-t** flag. The tab and the space characters are the blank characters in the C and English Language locales.

When using sort keys, the **sort** command first sorts all lines on the contents of the first sort key. Next, all the lines whose first sort keys are equal are sorted upon the contents of the second sort key, and so on. Sort keys are numbered according to the order they appear on the command line. If two lines sort equally on all sort keys, the entire lines are then compared based upon the collation order in the current locale.

When numbering columns within fields, the blank characters in a default field separator are counted as part of the following field. Field separator characters specified by the **-t** flag are not counted as parts of fields. Leading blank characters can be ignored using the **-b** flag.

Sort keys can be defined using the following two methods:

- **-k** *KeyDefinition*
- *FSkip.CSkip* (obsolescent version).

Sort Key Definition Using the -k Flag

The **-k** *KeyDefinition* flag uses the following form:

-k [*FStart* [*CStart*]] [*Modifier*] [, [*FEnd* [*CEnd*]] [*Modifier*]]

The sort key includes all characters beginning with the field specified by the *FStart* variable and the column specified by the *CStart* variable and ending with the field specified by the *FEnd* variable and the column specified by the *CEnd* variable. If *Fend* is not specified, the last character of the line is assumed. If *CEnd* is not specified the last character in the *FEnd* field is assumed. Any field or column number in the *KeyDefinition* variable may be omitted. The default values are:

Item	Description
<i>FStart</i>	Beginning of the line
<i>CStart</i>	First column in the field
<i>FEnd</i>	End of the line
<i>CEnd</i>	Last column of the field

If there is any space between the fields, **sort** considers them as separate fields.

The value of the *Modifier* variable can be one or more of the letters **b**, **d**, **f**, **i**, **n**, or **r**. The modifiers apply only to the field definition they are attached to and have the same effect as the flag of the same letter. The modifier letter **b** applies only to the end of the field definition to which it is attached. For example:

```
-k 3.2b,3r
```

specifies a sort key beginning in the second nonblank column of the third field and extending to the end of the third field, with the sort on this key to be done in reverse collation order. If the *FStart* variable and the *CStart* variable fall beyond the end of the line or after the *FEnd* variable and the *CEnd* variable, then the sort key is ignored.

A sort key can also be specified in the following manner:

[**+**[*FSkip1*] [*CSkip1*] [*Modifier*]] [**-**[*FSkip2*] [*CSkip2*] [*Modifier*]]

The **+FSkip1** variable specifies the number of fields skipped to reach the first field of the sort key and the **+CSkip** variable specifies the number of columns skipped within that field to reach the first character in the sort key. The **-FSkip** variable specifies the number of fields skipped to reach the first character *after* the sort key, and the **-CSkip** variable specifies the number of columns to skip within that field. Any of the field and column skip counts may be omitted. The defaults are:

Item	Description
<i>FSkip1</i>	Beginning of the line
<i>CSkip1</i>	Zero
<i>FSkip2</i>	End of the line
<i>CSkip2</i>	Zero

The modifiers specified by the *Modifier* variable are the same as in the **-k** flag key sort definition.

The field and column numbers specified by **+FSkip1.CSkip1** variables are generally one less than the field and column number of the sort key itself because these variables specify how many fields and columns to skip before reaching the sort key. For example:

```
+2.1b -3r
```

specifies a sort key beginning in the second nonblank column of the third field and extending to the end of the third field, with the sort on this key to be done in reverse collation order. The statement **+2.1b** specifies that two fields are skipped and then the leading blanks and one more column are skipped. If the

+FSkip1.CSkip1 variables fall beyond the end of the line or after the **-FSkip2.CSkip2** variables, then the sort key is ignored.

Note: The maximum number of fields on a line is 32.

Flags

Note: A **-b**, **-d**, **-f**, **-i**, **-n**, or **-r** flag that appears before any sort key definition applies to all sort keys. None of the **-b**, **-d**, **-f**, **-i**, **-n**, or **-r** flags may appear alone after a **-k** *KeyDefinition*; if they are attached to a *KeyDefinition* variable as a modifier, they apply only to the attached sort key. If one of these flags follows a **+FSkip.Cskip** or **-FSkip.Cskip** sort key definition, the flag only applies to that sort key.

Item	Description
-A	Sorts on a byte-by-byte basis using ASCII collation order instead of collation in the current locale.
-b	Ignores leading spaces and tabs to find the first or last column of a field.
-c	Checks that input is sorted according to the ordering rules specified in the flags. A nonzero value is returned if the input file is not correctly sorted.
-C	Checks that input is sorted according to the ordering rules specified in the flags except that a warning message shall not be sent to standard error if there is a disorder or, with -u option, a duplicate key is detected.
-d	Sorts using dictionary order. Only letters, digits, and spaces are considered in comparisons.
-f	Changes all lowercase letters to uppercase before comparison.
-i	Ignores all nonprinting characters during comparisons.
-k <i>KeyDefinition</i>	Specifies a sort key. The format of the <i>KeyDefinition</i> option is: <pre>[FStart [.CStart]] [Modifier] [, [FEnd [.CEnd]] [Modifier]]</pre> The sort key includes all characters beginning with the field specified by the <i>FStart</i> variable and the column specified by the <i>CStart</i> variable and ending with the field specified by the <i>FEnd</i> variable and the column specified by the <i>CEnd</i> variable. The value of the <i>Modifier</i> variable can be b, d, f, i, n, or r. The modifiers are equivalent to the flags of the same letter. When a modifier is attached to a key definition, then no flag is applied to it.
-m	Merges multiple input files only; the input are assumed to be already sorted.
-n	Sorts numeric fields by arithmetic value. A numeric field may contain leading blanks, an optional minus sign, decimal digits, thousands-separator characters, and an optional radix character. Numeric sorting of a field containing any nonnumeric character gives unpredictable results.
-o <i>OutFile</i>	Directs output to the file specified by the <i>OutFile</i> parameter instead of standard output. The value of the <i>OutFile</i> parameter can be the same as the <i>File</i> parameter.
-r	Reverses the order of the specified sort.
-t <i>Character</i>	Specifies <i>Character</i> as the single field separator character.
-u	Suppresses all but one line in each set of lines that sort equally according to the sort keys and options.
-T <i>Directory</i>	Places all temporary files that are created into the directory specified by the <i>Directory</i> parameter.

Item	Description
-y [<i>Kilobytes</i>]	Starts the sort command using the number of kilobytes of main storage specified by the <i>Kilobytes</i> parameter and adds storage as needed. (If the value specified in the <i>Kilobytes</i> parameter is less than the minimum storage site or greater than the maximum, the minimum or maximum is used instead). If the -y flag is omitted, the sort command starts with the default storage size. The -y0 flag starts with minimum storage, and the -y flag (with no <i>Kilobytes</i> value) starts with maximum storage. The amount of storage used by the sort command affects performance significantly. Sorting a small file in a large amount of storage is wasteful.
-z <i>RecordSize</i>	Prevents abnormal termination if any of the lines being sorted are longer than the default buffer size. The default buffer size is 20 KB. When the -c or -m flags are specified, the sorting phase is omitted and a system default buffer size is used. If sorted lines are longer than this size, the sort command terminates abnormally. The -z option specifies recording of the longest line in the sort phase so adequate buffers can be allocated in the merge phase. The <i>RecordSize</i> variable must designate a value in bytes equal to or greater than the longest line to be merged. The longest line size that is supported under C locale is approximately 2M characters and the longest line size that is supported under non-C locale is 1M characters. The -z option is ineffective under C locale.

Exit Status

This command returns the following exit values:

Item	Description
0	All input files were output successfully, or -c was specified and the input file was correctly sorted.
1	Under the -c option, the file was not ordered as specified, or if the -c and -u options were both specified, two input lines were found with equal keys.
>1	An error occurred.

Examples

1. To sort the `fruits` file with the **LC_ALL**, **LC_COLLATE**, or **LANG** environment variable set to `En_US`, enter:

```
LANG=En_US sort fruits
```

This command sequence displays the contents of the `fruits` file sorted in ascending lexicographic order. The characters in each column are compared one by one, including spaces, digits, and special characters. For instance, if the `fruits` file contains the text:

```
banana
orange
Persimmon
apple
%%banana
apple
ORANGE
```

the **sort** command displays:

```
%%banana
ORANGE
Persimmon
apple
apple
```

```
banana
orange
```

In the ASCII collating sequence, the % (percent sign) precedes uppercase letters, which precede lowercase letters. If your current locale specifies a character set other than ASCII, your results may be different.

2. To sort in dictionary order, enter:

```
sort -d fruits
```

This command sequence sorts and displays the contents of the `fruits` file, comparing only letters, digits, and spaces. If the `fruits` file is the same as in example 1, then the **sort** command displays:

```
ORANGE
Persimmon
apple
apple
%%banana
banana
orange
```

The **-d** flag ignores the % (percent sign) character because it is not a letter, digit, or space, placing %%banana with banana.

3. To group lines that contain uppercase and special characters with similar lowercase lines, enter:

```
sort -d -f fruits
```

The **-d** flag ignores special characters and the **-f** flag ignores differences in case. With the **LC_ALL**, **LC_COLLATE**, or **LANG** environment variable set to C, the output for the `fruits` file becomes:

```
apple
apple
%%banana
banana
ORANGE
orange
Persimmon
```

4. To sort, removing duplicate lines, enter:

```
sort -d -f -u fruits
```

The **-u** flag tells the **sort** command to remove duplicate lines, making each line of the file unique. This command sequence displays:

```
apple
%%banana
ORANGE
Persimmon
```

Not only is the duplicate `apple` removed, but `banana` and `ORANGE` as well. These are removed because the **-d** flag ignores the %% special characters and the **-f** flag ignores differences in case.

5. To sort as in example 4, removing duplicate instances unless capitalized or punctuated differently, enter:

```
sort -u +0 -d -f +0 fruits
```

Entering the `+0 -d -f` does the same type of sort that is done with `-d -f` in example 3. Then the `+0` performs another comparison to distinguish lines that are not identical. This prevents the **-u** flag from removing them.

Given the `fruits` file shown in example 1, the added `+0` distinguishes %%banana from banana and ORANGE from orange. However, the two instances of `apple` are identical, so one of them is deleted.

```
apple
%%banana
banana
```

```
ORANGE
orange
Persimmon
```

6. To specify the character that separates fields, enter:

```
sort -t: +1 vegetables
```

This command sequence sorts the `vegetables` file, comparing the text that follows the first colon on each line. The `+1` tells the **sort** command to ignore the first field and to compare from the start of the second field to the end of the line. The `-t:` flag tells the **sort** command that colons separate fields. If `vegetables` contains:

```
yams:104
turnips:8
potatoes:15
carrots:104
green beans:32
radishes:5
lettuce:15
```

Then, with the **LC_ALL**, **LC_COLLATE**, or **LANG** environment variable set to C, the **sort** command displays:

```
carrots:104
yams:104
lettuce:15
potatoes:15
green beans:32
radishes:5
turnips:8
```

Note that the numbers are not in numeric order. This happened when a lexicographic sort compares each character from left to right. In other words, 3 comes before 5, so 32 comes before 5.

7. To sort numbers, enter:

```
sort -t: +1 -n vegetables
```

This command sequence sorts the `vegetables` file numerically on the second field. If the `vegetables` file is the same as in example 6, then the **sort** command displays:

```
radishes:5
turnips:8
lettuce:15
potatoes:15
green beans:32
carrots:104
yams:104
```

8. To sort more than one field, enter:

```
sort -t: +1 -2 -n +0 -1 -r vegetables
```

OR

```
sort -t: -k2,2 n -k1,1 r vegetables
```

This command sequence performs a numeric sort on the second field (`+1 -2 -n`). Within that ordering, it sorts the first field in reverse alphabetic order (`+0 -1 -r`). With the **LC_ALL**, **LC_COLLATE**, or **LANG** environment variable set to C, the output looks like this:

```
radishes:5
turnips:8
potatoes:15
lettuce:15
green beans:32
yams:104
carrots:104
```

The command sorts the lines in numeric order. When two lines have the same number, they appear in reverse alphabetic order.

9. To replace the original file with the sorted text, enter:

```
sort -o vegetables vegetables
```

This command sequence stores the sorted output into the vegetables file (-o vegetables).

Files

Item	Description
/usr/bin/sort	Contains the sort command.

Item	Description
/var/tmp	Temporary space during the sort command processing.
/usr/tmp	Temporary space during the sort command processing, if file cannot be created in /var/tmp .
/tmp	Temporary space during the sort command processing, if file cannot be created in /var/tmp or /usr/tmp .

sortbib Command

Purpose

Sorts a bibliographic database.

Syntax

```
sortbib [ -sKeys ] [ Database ... ]
```

Description

The **sortbib** command sorts files of records containing **refer** command key letters by user-specified keys. The records can be separated by blank lines, or enclosed by the **.[** (period, left bracket) and the **.]** (period, right bracket) delimiters, but the two styles cannot be mixed together. The **sortbib** command reads through each database specified by the *Database* parameter and pulls out key fields, which are sorted separately. The sorted key fields contain the file pointer, byte offset, and length of corresponding records. These records are delivered using disk seeks and reads, so the **sortbib** command cannot be used in a pipeline to read standard input.

By default, the **sortbib** command alphabetizes by the first **%A** and **%D** fields, which contain the senior author and date.

The **sortbib** command sorts by the last word in the **%A** field, which is assumed to be the author's last name. A word in the final position, such as **j r .** or **ed .**, is ignored if the name preceding ends with a comma. Authors with two-word last names, or names with uncommon constructions, can be sorted correctly by using the **nroff** command convention **\O** in place of a space character. Specifying the **%Q** field is similar to the **%A** field, except sorting begins with the first, not the last, word.

Note: Records with missing author fields should be sorted by title.

The **sortbib** command sorts by the last word of the **%D** line, which is usually the year. It ignores leading articles when sorting by titles in the **%T** or **%J** fields. The articles ignored are specific to the locale and specified in the locale-specific **refer message catalog**. Within this catalog, the articles are contained in a single message. Each article is separated by any number of ASCII space or tab characters. If a sort-significant field is absent from a record, the **sortbib** command places the record before other records containing that field.

No more than 16 databases can be sorted together at one time. Records longer than 4096 characters are truncated.

The *Database* parameter contains **refer** command key letters by user-specified keys that the **sortbib** command sorts through.

Flags

Item	Description
-sKeys	Specifies field keys to sort on.

Examples

1. To sorts by author, title, and date:

```
sortbib -sATD Database
```

2. To sort by author and date:

```
sortbib -sA+D Database
```

Files

Item	Description
/tmp/SbibXXXXX	Contains the temporary file.
/usr/bin/sort	Contains the sort command.

sortm Command

Purpose

Sorts messages.

Syntax

sortm [**+Folder**] [*Messages*] [**-datefield** *Field*] [**-noverbose** | **-verbose**]

Description

The **sortm** command sorts messages according to their **Date :** field and renumbers them consecutively beginning with number one. Messages that are in the folder, but not specified to be sorted, are placed after the sorted messages. The **sortm** command displays a message if it cannot parse a date field.

To specify a field other than the **Date :** field, specify the **-datefield** flag. If you specify a folder, it becomes the current folder. The current message remains the current message for the specified folder, even if it moves during the sort.

Flags

Item	Description
-datefield <i>Field</i>	Specifies the header field to be used in the sort. The Date : field is the default.
+Folder	Specifies the folder with messages to be sorted. The default is the current folder.

Item	Description
-help	Lists the command syntax, available switches (toggles), and version information. Note: For Message Handler (MH), the name of this flag must be fully spelled out.
<i>Messages</i>	Specifies the messages to be sorted. Use the following references to specify messages: Number Number of the message. Sequence A group of messages specified by the user. Recognized values are: all All messages in a folder. This is the default. cur or . (period) Current message. first First message in a folder. last Last message in a folder. next Message following the current message. prev Message preceding the current message.
-noverbose	Prevents display of information during the sort. This flag is the default.
-verbose	Displays information during the sort. This information allows you to monitor the steps involved.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Profile Entries

The following entries are found in the *UserMhDirectory/.mh_profile* file:

Item	Description
Current-Folder:	Sets the default current folder.
Path:	Specifies the <i>UserMhDirectory</i> .

Examples

1. To sort all the messages in the current folder according to the date, enter:

```
sortm
```

2. To sort messages 5 through 10 in the `easter` folder according to the date, enter:

```
sortm +easter 5-10
```

Files

Item	Description
\$HOME/.mh_profile	Contains the MH user profile.
/usr/bin/sortm	Contains the sortm command.

spell Command

Purpose

Finds English Language spelling errors.

Syntax

spell [**-b**] [**-i**] [**-l**] [**-v**] [**-x**] [**-d** *HashList*] [**-h** *HistoryList*] [**-s** *HashStop*] [**+** *WordList*] [*File ...*]

Description

The **spell** command reads words in the file indicated by the *File* variable and compares them to those in a spelling list. Words that cannot be matched in the spelling list or derived from words in the spelling list (by applying certain inflections, prefixes, and suffixes) are written to standard output. If no file name is specified, the **spell** command reads standard input.

The **spell** command ignores the same **troff**, **tbl**, and **eqn** codes as does the **deroff** command.

The coverage of the spelling list is uneven. You should create your own dictionary of special words used in your files. Your dictionary is a file containing a sorted list of words, one per line. To create your dictionary, use the **spellin** command.

Files containing an alternate spelling list, history list, and stop list can be specified by file name parameters following the **-d**, **-f**, and **-h** flags. Copies of all output can be accumulated in the history file.

Three programs help maintain and check the hash lists used by the **spell** command:

Item	Description
/usr/lbin/spell/hashmake	Reads a list of words from standard input and writes the corresponding 9-digit hash code to standard output.
/usr/bin/spellin <i>Number</i>	Reads the specified <i>Number</i> of hash codes from standard input and writes a compressed spelling list to standard output.
/usr/lbin/spell/hashcheck <i>SpellingList</i>	Reads a compressed <i>SpellingList</i> , recreates the 9-digit hash codes for all the words in it, and writes these codes to standard output.

The *File* parameter specifies the files that the **spell** command reads and compares them with the spelling list. If no file is specified, the command reads standard input.

Flags

Item	Description
-b	Checks British spelling. However, this flag does not provide a reasonable prototype for British spelling. The algorithms to derive a match against the spelling dictionary by applying certain inflections, prefixes, and suffixes are based on American English spelling.
-d <i>HashList</i>	Specifies the <i>HashList</i> file as the alternative spelling list. The default is <u>/usr/share/dict/hlist[ab]</u> .
-h <i>HistoryList</i>	Specifies the <i>HistoryList</i> file as the alternative history list, which is used to accumulate all output. The default is <u>/usr/sbin/spell/spellhist</u> . Note: The <i>HistoryList</i> file must be an existing file with read and write permissions.
-i	Suppresses processing of include files.
-l	Follows the chain of all include files (.so and .nx formatting commands). Without this flag, the spell command follows chains of all include files except for those beginning with /usr/lib .
-s <i>HashStop</i>	Specifies the <i>HashStop</i> file as the alternative stop list, which is used to filter out misspellings that would otherwise pass. The default is <u>/usr/share/dict/hstop</u> .
-v	Displays all words not in the spelling list and indicates possible derivations from the words.
-x	Displays every possible word stem with an = (equal sign).
+ <i>WordList</i>	Checks <i>WordList</i> for additional word spellings. <i>WordList</i> is the name of a file you provide that contains a sorted list of words, one per line. With this flag, you can specify a set of correctly spelled words (in addition to the spell command's own spelling list) for each job.

Exit Status

The following exit values are returned:

Item	Description
0	Indicates successful completion.
>0	Indicates an error occurred.

Examples

1. To check your spelling, enter:

```
spell chap1 >mistakes
```

This creates a file named `mistakes` containing all the words found in `chap1` that are not in the system spelling dictionary. Some of these may be correctly spelled words that the **spell** command does not recognize. Save the output of the **spell** command in a file because the word list may be long.

2. To check British spelling, enter:

```
spell -b chap1 >mistakes
```

This checks `chap1` against the British dictionary and writes the questionable words in the `mistakes` file.

3. To see how the **spell** command derives words, enter:

```
spell -v chap1 >deriv
```

This lists words not found literally in the dictionary but are derived from forms of dictionary words. The prefixes and suffixes used to form the derivations are indicated for each word. Words that are not in the dictionary at all are also listed.

4. To check your spelling against an additional word list, enter:

```
spell +newwords chap1
```

This checks the spelling of words in chap1 against the system dictionary and against newwords. The newwords file lists words in alphabetical order, one per line. You can create this file with a text editor, such as the ed editor, and alphabetize it with the **sort** command.

Files

Item	Description
/usr/share/dict/hlist[ab]	Contains hashed spelling lists, both American and British.
/usr/share/dict/hstop	Contains a hashed stop list.
/usr/sbin/spell/spellhist	Contains a history file.
/usr/sbin/spell/compress	Contains an executable shell program to compress the history file.
/usr/sbin/spell/hashmake	Creates hash codes from a spelling list.
/usr/bin/spellin <i>Number</i>	Creates spelling list from hash codes.
/usr/sbin/spell/hashcheck <i>SpellingList</i>	Creates hash codes from a compressed spelling list.
/usr/sbin/spell/spellinprg	Main program called by the spellin file.
/usr/sbin/spell/spellprog	Checks spelling.

spellin Command

Purpose

Creates a spelling list.

Syntax

```
spellin [ List | Number ]
```

Description

The **spellin** command creates a spelling list for use by the **spell** command. The parameter for the **spellin** command can be a file name or a number. The **spellin** command combines the words from the standard input and the already existing spelling list file and places a new spelling list on the standard output. If no list file is specified, a new list is created. If *Number* is specified, the **spellin** command reads the specified number of hash codes from standard input and writes a compressed spelling list.

Examples

To add the word hookey to the spelling list named myhlist, enter:

```
echo hookey | spellin /usr/share/dict/hlista > myhlist
```

spellout Command

Purpose

Verifies that a word is not in the spelling list.

Syntax

spellout [**-d**] *List*

Description

The **spellout** command looks up each word from standard input and prints on standard output those that are missing from the hashed list file specified by the *List* parameter. The hashed list file is similar to the dictionary file used by the **spell** command.

Flags

Item	Description

-d	Prints those words that are present in the hashed list file.

Examples

To verify that the word hookey is not on the default spelling list, enter:

```
echo hookey | spellout /usr/share/dict/hlista
```

In this example, the **spellout** command prints the word hookey on standard output if it is not in the hashed list file. With the **-d** flag, **spellout** prints the word hookey if it is found in the hash file.

splat Command

Purpose

Simple Performance Lock Analysis Tool (splat). Provides kernel and **pthread** lock usage reports.

Syntax

splat **-i** *file* [**-n** *file*] [**-o** *file*] [**-d** [*bfta*]] [**-l** *address*] [**-c** *class*] [**-s** [*acelmsS*]] [**-C** *cpus*] [**-S** *count*] [**-t** *start*] [**-T** *stop*] [**-p**]

splat **-h** [*topic*]

splat **-j**

Description

splat (Simple Performance Lock Analysis Tool) is a software tool which post-processes AIX trace files to produce kernel simple and complex lock usage reports. It also produces **pthread** mutex read-write locks, and condition variables usage reports.

Flags

Item	Description
-i <i>inputfile</i>	AIX trace file (REQUIRED).
-n <i>namefile</i>	File containing output of gensyms command.
-o <i>outputfile</i>	File to write reports to (DEFAULT: stdout).
-d <i>detail</i>	Detail can be one of: [b] asic: summary and lock detail (DEFAULT) [f] unction: basic + function detail [t] hread: basic + thread detail [a] ll: basic + function + thread detail
-c <i>class</i>	If the user supplies a decimal lock class index, splat will only report activity for locks in that class.
-l <i>address</i>	If the user supplies a hexadecimal lock address, splat will only report activity for the lock at that address. splat will filter a trace file for lock hooks containing that lock address and produce a report solely for that lock.
-s <i>criteria</i>	Sort the lock, function, and thread reports by the following criteria: a acquisitions c percent processor hold time e percent elapsed hold time l lock address, function address, or thread ID m miss rate s spin count S percent processor spin hold time (DEFAULT) w percent real wait time W average waitq depth
-C <i>cpus</i>	Specify the number of processors present for this trace.
-S <i>count</i>	The maximum number of entries in each report (DEFAULT: 10).
-t <i>starttime</i>	Time offset in seconds from the beginning of the trace.
-T <i>stoptime</i>	Time offset in seconds from the beginning of the trace to stop analyzing trace data. (DEFAULT: the end of the trace.)
-h [<i>topic</i>]	Help on usage or a specific topic. Valid topics are: • all • overview • input • names • reports • sorting
-j	Print a list of trace hooks used by splat.

Item	Description
-p	Specifies the use of the PURR register to calculate processor times.

Help

The following is a list of available help topics and a brief summary of each:

Item	Description
OVERVIEW	This text.
INPUT	AIX trace hooks required in order to acquire useful output from splat .
NAMES	What name utilities can be used to cause splat to map addresses to human-readable symbols.
REPORTS	A description of each report that splat can produce and the formulas used to calculate reported values.
SORTING	A list of all the available sorting options and how they are applied to splat 's output.

Splat Trace

Splat takes as primary input an AIX trace file which has been collected with the AIX trace command. Before analyzing a trace with **splat**, you will need to make sure that the trace is collected with an adequate set of hooks, including the following:

```
106 DISPATCH
10C DISPATCH IDLE PROCESS
10E RELOCK
112 LOCK
113 UNLOCK
134 HKWD_SYSC_EXECVE
139 HKWD_SYSC_FORK
419 CPU_PREEMPT
465 HKWD_SYSC_CRTTHREAD
46D WAIT_LOCK
46E WAKEUP_LOCK
606 HKWD_PTHREAD_COND
607 HKWD_PTHREAD_MUTEX
608 HKWD_PTHREAD_RWLOCK
609 HKWD_PTHREAD_GENERAL
```

Capturing these lock and unlock trace events can cause serious performance degradation due to the frequency that locks are used in a multiprocessor environment. Therefore, lock trace event reporting is normally disabled. In order to enable lock trace event reporting, the following steps must be taken before a trace can be collected which will include lock trace events that splat requires (KornShell syntax):

```
1. bosboot -ad /dev/hdisk0 -L
2. shutdown -Fr
3. (reboot the machine)
4. locktrace -S
5. mkdir temp.lib; cd temp.lib
6. ln -s /usr/ccs/lib/perf/libpthreads.a
7. export LIBPATH=$PWD:$LIBPATH
```

Steps 1 through 3 are optional. They enable the display of kernel lock class names instead of addresses. Please refer to **bosboot(1)** for more information on **bosboot** and its flags. Steps 5 through 7 are necessary for activating the user **pthread** lock instrumentation; the **temp.lib** subdirectory can be put anywhere. Steps 1 through 7 are necessary in order for the report to be complete.

Splat Names

Splat can take the output of **gensyms** as an optional input and use it to map lock and function addresses to human-readable symbols.

Lock classes and **offsets** can be used to identify a lock broadly, but not as specifically as the actual symbol.

Splat Reports

The report generated by **splat** consists of a report summary, a lock summary report section, and a list of lock detail reports, each of which may have an associated function detail and/or thread detail report.

Report Summary ^^^^^^^^^^^^^^^^

The report summary consists of the following elements:

- The trace command used to collect the trace.
- The host that the trace was taken on.
- The date that the trace was taken on.
- The duration of the trace in seconds.
- The estimated number of CPUs
- The combined elapsed duration of the trace in seconds;
(the duration of the trace multiplied by the number of CPUs identified during the trace).
- Start time, which is the offset in seconds from the beginning of the trace that trace statistics begin to be gathered.
- Stop time, which is the offset in seconds from the beginning of the trace that trace statistics stop being gathered.
- Total number of acquisitions during the trace.
- Acquisitions per second, which is computed by dividing the total number of lock acquisitions by the real-time duration of the trace.
- % of Total Spin Time, this is the summation of all lock spin hold times, divided by the combined trace duration in seconds, divided by 100. The current goal is to have this value be less than 10% of the total trace duration.

Lock Summary ^^^^^^^^^^^^^^^^

The lock summary report has the following fields:

Lock	The name, lockclass or address of the lock.
Type	The type of the lock, identified by one of the following letters: Q A RunQ lock S A simple kernel lock D A disabled simple kernel lock C A complex kernel lock M A PThread mutex V A PThread condition-variable L A PThread read/write lock
Acquisitions	The number of successful lock attempts for this lock, minus the number of times a thread was preempted while holding this lock.
Spins	The number of unsuccessful lock attempts for this lock, minus the number of times a thread was undispached while spinning.
Wait or Transform	The number of unsuccessful lock attempts that resulted in the attempting thread going to sleep to wait for the lock to become available, or allocating a krlock.
%Miss	Spins divided by Acquisitions plus Spins, multiplied by 100.
%Total	Acquisitions divided by the total number of all lock acquisitions, multiplied by 100.
Locks/CSec	Acquisitions divided by the combined elapsed duration in seconds.
Percent HoldTime Real CPU	The percent of combined elapsed trace time that

	threads held the lock in question while dispatched. DISPATCHED_HOLDTIME_IN_SECONDS divided by combined trace duration, multiplied by 100.
Real Elaps(ed)	The percent of combined elapsed trace time that threads held the lock while dispatched or sleeping. UNDISPATCHED_AND_DISPATCHED_HOLDTIME_IN_SECONDS divided by combined trace duration, multiplied by 100.
Comb Spin	The percent of combined elapsed trace time that threads spun while waiting to acquire this lock. SPIN_HOLDTIME_IN_SECONDS divided by combined trace duration, multiplied by 100.

The lock summary report defaults to a list of ten locks, sorted in descending order by percent spin hold time (the tenth field). The length of the summary report can be adjusted using the **-S** switch. The sorted order of the summary report (and all other reports) can be set with the **-s** switch whose options are described in the SORTING help section, **splat -h** sorting.

```

Lock Detail
^^^^^^^^^^^^

The lock detail report consists of the following fields:

LOCK          The address (in hexadecimal) of the lock.
NAME          The symbol mapping for that address (if available)
CLASS         The lockclass name (if available) and hexadecimal offset,
              used to allocate this lock ( lock_alloc() kernel service ).
Parent Thread  Thread id of the parent thread. This field only exists for Mutex,
              Read/Write lock and Conditional Variable report.

creation time  Elapsed time in seconds after the first event recorded in trace,
              if available. This field only exists for Mutex, Read/Write lock
              and Conditional Variable report.
deletion time  Elapsed time in seconds after the first event recorded in trace,
              if available. This field only exists for Mutex, Read/Write lock
              and Conditional Variable report.

Pid           Pid number associated to the lock (this field only exists for Mutex,
              Read/Write lock and Conditional Variable report).

Process Name   Process name associated to the lock (this field only exists for Mutex,
              Read/Write lock and Conditional Variable report).

Call-Chain     Stack of called methods (if possible to have them, this field only
              exists for Mutex, Read/Write lock and Conditional Variable report).

Acquisitions   The number of successful lock attempts for this lock.
              This field is named Passes for the conditional variable lock report.

Miss Rate      The number of unsuccessful lock attempts divided by
              Acquisitions plus unsuccessful lock attempts, multiplied
              by 100.

Spin Count     The number of unsuccessful lock attempts.

Wait Count     The number of unsuccessful lock attempts that resulted in
              the attempting thread going to sleep to wait for the lock
              to become available.

Transform Count The number of krlock allocated and deallocated by the simple lock.

Busy Count     The number of simple_lock_try() calls that returned busy.

Seconds Held   The total time in seconds that this lock was held by
CPU            dispatched threads.

Elapsed        The total time in seconds that this lock was held by
              both dispatched and undispached threads.

NOTE: neither of these two values should exceed the
      total real elapsed duration of the trace.

```

Percent Held Real CPU	The percent of combined elapsed trace time that threads held the lock in question while dispatched. DISPATCHED_HOLDTIME_IN_SECONDS divided by trace duration, multiplied by 100.
Real Elaps(ed)	The percent of combined elapsed trace time that threads held the lock while dispatched or sleeping. UNDISPATCHED_AND_DISPATCHED_HOLDTIME_IN_SECONDS divided by trace duration, multiplied by 100.
Comb Spin	The percent of combined elapsed trace time that threads spun while waiting to acquire this lock. SPIN_HOLDTIME_IN_SECONDS divided by trace duration, multiplied by 100.
Wait	The percentage of combined elapsed trace time that threads unsuccessfully tried to acquire this lock.
SpinQ	Splat keeps track of the minimum, maximum and average depth of the spin queue (the threads spinning, waiting for a lock to become available).
WaitQ	As with the spin queue, splat also tracks the minimum, maximum and average depth of the queue of threads waited waiting for a lock to become available).
PROD	The associated krlocks prod calls count.
CONFER SELF	The confer to self calls count for the simple lock and the associated krlocks.
CONFER TARGET	The confer to target calls count for the simple lock and the associated krlocks. w/ preemption reports the successful calls count, resulting in a preemption.
CONFER ALL	The confer to all calls count for the simple lock and the associated krlocks. w/ preemption reports the successful calls count, resulting in a preemption.
HANDOFF	The associated krlocks handoff calls count.
Lock Activity w/Interrupts Enabled (mSecs)	

This section of the lock detail report are dumps of the raw data that splat collects for each lock, times expressed in milliseconds. The five states: LOCK, SPIN, WAIT, UNDISP(atched) and PREEMPT are the five basic states of **splat**'s enabled **simple_lock** finite state machine. The count for each state is the number of times a thread's actions resulted in a transition into that state. The durations in milliseconds show the minimum, maximum, average and total amounts of time that a lock request spent in that state.

LOCK: this state represents a thread successfully acquiring a lock.

SPIN: this state represents a thread unsuccessfully trying to acquire a lock.

WAIT: this state represents a spinning thread (in SPIN) going to sleep (voluntarily) after exceeding the thread's spin threshold.

UNDISP: this state represents a spinning thread (in SPIN) becoming undispached (involuntarily) before exceeding the thread's spin threshold.

PREEMPT: this state represents when a thread holding a lock is undispached.

Lock Activity w/Interrupts Disabled (mSecs)

This section of the lock detail report are dumps of the raw data that splat collects for each lock, times expressed in milliseconds. The six states: LOCK, SPIN, LOCK with KRLOCK, KRLOCK LOCK, KRLOCK SPIN and TRANSFORM are the six basic states of **splat**'s disabled **simple_lock** finite state machine. The count for each state is the number of times a thread's actions resulted in a transition into that state. The

durations in milliseconds show the minimum, maximum, average and total amounts of time that a lock request spent in that state.

LOCK: This state represents a thread successfully acquiring a lock.

SPIN: This state represents a thread unsuccessfully trying to acquire a lock.

LOCK with KRLOCK: The thread has successfully acquired the lock, while holding the associated krlock, and is currently executing.

KRLOCK LOCK: The thread has successfully acquired the associated krlock, and is currently executing.

KRLOCK SPIN: The thread is executing and unsuccessfully attempting to acquire the associated krlock.

TRANSFORM: The thread has successfully allocated a krlock it associates to, and is executing.

Function Detail ^^^^^^^^^^^^^^^^

The function detail report consists of the following fields:

Function Name	The name or return address of the function which used the lock.
Acquisitions	The number of successful lock attempts for this lock. For complex lock and read/write lock there is a distinction between acquisition for writing (Acquisition Write) and for reading (Acquisition Read).
Miss Rate	The number of unsuccessful lock attempts divided by Acquisitions, multiplied by 100.
Spin Count	The number of unsuccessful lock attempts. For complex lock and read/write lock there is a distinction between spin count for writing (Spin Count Write) and for reading (Spin Count Read).
Wait Count	The number of unsuccessful lock attempts that resulted in the attempting thread going to sleep to wait for the lock to become available. For complex lock and read/write lock there is a distinction between wait count for writing (Wait Count Write) and for reading (Wait Count Read).
Transform Count	The number of times that a simple lock has allocated a krlock, while the thread was trying to acquire the simple lock.
Busy Count	The number of simple_lock_try() calls that returned busy.
Percent Held of CPU	Total Time The percent of combined elapsed trace time that threads held the lock in question while dispatched. DISPATCHED_HOLDTIME_IN_SECONDS divided by trace duration, multiplied by 100.
Elaps(ed)	The percent of combined elapsed trace time that threads held the lock while dispatched or sleeping. UNDISPATCHED_AND_DISPATCHED_HOLDTIME_IN_SECONDS divided by trace duration, multiplied by 100.
Spin	The percent of combined elapsed trace time that threads spun while waiting to acquire this lock. SPIN_HOLDTIME_IN_SECONDS divided by combined trace duration, multiplied by 100.
Wait	The percentage of combined elapsed trace time that threads unsuccessfully tried to acquire this lock.
Return Address	The calling function's return address in hexadecimal.
Start Address	The start address of the calling function in hexadecimal.

Offset	The offset from the function start address in hexadecimal.
Thread Detail ^^^^^^^^^^^^^^	
The thread detail report consists of the following fields:	
ThreadID	Thread identifier.
Acquisitions	The number of successful lock attempts for this lock.
Miss Rate	The number of unsuccessful lock attempts divided by Acquisitions, multiplied by 100.
Spin Count	The number of unsuccessful lock attempts.
Wait Count	The number of unsuccessful lock attempts that resulted in the attempting thread going to sleep to wait for the lock to become available.
Transform Count	The number of times that a simple lock has allocated a krlock, while the thread was trying to acquire the simple lock.
Busy Count	The number of simple_lock_try() calls that returned busy.
Percent Held of CPU	Total Time The percent of combined elapsed trace time that threads held the lock in question while dispatched. DISPATCHED_HOLDTIME_IN_SECONDS divided by trace duration, multiplied by 100.
Elaps(ed)	The percent of combined elapsed trace time that threads held the lock while dispatched or sleeping. UNDISPATCHED_AND_DISPATCHED_HOLDTIME_IN_SECONDS divided by trace duration, multiplied by 100.
Spin	The percent of combined elapsed trace time that threads spun while waiting to acquire this lock. SPIN_HOLDTIME_IN_SECONDS divided by combined trace duration, multiplied by 100.
Wait	The percent of combined elapsed trace time that threads unsuccessfully tried to acquire this lock.
ProcessID	Process identifier (only for SIMPLE and COMPLEX Lock report).
Process Name	Name of the process (only for SIMPLE and COMPLEX Lock report).

Splat Sorting

splat allows the user to specify which criteria is used to sort the summary and lock detail reports using the **-s** option. The default sorting criteria is to sort by percent spin hold time, which is the ratio of time that threads spent spinning for a lock compared to the combined duration of the trace. Using **-s**, the sort criteria can be changed to the following:

Item	Description
a	Acquisitions; the number times a thread successfully acquired a lock.
c	Percent processor hold time; the ratio of processor hold time with the combined trace duration.
e	Percent Elapsed hold time; the ratio of elapsed hold time with the combined trace duration.
l	location; the address of the lock or function, or the ID of a thread.
m	Miss rate; the ratio missed lock attempts with the number of acquisitions.
s	Spin count; the number of unsuccessful lock attempts that result in a thread spinning waiting for the lock.
S	Percent processor spin hold time (default).

Item	Description
w	Percent elapsed wait time; the percent of the total time that a nonzero number of threads waited on the lock.
W	Average waitq depth; the average number of threads waiting on the lock, equivalent to the average time each waiting thread spends in this state.

splat will use the specified criteria to sort the lock reports in descending order.

Restrictions

Other types of locks, such as VMM, XMAP, and certain Java-specific locks are not analyzed.

Files

Item	Description
/etc/bin/splat	Simple Performance Lock Analysis Tool (splat). Provides kernel and pthread lock usage reports.

split Command

Purpose

Splits a file into pieces.

Syntax

To Split a File Into Multiple Files Containing a Specified Number of Lines

split [**-l** *LineCount*] [**-a** *SuffixLength*] [*File* [*Prefix*]]

To Split a File Into Multiple Files Containing a Specified Number of Bytes

split **-b** *Number* [**k** | **m**] [**-a** *SuffixLength*] [*File* [*Prefix*]]

Description

The **split** command reads the specified file and writes it in 1000-line pieces to a set of output files. The name of the first output file is constructed by combining the specified prefix (x by default) with the *aa* suffix, the second by combining the prefix with the *ab* suffix, and so on lexicographically through *zz* (a maximum of 676 files). The number of letters in the suffix, and consequently the number of output name files, can be increased by using the **-a** flag.

You cannot specify a *Prefix* longer than **PATH_MAX** - 2 bytes (or **PATH_MAX** - *SuffixLength* bytes if the **-a** flag is specified). The **PATH_MAX** variable specifies the maximum path-name length for the system as defined in the **/usr/include/sys/limits.h** file.

If you do not specify an input file or if you specify a file name of - (minus sign), the **split** command reads standard input.

The **split** command can be used with any regular text or binary files. After a file has been split, it can be restored to its original form by using the **cat** command, and the file fragments will be listed in the appropriate order.

Flags

Note: The **-b** and **-l** flags are mutually exclusive.

Item	Description
-a <i>SuffixLength</i>	Specifies the number of letters to use in forming the suffix portion of the output name files. The number of letters determines the number of possible output filename combinations. The default is two letters.
-b <i>Number</i>	Splits the file into the number of bytes specified by the <i>Number</i> variable. Adding the <i>k</i> (kilobyte) or <i>m</i> (megabyte) multipliers to the end of the <i>Number</i> value causes the file to be split into <i>Number</i> *1024 or <i>Number</i> *1,048,576 byte pieces, respectively.
-l <i>LineCount</i>	Specifies the number of lines in each output file. The default is 1000 lines.

Exit Status

This command returns the following exit values:

Ite	Description
m	
0	The command ran successfully.
>0	An error occurred.

Examples

1. To split a file into 1000-line segments, enter:

```
split book
```

This example splits book into 1000-line segments named xaa, xab, xac, and so forth.

2. To split a file into 50-line segments and specify the file-name prefix, enter:

```
split -l 50 book sect
```

This example splits book into 50-line segments named sectaa, sectab, sectac, and so forth.

3. To split a file into 2KB segments, enter:

```
split -b 2k book
```

This example splits the book into 2*1024-byte segments named xaa, xab, xac, and so forth.

4. To split a file into more than 676 segments, enter:

```
split -l 5 -a 3 book sect
```

This example splits a book into 5-line segments named sectaaa, sectaab, sectaac, and so forth, up to sectzzz (a maximum of 17,576 files).

Files

Item	Description
/usr/bin/split	Contains the split command.

splitlvcopy Command

Purpose

Splits copies from one logical volume and creates a new logical volume from them.

Syntax

splitlvcopy [-f] [-k] [-y *NewLogicalVolumeName*] [-Y *Prefix*] *LogicalVolume Copies* [*PhysicalVolume ...*]

> **splitlvcopy** [-f] [-k] [-y *NewLogicalVolumeName*] [-Y *Prefix*] [-p *mirror pool,...*] *LogicalVolume Copies* <

Description

Note:

1. To use this command, you must either have root user authority or be a member of the system group.
2. The **splitlvcopy** command is not allowed on a snapshot volume group or a volume group that has a snapshot volume group.
3. > The **splitlvcopy** command copies the passphrase authentication methods from the original encrypted logical volume to the new encrypted logical volume. <
4. > If the original logical volume has Platform keystore (PKS) method, the **splitlvcopy** command adds new PKS method in new encrypted logical volume if the PKS key slots are available. <
5. > The **splitlvcopy** command deletes the key server method in new encrypted logical volume. Use the -k flag to copy the key server method from the original encrypted logical volume. <
6. > The **splitlvcopy** command deletes the key file authentication method in new encrypted logical volume. <



Attention: Although the **splitlvcopy** command can split logical volumes that are open, including logical volumes containing mounted filesystems, this is not recommended. You may lose consistency between *LogicalVolume* and *NewLogicalVolume* if the logical volume is accessed by multiple processes simultaneously. When splitting an open logical volume, you implicitly accept the risk of potential data loss and data corruption associated with this action. To avoid the potential corruption window, close logical volumes before splitting and unmount filesystems before splitting.

The **splitlvcopy** command removes copies from each logical partition in *LogicalVolume* and uses them to create *NewLogicalVolume*. The *Copies* parameter determines the maximum number of physical partitions that remain in *LogicalVolume* after the split. Therefore, if *LogicalVolume* has 3 copies before the split, and the *Copies* parameter is 2, *LogicalVolume* will have 2 copies after the split and *NewLogicalVolume* will have 1 copy. You can not split a logical volume so that the total number of copies in *LogicalVolume* and *NewLogicalVolume* after the split is greater than the number of copies in *LogicalVolume* before the split.

The *NewLogicalVolume* will have all the same logical volume characteristics as *LogicalVolume*. If *LogicalVolume* does not have a logical volume control block the command will succeed with a warning message and creates *NewLogicalVolume* without a logical volume control block.

There are additional considerations to take when splitting a logical volume containing a filesystem. After the split there will be two logical volumes but there will only be one entry in the **/etc/filesystems** file which refers to *LogicalVolume*. To access *NewLogicalVolume* as a filesystem you must create an additional entry in **/etc/filesystems** with a different mount point which refers to *NewLogicalVolume*. If the mount point does not already exist, you have to create it before the new filesystem can be mounted. In addition, if *NewLogicalVolume* was created while *LogicalVolume* was open, you have to run the `fsck /dev/NewLogicalVolume` command before the new filesystem can be mounted.

You cannot use the System Management Interface Tool (SMIT) to run this command. Message catalogs are not supported for this command and therefore the error messages are provided in English only with no message catalog numbers.

Flags

Item	Description
-f	Specifies to split open logical volumes without requesting confirmation. By default, splitlvcopy requests confirmation before splitting an open logical volume. This includes open raw logical volumes and logical volumes containing mounted file systems.
> > -k	Copies the key server method from the original encrypted logical volume to the new logical volume. < <
> > -p <i>mirrorpool</i>	Specifies the mirror pool from which the logical volume copy must be split to create a new logical volume. < <
-y <i>NewLogicalVolumeName</i>	Specifies the name of the new logical volume to move copies from <i>LogicalVolume</i> .
-Y <i>Prefix</i>	Specifies the <i>Prefix</i> to use instead of the prefix in a system-generated name for the new logical volume. The prefix must be less than or equal to 13 characters. A name cannot begin with a prefix already defined in the PdDv class in the Device Configuration Database for other devices, nor be a name already used by another device.

Parameters

Item	Description
<i>Copies</i>	Specifies the maximum number of physical partitions that remain in logical volume after the split.
<i>LogicalVolume</i>	Specifies the logical volume name or logical volume ID to split.
<i>PhysicalVolume</i>	Specifies the physical volume name or the physical volume ID to remove copies from.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Security

Access Control: You must have root authority to run this command or be a member of the system group.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Auditing Events: N/A

Examples

- To split one copy of each logical partition belonging to logical volume named **oldlv** which currently has 3 copies of each logical partition, and create the logical volume **newlv**, enter the following command:

```
splitlvcopy -y newlv oldlv 2
```

Each logical partition in the logical volume **oldlv** now has two physical partitions. Each logical partition in the logical volume **newlv** now has one physical partition.

- **|<** To split the copy of the **lv00** logical volume in the **mp2** and the **mp3** mirror pools to create a logical volume, enter the following command:

```
splitlvcopy -p mp2,mp3 lv00 1
```

|<

Files

Item	Description
/usr/sbin/splitlvcopy	Contains the splitlvcopy command.
/tmp	Contains the temporary files created while the splitlvcopy command is running.

splitvg Command

Purpose

Splits a single mirror copy of a fully mirrored volume group.

Syntax

```
splitvg [-y SnapVGname] [-c Copy] |<-p mirrorpool|< [-k] [-f] [-i] VGname
```

Description

The **splitvg** command splits a single mirror copy of a fully mirrored volume group into a snapshot volume group. The original volume group **VGname** will stop using the disks that are now part of the snapshot volume group **SnapVGname**. Both volume groups will keep track of the writes within the volume group so that when the snapshot volume group is rejoined with the original volume group consistent data is maintained across the rejoined mirrors copies.

Note:

1. To split a volume group, all logical volumes in the volume group must have the target mirror copy and the mirror must exist on a disk or set of disks. Only the target mirror copy must exist on the target disk or disks.
2. The **splitvg** command will fail if any of the disks to be split are not active within the original volume group.
3. In the unlikely event of a system crash or loss of quorum while running this command, the **joinvg** command must be run to rejoin the disks back to the original volume group.
4. New logical volumes and file system mount points will be created in the snapshot volume group.
5. When the **splitvg** command targets a concurrent-capable volume group which is varied on in non-concurrent mode, the new volume group that is created will not be varied on when the **splitvg** command completes. The new volume group must be varied on manually.

6. You cannot add or remove an authentication method of an encrypted logical volume in primary or snapshot volume group.
7. The primary and the snapshot volume group can use the same authentication method.

Flags

Item

-c *Copy*

-f

-i

Description

Specifies the mirror copy that must be split. Valid values are 1, 2, or 3. The default is the second copy.

Force splits even if the mirror copy specified to create the snapshot volume group has stale partitions.

Splits the mirror copy of a volume group into a new volume group that cannot be rejoined into the original.

Note:

- Copies the passphrase authentication methods from the original encrypted logical volume to the new encrypted logical volume.
- If the original logical volume has PKS method, new PKS method are added in the new encrypted logical volume if PKS key slots are available.
- Deletes the key server method in new encrypted logical volume. Use the **-k** flag to copy the key server method from the original encrypted logical volume.
- Deletes the key file authentication method in the new encrypted logical volume.

>|>|-k

Copies the key server method from the original encrypted logical volume to the new logical volume in the new volume group.

Note: >| Both original and new encrypted logical volumes share the same key server authentication method. If you remove the key server authentication method from one logical volume, the other logical volume cannot access the key server authentication method as well. |<

|<<

>|>|-p *mirrorpool*

Specifies the mirror pool that must be split. The default is the second copy. |<|<

-y *SnapVGname*

Allows the volume group name to be specified rather than having the name generated automatically. Volume group names must be unique across the system and can range from 1 to 15 characters. The name cannot begin with a prefix already defined in the **PdDv** class in the Device Configuration database for other devices. The new volume group name is sent to standard output.

Security

Access Control: You must have root authority to run this command.

Examples

1. To split a volume group, enter the following command:

```
splitvg testvg
```

The second mirror copy of the volume group **testvg** is split into new volume group with an automatically generated name, which will be displayed.

2. To split first mirror copy of the volume group with the name **snapvg**, enter the following command:

```
splitvg -y snapvg -c 1 testvg
```

3. **>** To split the **mp1** mirror pool in the **vg00** volume group to a **snapvg** snapshot volume group, enter the following command:

```
splitvg -y snapvg -p mp1 vg00
```

I<

Files

Item	Description
/usr/sbin	Directory where the splitvg command resides.

splp Command

Purpose

Changes or displays printer driver settings.

Syntax

```
splp [ -b Option ] [ -B Number ] [ -c Option ] [ -C Option ] [ -e Option ] [ -f Option ] [ -F ] [ -i Number ]  
[ -l Number ] [ -n Option ] [ -N Option ] [ -p Option ] [ -P Option ] [ -r Option ] [ -s Number ] [ -S Option ]  
[ -t Option ] [ -T Number ] [ -w Number ] [ -W Option ] [ DevicePath ]
```

Description

The **splp** command changes or displays settings for a printer device driver. The default device path is **/dev/lp0**; all flags are optional. If the device path does not begin with a **/** (backslash) character, the **/dev** directory is assumed. Also, if no flags are specified, the **splp** command reports the current settings for the specified device path. To change the current settings, specify the appropriate flags. No other processing is done, and there is no other output.

The changes that the **splp** command makes remain in effect until the next time you restart the system or rerun the **splp** command. The **splp** command can be run from the **/etc/inittab** command file to configure your printer each time you start up the system.

Note: The **splp** command settings for the **-b**, **-c**, **-C**, **-f**, **-i**, **-l**, **-n**, **-p**, **-r**, **-t**, **-w**, and **-W** flags apply only when data is sent directly to the printer device (for example, redirecting the output of the **cat** command directly to the specifies device path). When files are queued for printing with the **enq**, **qprt**, **lp**, or **lpr** commands, the settings for these flags are ignored and are not changed.

Flags

Item	Description
-b <i>Option</i>	Specifies whether backspaces are sent to the printer: + Specifies backspaces be sent to the printer. ! Specifies backspaces be discarded.
-B <i>Number</i>	Sets the speed to the specified number of bits per second. Values for the <i>Number</i> variable are 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19,200, and 38,400.
-c <i>Option</i>	Specifies whether carriage returns are sent to the printer: + Sends carriage returns to the printer. ! Translates carriage returns to line feeds.
-C <i>Option</i>	Specifies whether all lowercase characters are converted to uppercase characters: + Converts lowercase characters to uppercase characters. ! Does not convert lowercase characters to uppercase characters.
-e <i>Option</i>	Specifies the processing to be performed when an error is detected: + Returns an error. ! Waits until error clears.
-f <i>Option</i>	Specifies whether the printer is sent form feeds or simulates a form feed with line feeds or carriage returns: + Sends form feeds to the printer. ! Simulates a form feed with line feeds or carriage returns.
-F!	Resets font status indicators for an 3812 Page Printer or an 3816 Page Printer. This flag causes fonts to be reloaded from the printer's font diskette into the printer's memory by the next spooled print job. This flag should be specified if the printer has been turned off and then turned back on, or if the fonts in the printer's memory have become corrupted.
-i <i>Number</i>	Indents the specified number of columns, where the value of the <i>Number</i> variable is an integer.
-l <i>Number</i>	Prints the specified number of lines per page, where the value of the <i>Number</i> variable is an integer.
-n <i>Option</i>	Specifies whether the printer is sent line feeds or translates line feeds to carriage returns: + Sends line feeds to the printer. ! Translates line feeds to carriage returns.

Item	Description
-N Option	Specifies whether parity generation and detection is enabled: + Enables parity generation and detection. ! Disables parity generation and detection.
-p Option	Specifies whether the system sends all characters to the printer unmodified or translates characters according to the settings for the -b , -c , -C , -f , -i , -l , -n , -r , -t , -w , and -W flags: + Sends all characters to the printer unmodified, overriding other settings. ! Translates characters according to the settings.
-P Option	Specifies the parity: + Specifies odd parity. ! Specifies even parity.
-r Option	Specifies whether carriage returns are added after line feeds: + Sends a carriage return after a line feed. ! Does not send a carriage return after a line feed.
-s Number	Selects character size where the <i>Number</i> variable is the number of bits. Values for the <i>Number</i> variable can be 5, 6, 7, or 8. See the termio.h special file for additional information on character size.
-S Option	Specifies the number of stop bits per character: + 2 stop bits per character. ! 1 stop bit per character.
-t Option	Specifies whether tabs are to be expanded: + Does not expand tabs. ! Expands tabs on 8 position boundaries.
-T Number	Sets the time-out period to the number of seconds specified by the <i>Number</i> variable. The value of the <i>Number</i> variable must be an integer.
-w Number	Prints the number of columns specified by the <i>Number</i> variable. The value of the <i>Number</i> variable must be an integer.

Item	Description
-W Option	Specifies whether to wrap characters beyond the specified width to the next line and print . . . (3 dots) after the new-line character:
+	Wraps characters beyond the specified width to the next line and prints . . . (3 dots) after the new-line character.
!	Truncates characters beyond the specified width.

Examples

1. To display the current printer settings for the **/dev/lp0** printer, enter:

```
splp
```

2. To change the printer settings, enter:

```
splp -w 80 -W + -C +
```

This changes the settings of the **/dev/lp0** printer for 80-column paper (the **-w 80** flag). It also wraps each line that is more than 80 columns wide onto a second line (the **-W+** flag), and prints all alphabetic characters in uppercase (the **-C+** flag).

Files

Item	Description
/dev/lp*	Contains the printer attribute file.
/etc/inittab	Contains the printer configuration command file.

spost Command

Purpose

Routes a message.

Syntax

```
spost [ -noalias | -alias File ... ] [ -format | -noformat ] [ -filter File | -nofilter ] [ -width Number ] [ -watch | -nowatch ] [ -remove | -noremove ] [ -backup | -nobackup ] [ -verbose | -noverbose ] File
```

Description

The **spost** command routes messages to the correct destinations. The **spost** command is not started by the user. The **spost** command is called by other programs only.

The **spost** command searches all components of a message that specify a recipient's address and parses each address to check for proper format. The **spost** command then puts addresses in the standard format and starts the **sendmail** command. The **spost** command performs a function similar to the **post** command, but it does less address formatting than the **post** command.

The **spost** command is the default (over the **post** command). Change the default by setting the **postproc** variable in your **.mh_profile**. For example:

```
postproc: /usr/lib/mh/post
```

The *File* parameter is the name of the file to be posted.

Flags

Item	Description
-alias <i>File</i>	Searches the specified mail alias file for addresses. You can repeat this flag to specify multiple mail alias files. The spost command automatically searches the /etc/mh/MailAliases file.
-backup	Renames the message file by placing a , (comma) before the file name after the spost command successfully posts the message.
-filter <i>File</i>	Uses the header components in the specified file to copy messages sent to the Bcc : field recipients.
-format	Puts all recipient addresses in a standard format for the delivery transport system. This flag is the default.
-help	Lists the command syntax, available switches (toggles), and version information. Note: For Message Handler (MH), the name of this flag must be fully spelled out.
-noalias	Does not use any alias files for delivering the message.
-nobackup	Does not rename the message after posting the file. This flag is the default.
-nofilter	Strips the Bcc : field header from the message and sends it to recipients specified in the Bcc : component. This flag is the default.
-noformat	Does not alter the format of the recipient addresses.
-noremove	Does not remove the temporary message file after posting the message.
-noverbose	Does not display information during the delivery of the message to the sendmail command. This flag is the default.
-nowatch	Does not display information during delivery by the sendmail command. This flag is the default.
-remove	Removes the temporary message file after the message has been successfully posted. This flag is the default.
-verbose	Displays information during the delivery of the message to the sendmail command. This information allows you to monitor the steps involved.
-watch	Displays information during the delivery of the message by the sendmail command. This information allows you to monitor the steps involved.
-width <i>Number</i>	Sets the width of components that contain addresses. The default is 72 columns.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Files

Item	Description
\$HOME/.mh_profile	Contains the Message Handler (MH) user profile.
/tmp/pst <i>Number</i>	Contains the temporary message file.

Item	Description
<code>/etc/mh/MailAliases</code>	Contains the default mail aliases.
<code>/usr/lib/mh/.mh_profile</code>	Contains the Message Handler (MH) user profile.

spray Command

Purpose

Sends a specified number of packets to a host and reports performance statistics.

Syntax

`/usr/sbin/spray Host [ -c Count ] [ -d Delay ] [ -i ] [ -l Length ]`

Description

The **spray** command uses the Remote Procedure Call (RPC) protocol to send a one-way stream of packets to the host you specify. This command reports how many packets were received and at what transfer rate. The *Host* parameter can be either a name or an Internet address. The host only responds if the **sprayed** daemon is running.

Note: The **spray** command does not support IPv6.

See the **rpc.sprayed** daemon documentation for factors that affect **spray** command performance.

Flags

Item	Description
<code>-c Count</code>	Specifies the number of packets to send. The default value is the number of packets required to make the total stream size 100,000 bytes.
<code>-d Delay</code>	Specifies the time, in microseconds, the system pauses between sending each packet. The default is 0.
<code>-i</code>	Uses the Internet Control Message Protocol (ICMP) echo packets rather than the RPC protocol. Since ICMP echoes automatically, it creates a two-way stream. You must be root user to use this option.
<code>-l Length</code>	Specifies the number of bytes in the packet that holds the RPC call message. The default value of the <i>Length</i> parameter is 86 bytes, the size of the RPC and UDP headers. The data in the packet is encoded using eXternal Data Representation (XDR). Since XDR deals only with 32-bit quantities, the spray command rounds smaller values up to the nearest possible value. When the <i>Length</i> parameter is greater than 1500 for Ethernet or 1568 for token-ring, the RPC call can no longer fit into one Ethernet packet. Therefore, the <i>Length</i> field no longer has a simple correspondence to Ethernet packet size.

Examples

1. When sending a **spray** command to a workstation, specify the number of packets to send and the length of time the system will wait between sending each packet as follows:

```
/usr/sbin/spray zorro -c 1200 -d 2
```

In this example, the **spray** command sends 1200 packets at intervals of 2 microseconds to the workstation named zorro.

2. To change the number of bytes in the packets you send, enter:

```
/usr/sbin/spray zorro -l 1350
```

In this example, the **spray** command sends 1350-byte packets to the workstation named **zorro**.

3. To send echo packets using the ICMP protocol instead of the RPC protocol, enter:

```
/usr/sbin/spray zorro -i
```

In this example, the **spray** command sends echo packets to the workstation named **zorro**.

sprayd Daemon

Purpose

Receives packets sent by the **spray** command.

Syntax

```
/usr/lib/netsvc/spray/rpc.sprayd
```

Description

The **rpc.sprayd** daemon is a server that records the packets sent by the **spray** command. The **rpc.sprayd** daemon is normally started by the **inetd** daemon.

UDP Performance

User Datagram Protocol (UDP) performance with the **spray** command and the **rpc.sprayd** daemon can be affected by the following factors:

- How memory buffers (mbufs) are tuned for system configuration.
- The incoming burst rate (that is, interframe gap) of UDP packets for the **spray** command.
- Other system activity. Since the **rpc.sprayd** daemon runs as a normal user process, other activity (such as the **init** process, or the **syncd** daemon) can affect the operation of the **rpc.sprayd** daemon.
- Priority of the **rpc.sprayd** daemon process. The **rpc.sprayd** daemon has a floating process priority that is calculated dynamically.
- The size of the receive socket buffer used by the **rpc.sprayd** daemon. Because various implementations use different socket buffer sizes, measuring UDP performance with the **spray** command and the **rpc.sprayd** daemon is difficult and inconclusive.

Files

Item	Description
<code>/etc/inetd.conf</code>	TCP/IP configuration file that starts RPC daemons and other TCP/IP daemons.

srcmstr Daemon

Purpose

Starts the System Resource Controller.

Syntax

```
srcmstr /usr/sbin/srcmstr [ -r ] [ -B ]
```

Description

The **srcmstr** daemon is the System Resource Controller (SRC). The **srcmstr** daemon creates and controls subsystems, handles short subsystem status requests, passes requests on to a subsystem, and handles error notification.

The **srcmstr** daemon is normally started by using an **inittab** file entry.

Flags

Item	Description
-r	Accepts remote requests if the daemon is started with the -r flag. If you start srcmstr without the -r flag, remote requests are ignored.
-B	Specifies the -B flag that causes the srcmstr daemon to run as in previous releases (AIX 4.3.1 and earlier).
Note:	
1. The srcmstr daemon is typically started from inittab . To add the -r or -B flags, edit /etc/inittab and run init q or reboot.	
2. The user must be running as root on the remote system. The local /etc/hosts.equiv file or the /.rhosts file must be configured to allow remote requests.	

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Auditing Events: If the auditing subsystem has been properly configured and is enabled, the **srcmstr** command will generate the following audit record (event) every time the command is executed:

Event	Information
SRC_Start	Lists in an audit log the name of the subsystems being started.
SRC_Stop	Lists in an audit log the name of the subsystems being stopped.

See **Setting Up Auditing** in Security for more details about how to properly select and group audit events, and how to configure audit event data collection.

Error Recovery

The default **/etc/inittab** specifies the **respawn** flag for the **srcmstr** daemon. If the **srcmstr** daemon terminates abnormally and the **/etc/inittab** specifies the **respawn** flag, the **srcmstr** daemon is restarted. It then determines which SRC subsystems were active during the previous invocation. The daemon re-establishes communication with these subsystems (if it existed previously), and initializes a private kernel extension and the **srcd** daemon to monitor the subsystem processes.

If a subsystem known to the previous invocation of **srcmstr** terminates, the SRC kernel extension notifies the **srcd** daemon. The **srcd** daemon sends a socket message to **srcmstr** and subsystem termination is handled as if the subsystem had been started by the current **srcmstr**. This function can be disabled by specifying the **-B** flag when the **srcmstr** daemon is started. The SRC kernel extension is in **/usr/lib/drivers/SRC_kex.ext**. The executable for **srcd** is **/usr/sbin/srcd**.

Files

Item	Description
/etc/inittab	Specifies stanzas read by the init command.
/etc/objrepos/SRCsubsys	Specifies the SRC Subsystem Configuration Object Class.
/etc/objrepos/SRCnotify	Specifies the SRC Notify Method Object Class.
/etc/hosts.equiv	Specifies that no remote requests will work if the specified host name is not in the /etc/hosts.equiv file.
/etc/services	Defines the sockets and protocols used for Internet services.
/dev/SRC	Specifies the AF_UNIX socket file.
/dev/.SRC-unix	Specifies the location for temporary socket files.
/dev/.SRC-unix/SRCD	Specifies the AF_UNIX socket file for the srcd daemon.
/var/adm/SRC/active_list	Contains a list of active subsystems. Caution: The structure of this file is internal to SRC and is subject to change.
/var/adm/SRC/watch_list	Contains a list of subsystem processes active during the previous invocation of the srcmstr daemon. Caution: The structure of this file is internal to SRC and is subject to change.
/.rhosts	Specifies remote machines and users (root only) that are allowed to request SRC function from this machine.

start-secldapclntd Command

Purpose

The **start-secldapclntd** script is used to start the **secldapclntd** LDAP client daemon.

Syntax

```
/usr/sbin/start-secldapclntd [ -C CacheSize ] [ -p NumOfThread ] [ -t CacheTimeOut ] [ -T HeartBeatIntv ]  
[ -o ldapTimeOut ]
```

Description

The **start-secldapclntd** script starts the **secldapclntd** daemon if it is not running. It does not do anything if the **secldapclntd** daemon is already running. The script also cleans the portmapper registration (if there is any) from previous **secldapclntd** daemon process before it starts the **secldapclntd** daemon. This prevents the startup failure of the new daemon process from portmap-per registration failure.

Flags

By default, the **secldapclntd** daemon reads the configuration information specified in the **/etc/security/ldap/ldap.cfg** file at startup. If the following options are given in command line when starting **secldapclntd** process, the options from the command line will overwrite the values in the **/etc/security/ldap/ldap.cfg** file.

Item	Description
-C <i>CacheSize</i>	Sets the maximum cache entries used by the secdapclntd daemon to CacheSize number of entries. The valid range is 100-65536 entries for user cache entry. The default value is 1000. The valid range is 10-65536 for group cache entry. The default value is 100. If you set the user cache entry in the start-secdapclntd command by using the -C option, the group cache entry is set to 10% of the user cache entry.
-o <i>ldapTimeOut</i>	Timeout period in seconds for LDAP client requests to the server. This value determines how long the client will wait for a response from the LDAP server. Valid range is 0 - 3600 (1 hour). Default is 60 seconds. Set this value to 0 to disable the timeout and force the client to wait indefinitely.
-p <i>NumOfThread</i>	Sets the number of thread used by the secdapclntd daemon to NumOfThread threads. Valid range is 1-256. The default is 10.
-t <i>CacheTimeout</i>	Sets the cache to expire in CacheTimeout seconds. Valid range is 60- 3600 seconds. The default is 300 seconds.
-T <i>HeartBeatIntv</i>	Sets the time interval of heartbeat between this client and the LDAP server. Valid values are 60-3,600 seconds. Default is 300.

Security

A user with the **aix.security.ldap** authorization is authorized to use this command.

Examples

1. To start the **secdapclntd** daemon, type:

```
/usr/sbin/start-secdapclntd
```

2. To start the **secdapclntd** with using 20 threads and cache timeout value of 600 seconds, type:

```
/usr/sbin/start-secdapclntd -p 20 -t 600
```

It is recommended that you specify these values in the **/etc/security/ldap/ldap.cfg** file, so that these values will be used each time you start the **secdapclntd** process.

Files

Item	Description
/usr/sbin/start-secdapclntd	Used to start the secdapclntd LDAP client daemon.

startcondresp Command

Purpose

Starts monitoring a condition that has one or more linked responses.

Syntax

To start monitoring a condition:

```
startcondresp [-h] [-TV] condition[:node_name] [response [response...]]
```

To unlock or lock the condition/response association:

```
startcondresp {-U | -L} [-h] [-TV] condition[:node_name] response
```

Description

The `startcondresp` command starts the monitoring of a condition that has a linked response. A link between a condition and a response is called a *condition/response association*. In a cluster environment, the condition and the response must be defined on the same node. After monitoring is started, when the condition occurs, the response is run. If no responses are specified, monitoring is started for all responses linked to the condition. This causes all of the linked responses to run when the condition occurs. If more than one response is specified, monitoring is started only for those linked responses.

If one or more responses are specified and the responses are not linked with the condition, the `startcondresp` command links the specified responses to the condition, and monitoring is started. Use the `mkcondresp` command to link a response to a condition without starting monitoring.

If a particular condition/response association is needed for system software to work properly, it may be locked. A locked condition/response association cannot be started by the `startcondresp` command. If the condition/response association you specify on the `startcondresp` command is locked, it will not be started; instead an error will be generated informing you that this condition/response association is locked. To unlock a condition/response association, you can use the `-U` flag. However, because a condition/response association is typically locked because it is essential for system software to work properly, you should exercise caution before unlocking it. To lock a condition/response association so it cannot be started, stopped, or removed, reissue this command using its `-L` flag.

Flags

-h

Writes the command's usage statement to standard output.

-T

Writes the command's trace messages to standard error. For your software service organization's use only.

-V

Writes the command's verbose messages to standard output.

-U

Unlocks a condition/response association so it can be started, stopped, or removed. If a condition/response association is locked, this is typically because it is essential for system software to work properly. For this reason, you should exercise caution before unlocking it. When unlocking a condition/response association using the `-U` flag, no other operation can be performed by this command.

-L

Locks a condition/response association so it cannot be started, stopped, or removed. When locking a condition/response association using the `-L` flag, no other operation can be performed by this command.

Parameters

condition

Specifies the name of the condition linked to the response. The condition is always specified first.

node_name

Specifies the node in the domain where the condition is defined. If *node_name* is not specified, the local node is used. *node_name* is a node within the scope determined by the `CT_MANAGEMENT_SCOPE` environment variable.

response

Specifies the name of one or more responses. Specifying more than one response links the responses to the condition if they are not already linked and starts monitoring for the specified responses.

Security

The user needs write permission for the IBM.Association resource class to run `startcondresp`. Permissions are specified in the access control list (ACL) file on the contacted system. See the *RSCT: Administration Guide* for details on the ACL file and how to modify it.

Exit Status

- 0**
The command ran successfully.
- 1**
An error occurred with RMC.
- 2**
An error occurred with a command-line interface script.
- 3**
An incorrect flag was entered on the command line.
- 4**
An incorrect parameter was entered on the command line.
- 5**
An error occurred that was based on incorrect command-line input.

Environment Variables

CT_CONTACT

Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When `CT_CONTACT` is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If `CT_CONTACT` is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

CT_IP_AUTHENT

When the `CT_IP_AUTHENT` environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the `CT_CONTACT` environment variable is set. `CT_IP_AUTHENT` only has meaning if `CT_CONTACT` is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon in processing the resources of the event-response resource manager (ERRM). The management scope determines the set of possible target nodes where the resources can be processed. The valid values are:

- 0**
Specifies *local* scope.
 - 1**
Specifies *local* scope.
 - 2**
Specifies *peer domain* scope.
 - 3**
Specifies *management domain* scope.
- If this environment variable is *not* set, *local* scope is used.

Implementation Specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) fileset for AIX.

Standard Output

When the `-h` flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

These examples apply to standalone systems:

1. To start monitoring for the condition "FileSystem space used " by using the response "Broadcast event on-shift", whether or not the response is linked with the condition, run this command:

```
startcondresp "FileSystem space used" "Broadcast event on-  
shift"
```

2. To start monitoring for the condition "FileSystem space used " by using all of its linked responses, run this command:

```
startcondresp "FileSystem space used"
```

3. To start monitoring for the condition "FileSystem space used " by using the response "Broadcast event on-shift" and "E-mail root anytime", whether or not they are linked with the condition, run this command:

```
startcondresp "FileSystem space used" "Broadcast event on-shift" "E-mail root anytime"
```

These examples apply to management domains:

1. To start monitoring for the condition "FileSystem space used" on the management server using the response "Broadcast event on-shift", whether or not the response is linked with the condition, run this command on the management server:

```
startcondresp "FileSystem space used" "Broadcast event on-  
shift"
```

2. To start monitoring for the condition "FileSystem space used" on the managed node nodeB using the response "Broadcast event on-shift", whether or not the response is linked with the condition, run this command on the management server:

```
startcondresp "FileSystem space used":nodeB "Broadcast event on-  
shift"
```

This example applies to peer domains:

1. To start monitoring for the condition "FileSystem space used" on nodeA in the domain using the response "Broadcast event on-shift" (also on nodeA in the domain), whether or not the response is linked with the condition, run this command on any node in the domain:

```
startcondresp "FileSystem space used":nodeA "Broadcast event on-shift"
```

Location

/opt/rsct/bin/startcondresp

startripdomain Command

Purpose

Brings a peer domain that has already been defined online.

Syntax

```
startripdomain [ -A | -L ] [-t timeout] [ -Q quorum_type | quorum_type_name ] [-m fanout] [-h] [-w  
[-s Seconds]] [-TV] peer_domain
```

Description

The `startripdomain` command brings a defined peer domain online by starting the resources on each node belonging to the peer domain.

The `startripdomain` command must be run on a node that is defined to the peer domain. The command invites all offline nodes defined to the peer domain to come online in the peer domain every time the command is run for the peer domain. The command can be run more than once in the peer domain. If all the nodes defined in the peer domain are already online, no action is performed.

The `startripdomain` command determines the peer domain configuration to use to bring the peer domain online by examining the peer domain configuration on the nodes defined to the peer domain. The latest version of the peer domain configuration information that is found is used to bring the peer domain online. By default, the latest version of the peer domain configuration found on at least half of the nodes is used. Specifying the `-A` flag causes the latest version of the peer domain configuration found on all of the nodes defined in the peer domain to be used. Specifying the `-L` flag causes the configuration on the local node to be used.

In determining the latest version of the peer domain configuration information, a configuration timeout defines when to stop checking versions and begin to bring the peer domain online. The default timeout value is 120 seconds. The timeout value can be changed using the `-t` flag. The timeout value should be at least long enough so that the latest version of the peer domain configuration information from at least half of the nodes can be found.

A node can only be online to one peer domain at a time. The `startripdomain` command cannot be run on a node for a peer domain when another peer domain is already online for that node.

Flags

-A

Finds and uses the latest version of the peer domain configuration information from all of the nodes in the peer domain. This flag cannot be specified if the `-L` flag is specified. If neither flag (`-A` or `-L`) is specified, the latest version of the peer domain configuration information from at least half of the nodes in the peer domain is used.

-L

Uses the latest version of the peer domain configuration information that is on the local node. This flag cannot be specified if the `-A` flag is specified. If neither flag (`-A` or `-L`) is specified, the latest version of the peer domain configuration information from at least half of the nodes in the peer domain is used.

-t *timeout*

Specifies the timeout value in seconds. This flag limits the amount of time used to find the latest version of the peer domain configuration. When the timeout value is exceeded, the latest version of the peer domain configuration information found thus far is used. The timeout value should be long enough so that the latest version of the peer domain configuration information from at least half of the nodes can be found. The default timeout value is 120 seconds.

-Q *quorum_type* | *quorum_type_name*

Enables you to override the startup quorum mode. This can be specified as an integer quorum type or quorum type name. If you do not specify this flag, startup quorum mode will be specified using the `mkrpdomain` command's `-Q` flag (or the default quorum mode for your environment) when you created the peer domain. You can override the quorum startup mode only if the quorum mode has been defined as `normal` or `quick`. The valid values are:

0 | *normal*

Specifies normal start-up quorum rules. Half of the nodes will be contacted for configuration information.

1 | *quick*

Specifies quick start-up quorum rules. One node will be contacted for configuration information.

-m *fanout*

Specifies the maximum number of threads to use for this start operation. The `-m` flag overrides the default *fanout* value for the specified peer domain. This value is stored as a persistent attribute in the peer domain's `IBM.PeerNode` class. *fanout* can be an integer from 16 to 2048.

-h

Writes the command's usage statement to standard output.

-s

Specifies the wait time in seconds for the peer domain to be online before the command completes when the `-s` flag is used with the `-w` flag. If the waiting time exceeds the number of seconds, the command returns, but the online operation continues. The default value is 300 seconds (5 minutes). Use 0 to specify that the command must not return until the peer domain is online (no timeout on waiting).

-T

Writes the command's trace messages to standard error. For your software service organization's use only.

-V

Writes the command's verbose messages to standard output.

-w

Waits for the peer domain to be online before the command completes. Use the `-s` flag to specify the waiting time in seconds.

Parameters

peer_domain

Specifies the name of a previously-defined peer domain that is to be brought online.

Security

The user of the `starttrpdomain` command needs write permission for the `IBM.PeerDomain` resource class on each node that is defined to the peer domain. By default, `root` on any node in the peer domain has read and write access to this resource class through the configuration resource manager.

Exit Status

0

The command ran successfully.

1

An error occurred with RMC.

2

An error occurred with a command-line interface script.

3

An incorrect flag was entered on the command line.

4

An incorrect parameter was entered on the command line.

5

An error occurred that was based on incorrect command-line input.

6

The peer domain definition does not exist.

Environment Variables

CT_CONTACT

Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When CT_CONTACT is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If CT_CONTACT is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. CT_IP_AUTHENT only has meaning if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

Restrictions

This command must be run from a node that is defined to the peer domain.

Implementation Specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) fileset for AIX.

Standard Input

When the -F " - " flag is specified, this command reads one or more node names from standard input.

Standard Output

When the -h flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

In these examples, nodeA is one of the nodes defined to ApplDomain.

1. To bring ApplDomain online, run this command on nodeA:

```
startlppdomain ApplDomain
```

2. To bring ApplDomain online using all of the nodes in the peer domain to obtain the latest version of the peer domain configuration information, run this command on nodeA:

```
startlppdomain -A ApplDomain
```

3. To bring ApplDomain online using a peer domain configuration timeout value of 240 seconds (to make sure that at least half of the nodes in the peer domain are used), run this command on nodeA:

```
starttrpdomain -t 240 ApplDomain
```

Location

/opt/rsct/bin/starttrpdomain

starttrpnode Command

Purpose

Brings one or more nodes online to a peer domain.

Syntax

```
starttrpnode [-h] [-w [-s Seconds]] [-TV] node_name1 [node_name2 ...]
```

```
starttrpnode -f | -F {file_name | "-" } [-h] [-w [-s Seconds]] [-TV]
```

Description

The `starttrpnode` command brings one or more offline nodes online to a peer domain. The peer domain is determined by the online peer domain where the command is run. The command must be run from a node that is online to the desired peer domain.

The node that is being brought online must have already been defined to be in this peer domain using the `addrpnode` command or the `mktrpdomain` command. The node must not be online to any other peer domain.

Flags

-f | -F {file_name | "-" }

Reads a list of node names from *file_name*. Each line of the file is scanned for one node name. The pound sign (#) indicates that the remainder of the line (or the entire line if the # is in column 1) is a comment.

Use `-f "-"` or `-F "-"` to specify STDIN as the input file.

-h

Writes the command's usage statement to standard output.

-s

Specifies the wait time in seconds for all of the specified nodes to be online before the command completes when the `-s` flag is used with the `-w` flag. If the waiting time exceeds the number of seconds, the command returns, but the online operation continues. The default value is 300 seconds (5 minutes). Use 0 to specify that the command must not return until all of the specified nodes are online (no timeout on waiting).

-T

Writes the command's trace messages to standard error. For your software service organization's use only.

-V

Writes the command's verbose messages to standard output.

-w

Waits for all of the specified nodes to be online before the command completes. Use the `-s` flag to specify the waiting time in seconds.

Parameters

***node_name1* [*node_name2* ...]**

Specifies the peer domain node names of the nodes to be brought online to the peer domain. You can bring one or more nodes online using the `startxnode` command. You must specify the node names in exactly the same format as they were specified with the `addxnode` command or the `mkxpdomain` command. To list the peer domain node names, run the `lsxnode` command.

Security

The user of the `startxnode` command needs write permission for the `IBM.PeerNode` resource class on each node that is to be started in the peer domain. By default, `root` on any node in the peer domain has read and write access to this resource class through the configuration resource manager.

Exit Status

0

The command ran successfully.

1

An error occurred with RMC.

2

An error occurred with a command-line interface script.

3

An incorrect flag was entered on the command line.

4

An incorrect parameter was entered on the command line.

5

An error occurred that was based on incorrect command-line input.

Environment Variables

CT_CONTACT

Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When `CT_CONTACT` is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If `CT_CONTACT` is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

CT_IP_AUTHENT

When the `CT_IP_AUTHENT` environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the `CT_CONTACT` environment variable is set. `CT_IP_AUTHENT` only has meaning if `CT_CONTACT` is set to an IP address; it does not rely on the domain name system (DNS) service.

Restrictions

This command must be run from a node that is online to the peer domain. The node that is to be brought online must be offline to the peer domain, must not be online to any other peer domain, and must be reachable from where the command is run.

Implementation Specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) fileset for AIX.

Standard Input

When the `-f "-"` or `-F "-"` flag is specified, this command reads one or more node names from standard input.

Standard Output

When the `-h` flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

In this example, `nodeA` is defined and online to `App1Domain`, `nodeB` is reachable from `nodeA`, and `nodeB` is not online to `App1Domain` or any other peer domain. To bring `nodeB` online to `App1Domain`, run this command from `nodeA`:

```
starttrnode nodeB
```

Location

`/opt/rsct/bin/starttrnode`

startsrc Command

Purpose

Starts a defined resource (that is, brings it online).

Syntax

To start one or more resources, using data entered on the command line:

```
startsrc -s "selection_string" [-N { node_file | "-" }] [-n node_name] [-h] [-TV] resource_class  
[arg=value...]
```

```
startsrc -r [-n node_name] [-h] [-TV] resource_handle [arg=value...]
```

To start one or more resources using command arguments that are predefined in an input file:

```
startsrc -f resource_data_input_file -s "selection_string" [-N { node_file | "-" }] [-n node_name]  
[-h] [-TV] resource_class
```

```
startsrc -f resource_data_input_file -r [-n node_name] [-h] [-TV] resource_handle
```

To list the names and data types of the command arguments:

```
startsrc -l [-h] resource_class
```

Description

The `startsrc` command requests that the resource monitoring and control (RMC) subsystem bring one or more resources online. The request is performed by the appropriate resource manager.

To start one or more resources, use the `-s` flag to bring online all of the resources that match the specified selection string.

Instead of specifying multiple node names in *selection_string*, you can use the `-N node_file` flag to indicate that the node names are in a file. Use `-N "-"` to read the node names from standard input.

To start one specific resource, use the `-r` flag to specify the resource handle that represents that specific resource.

Use the `-l` flag to determine whether the specified resource class accepts any additional command arguments.

If Cluster Systems Management (CSM) is installed on your system, you can use CSM defined node groups as node name values to refer to more than one node. For information about working with CSM node groups and using the `CSM nodegrp` command, see the *CSM: Administration Guide* and the *CSM: Command and Technical Reference*.

The successful completion of this command does not guarantee that the resource is online, only that the resource manager successfully received the request to bring this resource online. Monitor the dynamic attribute `OpState` of the resource to determine when the resource is brought online. Register an event for the resource, specifying the `OpState` attribute, to know when the resource is actually online. Or, intermittently run the `lsrsrc` command until you see that the resource is online (the value of `OpState` is 1). For example:

```
lsrsrc -s 'Name == "/filesystem1"' -t IBM.FileSystem Name OpState
```

Parameters

resource_class

Specifies the name of the resource class that contains the resources that you want to bring online.

resource_handle

Specifies the resource handle that corresponds to the resource you want to bring online. Use the `lsrsrc` command to obtain a list of valid resource handles. The resource handle must be enclosed within double quotation marks, for example:

```
"0x4017 0x0001 0x00000000 0x0069684c 0x0d4715b0 0xe9635f69"
```

arg=value...

Specifies one or more pairs of command argument names and values.

arg

Specifies the argument name.

value

Specifies the value for this argument. The value data type must match the definition of the argument data type.

Command arguments are optional. If any `arg=value` pairs are entered, there must be one `arg=value` pair for each command argument defined for the online function for the specified resource class.

Use `startlsrsrc -l` to get a list of the command argument names and data types for the specific resource class.

Flags

-f resource_data_input_file

Specifies the name of the file that contains resource argument information. The contents of the file would look like this:

```
PersistentResourceArguments::  
argument1 = value1  
argument2 = value2
```

-l

Lists the command arguments and data types. Some resource managers accept additional arguments that are passed to the online request. Use this flag to list any defined command arguments and the data types of the command argument values.

-n *node_name*

Specifies the name of the node where the resource is to be brought online. *node_name* is a `NodeNameList` attribute value. Use this flag to bring a floating resource online on a different node if the node where it was online might be down.

Do *not* specify this flag if you want the resource to be brought online on the node where it is known.

-N { *node_file* | "-" }

Specifies that node names are read from a file or from standard input. Use `-N node_file` to indicate that the node names are in a file.

- There is one node name per line in *node_file*
- A number sign (#) in column 1 indicates that the line is a comment
- Any blank characters to the left of a node name are ignored
- Any characters to the right of a node name are ignored

Use `-N "-"` to read the node names from standard input.

The `CT_MANAGEMENT_SCOPE` environment variable determines the scope of the cluster. If `CT_MANAGEMENT_SCOPE` is not set, management domain scope is chosen first (if a management domain exists), peer domain scope is chosen next (if a peer domain exists), and then local scope is chosen, until the scope is valid for the command. The command runs once for the first valid scope it finds. For example, if a management domain and a peer domain both exist and `CT_MANAGEMENT_SCOPE` is not set, this command applies to the management domain. If you want this command to apply to the peer domain, set `CT_MANAGEMENT_SCOPE` to 2.

-s "*selection_string*"

Specifies the selection string. All selection strings must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
-s 'Name ?= "test"'
```

Only persistent attributes can be listed in a selection string.

-h

Writes the command usage statement to standard output.

-T

Writes the command trace messages to standard error. For your software service organization use only.

-V

Writes the command verbose messages (if there are any available) to standard output.

Environment variables**CT_CONTACT**

When the `CT_CONTACT` environment variable is set to a host name or IP address, the command contacts the resource monitoring and control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are on the system to which the connection is established.

CT_IP_AUTHENT

When the `CT_IP_AUTHENT` environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the `CT_CONTACT` environment variable is set. `CT_IP_AUTHENT` only has meaning if `CT_CONTACT` is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon to monitor and control the resources and resource classes. The management scope determines the set of possible target nodes where the resources and resource classes can be monitored and controlled. The valid values are:

- 0** Specifies *local* scope.
- 1** Specifies *local* scope.
- 2** Specifies *peer domain* scope.
- 3** Specifies *management domain* scope.

If this environment variable is *not* set, *local* scope is used.

Standard output

When the `-h` flag is specified, this command usage statement is written to standard output. When the `-V` flag is specified, this command verbose messages (if there are any available) are written to standard output.

Standard error

All trace messages are written to standard error.

Exit status

- 0** The command ran successfully.
- 1** An error occurred with RMC.
- 2** An error occurred with the command-line interface (CLI) script.
- 3** An incorrect flag was specified on the command line.
- 4** An incorrect parameter was specified on the command line.
- 5** An error occurred with RMC that was based on incorrect command-line input.
- 6** No resources were found that match the specified selection string.

Security

You need write permission for the *resource_class* specified in `startsrc` to run `startsrc`. Permissions are specified in the access control list (ACL) file on the contacted system. See the *Administering RSCT* guide for information about the ACL file and how to modify it.

Implementation specifics

This command is part of the `rsct.core.rmc` fileset for AIX operating system and `rsct.core-3.1.0.0-0.platform.rpm` package for Linux, Solaris, and Windows operating systems, where *platform* is `i386`, `ppc`, `ppc64`, `s390`, or `x86_64`.

Location

/opt/rsct/bin/startsrc

Examples

Suppose that you have a peer domain called `foo` with three defined nodes: `nodeA`, `nodeB`, and `nodeC`. `nodeA` has two Ethernet cards: `ent0` and `ent1`.

1. Suppose `nodeA` is online and `ent0` (on `nodeA`) is offline. To bring `ent0` online on `nodeA`, run this command on `nodeA`:

```
startsrc -s 'Name == "ent0"' IBM.EthernetDevice
```

2. Suppose `nodeA` and `nodeB` are online, `ent0` (on `nodeA`) is offline, and you are currently logged on to `nodeB`. To bring `ent0` online on `nodeA`, run this command on `nodeB`:

```
startsrc -s 'Name == "ent0"' -n nodeA IBM.EthernetDevice
```

3. Suppose file system `/filesys1` is defined, but not mounted on `nodeB`. To bring `/filesys1` online on `nodeB`, run this command on `nodeA`:

```
startsrc -s 'Name == "/filesys1"' -n nodeB IBM.FileSystem
```

4. Suppose the resource handle for `ent0` on `nodeA` is:

```
0x406b 0x0001 0x00000000 0x0069564c 0x0dc1f272 0xb9de145e
```

To bring `ent0` online on `nodeA`, run this command on `nodeA`:

```
startsrc -r "0x406b 0x0001 0x00000000 0x0069564c 0x0dc1f272 0xb9de145e"
```

startsrc Command

Purpose

Starts a subsystem, a group of subsystems, or a subserver.

Syntax

To Start a Subsystem

```
startsrc [ -a Argument] [ -e Environment] [ -h Host] { -s Subsystem | -g Group}
```

To Start a Subserver

```
startsrc [ -h Host] -t Type [ -o Object] [ -p SubsystemPID]
```

Description

The **startsrc** command sends the System Resource Controller (SRC) a request to start a subsystem or a group of subsystems, or to pass on a packet to the subsystem that starts a subserver.

If a start subserver request is passed to the SRC and the subsystem to which the subserver belongs is not currently active, the SRC starts the subsystem and transmits the start subserver request to the subsystem.

Flags

Item	Description
-a <i>Argument</i>	Specifies an argument string that is passed to the subsystem when the subsystem is executed. This string is passed from the command line and appended to the command line arguments from the subsystem object class. The <i>Argument</i> string specified is a maximum of 1200 characters or the command is unsuccessful. The command argument is passed by the SRC to the subsystem, according to the same rules used by the shell. Quoted strings are passed as a single argument, and blanks outside a quoted string delimit an argument. Single and double quotes can be used.
-e <i>Environment</i>	Specifies an environment string that is placed in the subsystem environment when the subsystem is executed. The <i>Environment</i> string specified is a maximum of 1200 characters, or the command is unsuccessful. Using the same rules that are used by the shell, the SRC sets up the environment for the subsystem. Quoted strings are assigned to a single environment variable and blanks outside quoted strings delimit each environment variable to be set. For example: <code>-e "HOME=/tmp TERM=dumb MESSAGE=\"Multiple word message\""</code> would set HOME=/tmp as the first, TERM=dumb as the second, and MESSAGE="Multiple word message" as the third environment variable for the subsystem.
-g <i>Group</i>	Specifies a group of subsystems to be started. The command is unsuccessful if the <i>Group</i> name is not contained in the subsystem object class.
-h <i>Host</i>	Specifies the foreign host on which this start action is requested. The local user must be running as "root". The remote system must be configured to accept remote System Resource Controller requests. That is, the srcmstr daemon (see /etc/inittab) must be started with the -r flag and the /etc/hosts.equiv or .rhosts file must be configured to allow remote requests.
-o <i>Object</i>	Specifies that a subserver object is to be passed to the subsystem as a character string. It is the subsystems responsibility to determine the validity of the <i>Object</i> string.
-p <i>SubsystemPID</i>	Specifies a particular instance of the subsystem to which the start subserver request is to be passed.
-s <i>Subsystem</i>	Specifies a subsystem to be started. The <i>Subsystem</i> can be the actual subsystem name or the synonym name for the subsystem. The command is unsuccessful if the <i>Subsystem</i> is not contained in the subsystem object class.
-t <i>Type</i>	Specifies that a subserver is to be started. The command is unsuccessful if <i>Type</i> is not contained in the subserver object class.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To start a subsystem with arguments and environment variables, enter:

```
startsrc -s srctest -a "-D DEBUG" -e "TERM=dumb HOME=/tmp"
```

This starts the `src`test subsystem with "TERM=dumb", "HOME=/tmp" in its environment and "-D DEBUG" as two arguments to the subsystem.

2. To start a subsystem group on a foreign host, enter:

```
startsrc -g tcpip -h zork
```

This starts all the subsystems in the subsystem `tcpip` group on the `zork` machine.

3. To start a subserver, enter:

```
startsrc -t tester
```

This sends a start subserver request to the subsystem that owns the `tester` subsystem.

4. To start a subsystem with command arguments, enter:

```
startsrc -s srctest -a "-a 123 -b \"4 5 6\""
```

This places "-a" as the first argument, "123" as the second, "-b" as the third, and "456" as the fourth argument to the `src`test subsystem.

Files

Item	Description
<code>/etc/objrepos/SRCsubsys</code>	Specifies the SRC Subsystem Configuration Object Class.
<code>/etc/objrepos/SRCsubsvr</code>	Specifies the SRC Subserver Configuration Object Class.
<code>/etc/services</code>	Defines the sockets and protocols used for Internet services.
<code>/dev/SRC</code>	Specifies the AF_UNIX socket file.
<code>/dev/.SRC-unix</code>	Specifies the location for temporary socket files.

startup Command

Purpose

Turns on accounting functions at system startup.

Syntax

```
/usr/sbin/acct/startup
```

Description

The **startup** command turns on the accounting functions when the system is started, if called by the `/etc/rc` command file. See the **startup** example for the command to add to the `/etc/rc` file.

Security

Access Control: This command should grant execute (x) access only to members of the `adm` group.

Examples

To turn on the accounting functions when the system is started, add the following to the `/etc/rc` file:

```
/usr/bin/su - adm -c /usr/sbin/acct/startup
```

The **startup** shell procedure will then record the time and clean up the previous day's records.

Files

Item	Description
<code>/usr/sbin/acct</code>	The path to the accounting commands.

startvsd Command

Purpose

startvsd – Makes a virtual shared disk available and activates it.

Syntax

startvsd [-p | -b] {-a | *vsd_name* ...}

Description

The **startvsd** command makes the specified virtual shared disks available and activates them. It is equivalent to running the **preparevsd** command followed by the **resumevsd** command on the specified virtual shared disk.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit vsd_mgmt
```

and select the **Start a Virtual Shared Disk** option.

Under normal circumstances, you should not issue this command. The Recoverable virtual shared disk subsystem uses this command to manage shared disks in a controlled manner. If you issue this command, the results may be unpredictable.

Flags

- p**
Specifies that the primary server node defined for the global volume group is to be the active server.
See the *RSCT: Managing Shared Disks* for more information.
- b**
Specifies that the secondary server node defined for the global volume group is to be the active server.
- a**
Specifies that all virtual shared disks that have been defined are to be started.

Parameters

vsd_name
Specifies a virtual shared disk.

Security

You must have root authority to run this command.

Exit Status

- 0**
Indicates the successful completion of the command.

nonzero

Indicates that an error occurred.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **starttrpdomain** command. To bring a particular node online in an existing peer domain, use the **starttrpnode** command. For more information on creating and administering an RSCT peer domain, refer to the *RSCT: Administration Guide*.

Under normal circumstances, you should not issue this command. The Recoverable virtual shared disk subsystem uses this command to manage shared disks in a controlled manner. If you issue this command, the results may be unpredictable.

Standard Output

Current RVSD subsystem run level.

Examples

To make available and activate the virtual shared disk **vsd1vg1n1**, enter:

```
startvsd vsd1vg1n1
```

Location

/opt/rsct/vsd/bin/startvsd

Related Information

Commands: **cfgvsd**, **lsvsd**, **preparevsd**, **resumevsd**, **stopvsd**, **suspendvsd**, **ucfgvsd**

startwpar Command

Purpose

Activates a workload partition.

Syntax

/usr/sbin/startwpar [**-a**] [**-m**] [**-v**] [**-1** [**-R**]] [**-2** [**-eVAR=values ...**]] [**-I**] *WparName*

Description

The **startwpar** command activates a workload partition that is defined by the **mkwpar** command. It includes:

- Exporting devices from the global environment into the workload partition
- Mounting the workload partition file systems
- Assigning and activating the workload partition IP addresses
- Activating the workload partition WLM class, if any
- Creating the **init** command
-

The **startwpar** command fails if no workload partition exists with the given name.

Flags

Item	Description
-1	Phase 1: the loaded state. Specifies the startwpar command to stop before creating or running any process. Only programmatic consumers (consumers with administrative lock) can use this -1 flag.
-2	Phase 2: starts initial processes. If the workload partition is already configured with the startwpar -1 option, use the -2 flag to complete the startup of the workload partition by spawning the registered application (application workload partitions), init (system workload partitions), or the registered alternate init if the workload partition was created with the -c (checkpointable) option of the mkwpar or wparexec commands. The operation context is identical to that of the normal startwpar operation for the type of workload partition that is queried. This option is in contrast with the -I option, whereby the startwpar process is replaced by the workload partition process. Only programmatic consumers can use this -2 flag.
-a	Automatically resolves conflicting static settings if they occurred. Resolvable settings include hostname and network configuration.
-e <i>VAR=values ...</i>	Allows customization of the environment available to the initial process created by the startwpar -2 flag. The parameter should be a single argument (appropriately quoted and escaped) in the form of <i>VAR=value ...</i> . Only programmatic consumers can use this -e flag.
-I	Specifies that the startwpar command to exec the initial process for the workload partition: /usr/lib/wpars/wparinit for system workload partitions, and /usr/lib/wpars/vinit for application workload partitions. The alternate init command, if registered, is never run through this flag. The process is created and run through exec , and replaces the startwpar process. This is in contrast to the -2 flag, whereby the initial process is run in its usual context. Only programmatic consumers can use this -I flag.
-m	Specifies that the workload partition should be started in maintenance mode. Networks that are associated with the workload partition are not configured, so the only access to the workload partition is from the global system. Do not use the -m flag to configure workload partitions with NFS file systems.
-R	When used with the -1 flag, the -R flag specifies that the workload partition is to be configured for restart rather than fresh start. Only programmatic consumers can use this -R flag.
-v	Specifies to show verbose output.

Parameters

Item	Description
<i>VAR=values ...</i>	Values that can be interpreted into the -e flag as a single argument by the shell. It can contain punctuations, spaces and so on.
<i>WparName</i>	The name of the workload partition to be started.

Security

Access Control: Only the root user can run this command.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

To start the workload partition called *roy*, enter:

```
startwpar roy
```

startx Command

Purpose

Initializes an X session.

Syntax

```
startx [ -d Display:0 ] [ -t | -w ] [ -x Startup | [ -r Resources ] [ -m Window_Manager ] ] [ -wait ]
```

Description

The **startx** command streamlines the process of starting an X session.

The command does the following:

- Sets the user's **DISPLAY** environment variable to identify the X server to the X clients
- When run from a workstation, starts the X server
- Starts the X clients.

The **startx** command redirects X server and X client error messages to the file specified by the user's **XERRORS** environment variable. This process is useful for debugging and gives the X server a clean startup and shutdown appearance on a workstation.

If a startup script file name is not given at the command line with the **-x** option, then the **startx** command searches for a file specified by the user's **XINITRC** environment variable. If the **XINITRC** environment variable is not set, then the **startx** command searches the user's home directory for a file called **.Xinit**, **.xinit**, **.Xinitrc**, **.xinitrc**, or **.xsession**, respectively, to begin the X client programs.

If a startup file is not found, the **startx** command runs the Window Manager indicated at the command line with the **-m** option, or invokes the window manager **mwm**, **twm**, **awm**, or **uwm** after finding the associated configuration file (**.mwmrc**, **.twmrc**, **.awmrc**, or **.uwmrc**, respectively). If a window manager configuration file is not found in the user's home directory, **startx** initiates an **Xterm** client and the **mwm** window manager.

When a startup file is not found, the **startx** command also instructs the loading of the resources file given at the command line with the **-r** option, or a file from the user's home directory called **.Xdefaults**, **.xdefaults**, **.Xresources**, or **.xresources**, respectively. If an X resources file is not found, then the X session will not be personalized.

If a startup file exists for a workstation and no resources are loaded by the user, then the **xinit** command within the **startx** command attempts to load an **.Xdefaults** file.

The use of a workstation is assumed when the X session is initiated from **/dev/lft***. If this is not the case, then the **-t** or **-w** option must be used.

Flags

Item	Description
-d <i>Display:0</i>	Specifies the display name of the X server to pass to the X clients during the process for startup.
-m <i>Window_Manager</i>	Starts the Window Manager when no startup script is found.
-r <i>Resources</i>	Loads the resources file when no startup script is found.
-t	Starts X clients for an X terminal.
-w	Starts the X server and X clients for an X window session on a workstation.
-wait	Prevents the X session from being restarted when the x dm command invokes startx .
-x Startup	Starts an X window session using the startup script.

Note: You can use one or both of the **-m** and **-r** options, or the **-x** option, but you cannot use the **-x** option with the **-m** and **-r** options. In the startup script, it is the responsibility of the user to start a window manager session, load X resources, and spawn X clients.

Examples

1. To start an X session on a workstation, or an X terminal, enter:

```
startx
```

2. To force start an X session on a workstation, enter:

```
startx -w
```

3. To start an X session for an X terminal, and log off the user's telnet session, enter:

```
startx; kill -9 $$
```

4. To start an X session using the **.xinitrc** script, enter:

```
startx -x .xinitrc
```

5. To start an X session using the **mwm** window manager, enter:

```
startx -m mwm
```

However, if a startup script file is found, the **-w** option is ignored.

6. In the startup script, it is the responsibility of the user to start a window manager, load X resources, and spawn X clients. The following is an example of an **.xsession** script.

```
#!/bin/csh
(mwm &)
xrdb -load .Xdefaults
(xclock -g 75x75+0+0 &)
(xbiff -g 75x75+101-0 &)
if ("/dev/lft*" == "`tty`") then
    aixterm -g 80x24+0+0 +ut -C -T `hostname`
else
    aixterm -g 80x24+0+0 +ut -T `hostname`
endif
```

For a workstation, the last line in the startup script should be a foreground **[aixterm](#)** command with the **-C** option for console messages.

For an X terminal, the last line in the startup script should be a foreground **aixterm** command without the **-C** option. In addition, because some X terminals do not terminate the **telnet** session upon closing, the user must exit the current telnet session before using hot keys to switch to the X session.

Also, the **startx** command can be used by the **xdm** command in the **/usr/lib/X11/xdm/Xsession** file. This provides the **xdm** command with the features of the **startx** command.

Files

The following file names have been historically used for the startup of an X session.

Item	Description
\$HOME/.xerrors	Where startx is to redirect error messages. By default, startx redirects errors to the .xerrors file in user's home directory.
\$HOME/.Xinit, \$HOME/.xinit, \$HOME/.Xinitrc, \$HOME/.xinitrc, \$HOME/.xsession	Used as a Startup file containing shell commands to start a window manager, load X resources, and spawn X clients.
\$HOME/.Xdefaults, \$HOME/.xresources	Used as an X resources file loaded to set user preferences for X clients.
\$HOME/.mwmrc	An mwm configuration file.
\$HOME/.twmrc	A twm configuration file.
\$HOME/.awmrc	An awm configuration file.
\$HOME/.uwmrc	A uwm configuration file.
/dev/lft*	The terminal, or tty, interface of a workstation's initial login shell.

statd Daemon

Purpose

Provides crash and recovery functions for the locking services on NFS.

Syntax

/usr/sbin/rpc.statd [**-d** *DebugLevel*] [**-D**] [**-t** *threads*]

Description

The **statd** daemon interacts with the **lockd** daemon to provide crash and recovery functions for the locking services on Network File System (NFS). The **statd** daemon must always be started before the **lockd** daemon.

The **statd** daemon is started and stopped by the following SRC commands:

```
startsrc -s rpc.statd
```

```
stopsrc -s rpc.statd
```

The status monitor maintains information on the location of connections as well as the status in the **/var/statmon/sm** directory, the **/var/statmon/sm.bak** directory, and the **/var/statmon/state** file. When

restarted, the **statd** daemon queries these files and tries to reestablish the connection it had prior to termination. To restart the **statd** daemon, and subsequently the **lockd** daemon, without prior knowledge of existing locks or status, delete these files before restarting the **statd** daemon.

Flags

Item	Description
-t <i>threads</i>	Specifies the maximum number of rpc.statd threads allowed. The Default value is 50.
-d <i>DebugLevel</i>	Specifies the debug level of rpc.statd. The debug level is disabled by default.
-D	Specifies which statmon directory to use. Without the -D flag, rpc.statd will use the /var/statmon directory. With the -D flag, rpc.statd will use the statmon directory under the current directory. The -D flag is disabled by default. Note: When statd is started manually with the startsrc command and using the -D flag, the current work directory (CWD) is used for srcmstr. Being srcmstr executed at boot for root and if anv \$HOME for root is different from /, eg /root then statmon data will go into /root/statmon directory.

statvsd Command

Purpose

Displays virtual shared disk device driver statistics of a node.

Syntax

statvsd

Description

The **statvsd** command displays virtual shared disk statistics of a node. For example, on a busy server an increasing number of "requests queued waiting for a buddy buffer" is normal and does not necessarily imply a problem. Of more value is the "average buddy buffer wait_queue size" which is the number of requests queued for a buddy buffer when the **statvsd** command was issued. See the "Examples" section for the meaning of output lines.

Flags

None.

Parameters

None.

Security

You must be in the AIX **bin** group to run this command.

Exit Status

- 0**
Indicates the successful completion of the command.

nonzero

Indicates that an error occurred.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **startprdomain** command. To bring a particular node online in an existing peer domain, use the **startprnode** command. For more information on creating and administering an RSCT peer domain, refer to the *RSCT: Administration Guide*.

Standard Output

Current RVSD subsystem run level.

Examples

The following examples display virtual shared disk device driver statistics.

1. The header line indicates the version and release of the code. For example:

```
VSD driver (vsdd): IP/SMP Version:4 Release:1
```

2. The level of virtual shared disk parallelism defaults to 9 and is the buf_cnt parameter on the uphysio call that the device driver makes in the kernel. For example:

```
9 vsd parallelism
```

3. The maximum IP message size in bytes. For example:

```
61440 vsd max IP message size
```

4. The number of requests that had to wait for a request block. For example:

```
61440 vsd max IP message size
```

5. The number of requests that had to wait for a pbuf (a buffer used for the actual physical I/O request submitted to the disk). For example:

```
0 requests queued waiting for a pbuf
```

6. The number of requests that had to wait for a buddy buffer. A buffer that is used on a server to temporarily store data for I/O operations originating at a client node. For example:

```
2689 requests queued waiting for a buddy buffer
```

7. The number of requests queued for a buddy buffer when the statvsd command was issued. For example:

```
0 average buddy buffer wait_queue size
```

8. The number of requests that a server has rejected, typically because of an out-of-range sequence number or an internal problem. For example:

```
4 rejected requests
```

9. The number of responses that a client has rejected. Typically because a response arrived after a retry was already sent to the server. For example:

```
0 rejected responses
```

10. The number of requests that were placed on the rework queue. For example:

```
0 requests rework
```

11. The number of read requests that were not on a 64 byte boundary. For example:

```
0 64 byte unaligned reads
```

12. The number of requests that got a DMA shortage. This condition would require the I/O operation to be executed in nonzero copy mode. For example:

```
0 DMA space shortage
```

13. The number of requests that have timed out. The current timeout period is approximately 15 minutes. For example:

```
0 timeouts
```

14. There are a fixed number of retries. The retries counters display the number of requests that have been retried for that particular "retry bucket." Numbers appearing further to the right represent requests that have required more retries. When a request exhausts its number of retries, it gets recorded as a timeout. For example:

```
retries: 0 0 0 0 0 0 0 0 0
          0 total retries
```

15. Sequence numbers are internally used by the device driver. These numbers are managed by the device driver and the Recoverable virtual shared disk subsystem. For example:

```
Non-zero Sequence Numbers
node#      expected      outgoing      outcase?      Incarnation:0
  11         125092         0             |
11 Nodes Up with zero sequence numbers: 1 3 5 7 9 11 12 13 14 15 16
```

Location

/opt/rsct/vsd/bin/statvsd

stop-secdapclntd Command

Purpose

The **stop-secdapclntd** script is used to terminate the **secdapclntd** LDAP client daemon.

Syntax

/usr/sbin/stop-secdapclntd

Description

The **stop-secdapclntd** script terminates the running **secdapclntd** daemon process. It returns an error if the **secdapclntd** daemon is not running.

Security

A user with the **aix.security.ldap** authorization is authorized to use this command.

Example

To stop the running **secdapclntd** daemon process, type:

```
/usr/sbin/stop-secdapclntd
```

Files

Item	Description
<code>/usr/sbin/stop-secdapclntd</code>	Used to terminate the secdapclntd LDAP client daemon.

stopcondresp Command

Purpose

Stops the monitoring of a condition that has one or more linked responses.

Syntax

To stop monitoring a condition:

```
stopcondresp [-q] [-h] [-TV] condition[:node_name] [response [response...]]
```

To unlock or lock the condition/response association:

```
stopcondresp {-U | -L} [-h] [-TV] condition[:node_name] response
```

Description

The `stopcondresp` command stops the monitoring of a condition that has one or more linked responses. If no response is specified, all of the linked responses for the condition are stopped. If one or more responses is specified, only those responses that are linked to the condition are stopped. When the condition occurs, the response is not run. If no responses are active for a condition, the condition is no longer monitored.

If a particular condition/response association is needed for system software to work properly, it may be locked. A locked condition/response association cannot be stopped by the `stopcondresp` command. If the condition/response link you specify on the `stopcondresp` command is locked, it will not be stopped; instead an error will be generated informing you that the condition/response association is locked. To unlock a condition/response association, you can use the `-U` flag. A condition/response association is typically locked because it is essential for system software to work properly, so you should exercise caution before unlocking it.

Flags

-q

Does not return an error when either *condition* or *response* does not exist or when the *condition* linked with *response* is not being monitored.

-h

Writes the command's usage statement to standard output.

-T

Writes the command's trace messages to standard error. For your software service organization's use only.

-V

Writes the command's verbose messages to standard output.

-U

Unlocks a condition/response association so it can be started, stopped, or removed. If a condition/response association is locked, this is typically because it is essential for system software to work properly. For this reason, you should exercise caution before unlocking it. When unlocking a condition/response association using the `-U` flag, no other operation can be performed by this command.

-L

Locks a condition/response association so it cannot be started, stopped, or removed. When locking a condition/response association using the -L flag, no other operation can be performed by this command.

Parameters

condition

Specifies the name of the condition linked to the response. The condition is always specified first.

node_name

Specifies the node in the domain where the condition is defined. If *node_name* is not specified, the local node is used. *node_name* is a node within the scope determined by the CT_MANAGEMENT_SCOPE environment variable.

response

Specifies the names of one or more responses. Monitoring is stopped for the specified responses. (If a specified response is not linked to the condition, it is ignored.)

Security

The user needs write permission for the IBM.Association resource class to run stopcondresp. Permissions are specified in the access control list (ACL) file on the contacted system. See the *RSCT: Administration Guide* for details on the ACL file and how to modify it.

Exit Status

0

The command ran successfully.

1

An error occurred with RMC.

2

An error occurred with a command-line interface script.

3

An incorrect flag was entered on the command line.

4

An incorrect parameter was entered on the command line.

5

An error occurred that was based on incorrect command-line input.

Environment Variables

CT_CONTACT

Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When CT_CONTACT is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If CT_CONTACT is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. CT_IP_AUTHENT only has meaning if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon in processing the resources of the event-response resource manager (ERRM). The management scope determines the set of possible target nodes where the resources can be processed. The valid values are:

- 0** Specifies *local* scope.
- 1** Specifies *local* scope.
- 2** Specifies *peer domain* scope.
- 3** Specifies *management domain* scope.

If this environment variable is *not* set, *local* scope is used.

Implementation Specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) fileset for AIX.

Standard Output

When the `-h` flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

These examples apply to standalone systems:

1. To stop monitoring for the condition "FileSystem space used " which has the response "Broadcast event on-shift" linked with it, run this command:

```
stopcondresp "FileSystem space used" "Broadcast event on-shift"
```

2. To stop monitoring for the condition "FileSystem space used " using all of its linked responses, run this command:

```
stopcondresp "FileSystem space used"
```

This example applies to management domains:

1. To stop monitoring for the condition "FileSystem space used " on the managed node nodeB which has the response "Broadcast event on-shift" linked with it, run this command on the management server:

```
stopcondresp "FileSystem space used:nodeB" "Broadcast event on-shift"
```

This example applies to peer domains:

1. To stop monitoring for the condition "FileSystem space used " on the node nodeA which has the response "Broadcast event on-shift" linked with it, run this command on any node in the domain:

```
stopcondresp "FileSystem space used:nodeA" "Broadcast event on-shift"
```

Location

/opt/rsct/bin/stopcondresp

stoprpdomain Command

Purpose

Takes an online peer domain offline.

Syntax

```
stoprpdomain [-f] [-h] [-w [-s Seconds]] [-TV] peer_domain
```

Description

The stoprpdomain command takes all of the nodes that are currently online in the peer domain offline. The peer domain definition is not removed from the nodes.

The command must be run on a node that is online in the peer domain. If the command is run on a node that is offline to the peer domain, no action is performed.

If a Cluster-Aware AIX (CAA) cluster is configured, no action is performed because a peer domain operation in a CAA environment exists and is online for the life of the CAA cluster.

The -f flag must be used to override a subsystems rejection of the request to take the peer domain offline. A subsystem may reject the request if a peer domain resource is busy, such as in the case of a shared disk. Specifying the -f flag in this situation indicates to the subsystems that the peer domain must be brought offline regardless of the resource state.

Flags

-f

Forces the subsystems to accept the stop request when it otherwise would not.

-h

Writes the command's usage statement to standard output.

-s

Specifies the wait time in seconds for the peer domain to be offline before the command completes when the -s flag is used with the -w flag. If the waiting time exceeds the number of seconds, the command returns, but the offline operation continues. The default value is 300 seconds (5 minutes). Use 0 to specify that the command must not return until the peer domain is offline (no timeout on waiting).

-T

Writes the command's trace messages to standard error. For your software service organization's use only.

-V

Writes the command's verbose messages to standard output.

-w

Waits for the peer domain to be offline before the command completes. Use the -s flag to specify the waiting time in seconds.

Parameters

peer_domain

Specifies the name of the online peer domain that is to be brought offline.

Security

The user of the stoprpdomain command needs write permission for the IBM.PeerDomain resource class on each node that is defined to the peer domain. By default, root on any node in the peer domain has read and write access to this resource class through the configuration resource manager.

Exit Status

0

The command ran successfully.

1

An error occurred with RMC.

2

An error occurred with a command-line interface script.

3

An incorrect flag was entered on the command line.

4

An incorrect parameter was entered on the command line.

5

An error occurred that was based on incorrect command-line input.

6

The peer domain definition does not exist.

Environment Variables

CT_CONTACT

Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When CT_CONTACT is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If CT_CONTACT is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. CT_IP_AUTHENT only has meaning if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

Restrictions

This command must be run on a node that is online in the peer domain.

Implementation Specifics

This command is part of the **rsct.basic.rte** fileset for the AIX® operating system.

Standard Input

When the **-f** " " or **-F** " " flag is specified, this command reads one or more node names from standard input.

Standard Output

When the **-h** flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

In these examples, nodeA is one of the nodes defined and is online to App1Domain.

1. To take ApplDomain offline, run this command on nodeA:

```
stoprpdomain ApplDomain
```

2. To take ApplDomain offline while making sure the stop request will not be rejected by any subsystem, run this command on nodeA:

```
stoprpdomain -f ApplDomain
```

Location

/opt/rsct/bin/stoprpdomain

stoprpnode Command

Purpose

Takes one or more nodes offline from a peer domain.

Syntax

```
stoprpnode [-f] [-h] [-w [-s Seconds]] [-TV] node_name1 [node_name2...]
```

```
stoprpnode -F { file_name | "-" } [-f] [-h] [-w [-s Seconds]] [-TV]
```

Description

The stoprpnode command takes an online node offline from a peer domain. The peer domain is determined by the online peer domain where the command is run. The command must be run from a node that is online to the desired peer domain.

If a Cluster-Aware AIX (CAA) cluster is configured, no action is performed because a peer domain operation in a CAA environment exists and is online for the life of the CAA cluster.

The -f flag must be used to override a subsystem's rejection of the request to take a node offline. A subsystem may reject the request if a node resource is busy, such as in the case of a shared disk. Specifying the -f flag in this situation indicates to the subsystems that the node must be brought offline regardless of the resource state.

If this command is used to take more than one node offline by specifying more than one *node_name* parameter, and the node that this command is running on is in the list, it will be brought offline last.

Flags

-f

Forces the subsystems to accept the stop request when it otherwise would not.

-F { file_name | "-" }

Reads a list of node names from *file_name*. Each line of the file is scanned for one node name. The pound sign (#) indicates that the remainder of the line (or the entire line if the # is in column 1) is a comment.

Use -F "-" to specify STDIN as the input file.

-h

Writes the command's usage statement to standard output.

-s

Specifies the wait time in seconds for all of the specified nodes to be offline before the command completes when the -s flag is used with the -w flag. If the waiting time exceeds the number of seconds, the command returns, but the offline operation continues. The default value is 300 seconds

(5 minutes). Use 0 to specify that the command must not return until all of the specified nodes are offline (no timeout on waiting).

-T

Writes the command's trace messages to standard error. For your software service organization's use only.

-V

Writes the command's verbose messages to standard output.

-w

Waits for all of the specified nodes to be offline before the command completes. Use the **-s** flag to specify the waiting time in seconds.

Parameters

node_name1 [node_name2...]

Specifies the peer domain node names of the nodes that are to be brought offline from the peer domain. You must specify the node names in exactly the same format as they were specified with the `addrpnnode` command or the `mkrpnomain` command. To list the peer domain node names, run the `lsrpnnode` command.

Security

The user of the `stoprpnnode` command needs write permission for the `IBM.PeerNode` resource class on each node that is to be stopped in the peer domain. By default, `root` on any node in the peer domain has read and write access to this resource class through the configuration resource manager.

Exit Status

0

The command ran successfully.

1

An error occurred with RMC.

2

An error occurred with a command-line interface script.

3

An incorrect flag was entered on the command line.

4

An incorrect parameter was entered on the command line.

5

An error occurred that was based on incorrect command-line input.

Environment Variables

CT_CONTACT

Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When `CT_CONTACT` is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If `CT_CONTACT` is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

CT_IP_AUTHENT

When the `CT_IP_AUTHENT` environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the `CT_CONTACT` environment variable is set. `CT_IP_AUTHENT` only has meaning if `CT_CONTACT` is set to an IP address; it does not rely on the domain name system (DNS) service.

Restrictions

This command must be run on a node that is online to the peer domain. The node to be brought offline must be reachable from the node on which the command is run.

Implementation Specifics

This command is part of the **rsct.basic.rte** fileset for the AIX® operating system.

Standard Input

When the **-F** **" - "** flag is specified, this command reads one or more node names from standard input.

Standard Output

When the **-h** flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

In these examples, nodeA and nodeB are online to ApplDomain.

1. To take nodeB offline, run this command on nodeA:

```
stopprnode nodeB
```

2. To take nodeB offline and force the offline request, run this command on nodeA:

```
stopprnode -f nodeB
```

Location

/opt/rsct/bin/stopprnode

stopprsrc Command

Purpose

Stops a resource (that is, takes it offline).

Syntax

To stop one or more resources, using data entered on the command line:

```
stopprsrc -s "selection_string" [ -N { node_file | " - " } ] [-h] [-TV] resource_class [arg=value...]
```

```
stopprsrc -r [-h] [-TV] resource_handle [arg=value...]
```

To stop one or more resources using command arguments that are predefined in an input file:

```
stopprsrc -f resource_data_input_file -s "selection_string" [ -N { node_file | " - " } ] [-h] [-TV]  
resource_class
```

```
stopprsrc -f resource_data_input_file -r [-h] [-TV] resource_handle
```

To list the names and data types of the command arguments:

```
stopprsrc -l [-h] resource_class
```

Description

The `stoprsrc` command requests that the resource monitoring and control (RMC) subsystem take one or more resources offline. The request is performed by the appropriate resource manager.

To stop one or more resources, use the `-s` flag to take offline all of the resources that match the specified selection string.

Instead of specifying multiple node names in *selection_string*, you can use the `-N node_file` flag to indicate that the node names are in a file. Use `-N " - "` to read the node names from standard input.

To stop one specific resource, use the `-r` flag to specify the resource handle that represents that specific resource.

Use the `-l` flag to determine whether the specified resource class accepts any additional command arguments.

If Cluster Systems Management (CSM) is installed on your system, you can use CSM defined node groups as node name values to refer to more than one node. For information about working with CSM node groups and using the `CSM nodegrp` command, see the *CSM: Administration Guide* and the *CSM: Command and Technical Reference*.

The successful completion of this command does not guarantee that the resource is offline, only that the resource manager successfully received the request to take this resource offline. Monitor the resource dynamic attribute `OpState` to determine when the resource is taken offline. Register an event for the resource, specifying the `OpState` attribute, to know when the resource is offline. Or, intermittently run the `lsrsrc` command until you see that the resource is offline (the value of `OpState` is 2). For example:

```
lsrsrc -s 'Name == "/filesystem1"' -t IBM.FileSystem Name OpState
```

Parameters

resource_class

Specifies the name of the resource class that contains the resources that you want to take offline.

resource_handle

Specifies the resource handle that corresponds to the resource you want to take offline. Use the `lsrsrc` command to obtain a list of valid resource handles. The resource handle must be enclosed within double quotation marks, for example:

```
"0x4017 0x0001 0x00000000 0x0069684c 0x0d4715b0 0xe9635f69"
```

arg=value...

Specifies one or more pairs of command argument names and values.

arg

Specifies the argument name.

value

Specifies the value for this argument. The value datatype must match the definition of the argument datatype.

Command arguments are optional. If any *arg=value* pairs are entered, there should be one *arg=value* pair for each command argument defined for the offline function for the specified resource class.

Use `stoprsrc -l` to get a list of the command argument names and datatypes for the specific resource class.

Flags

-f *resource_data_input_file*

Specifies the name of the file that contains resource argument information. The contents of the file would look like this:

```
PersistentResourceArguments::  
argument1 = value1  
argument2 = value2
```

-l

Lists the command arguments and data types. Some resource managers accept additional arguments that are passed to the offline request. Use this flag to list any defined command arguments and the data types of the command argument values.

-N { *node_file* | "-" }

Specifies that node names are read from a file or from standard input. Use `-N node_file` to indicate that the node names are in a file.

- There is one node name per line in *node_file*
- A number sign (#) in column 1 indicates that the line is a comment
- Any blank characters to the left of a node name are ignored
- Any characters to the right of a node name are ignored

Use `-N "-"` to read the node names from standard input.

The `CT_MANAGEMENT_SCOPE` environment variable determines the scope of the cluster. If `CT_MANAGEMENT_SCOPE` is not set, management domain scope is chosen first (if a management domain exists), peer domain scope is chosen next (if a peer domain exists), and then local scope is chosen, until the scope is valid for the command. The command runs once for the first valid scope it finds. For example, if a management domain and a peer domain both exist and `CT_MANAGEMENT_SCOPE` is not set, this command applies to the management domain. If you want this command to apply to the peer domain, set `CT_MANAGEMENT_SCOPE` to 2.

-s "*selection_string*"

Specifies the selection string. All selection strings must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'  
-s 'Name ?= "test"'
```

Only persistent attributes can be listed in a selection string.

-h

Writes the command usage statement to standard output.

-T

Writes the command trace messages to standard error. For your software service organization use only.

-V

Writes the command verbose messages (if there are any available) to standard output.

Environment variables

CT_CONTACT

When the `CT_CONTACT` environment variable is set to a host name or IP address, the command contacts the resource monitoring and control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are on the system to which the connection is established.

CT_IP_AUTHENT

When the CT_IP_AUTHENT environment variable exists, the RMC daemon uses IP-based network authentication to contact the RMC daemon on the system that is specified by the IP address to which the CT_CONTACT environment variable is set. CT_IP_AUTHENT only has meaning if CT_CONTACT is set to an IP address; it does not rely on the domain name system (DNS) service.

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon to monitor and control the resources and resource classes. The management scope determines the set of possible target nodes where the resources and resource classes can be monitored and controlled. The valid values are:

- 0** Specifies *local* scope.
- 1** Specifies *local* scope.
- 2** Specifies *peer domain* scope.
- 3** Specifies *management domain* scope.

If this environment variable is *not* set, *local* scope is used.

Standard output

When the -h flag is specified, this command usage statement is written to standard output. When the -V flag is specified, this command verbose messages (if there are any available) are written to standard output.

Standard error

All trace messages are written to standard error.

Exit status

- 0** The command ran successfully.
- 1** An error occurred with RMC.
- 2** An error occurred with the command-line interface (CLI) script.
- 3** An incorrect flag was specified on the command line.
- 4** An incorrect parameter was specified on the command line.
- 5** An error occurred with RMC that was based on incorrect command-line input.
- 6** No resources were found that match the specified selection string.

Security

You need write permission for the *resource_class* specified in stoprsrc to run stoprsrc. Permissions are specified in the access control list (ACL) file on the contacted system. See the *Administering RSCT* guide for information about the ACL file and how to modify it.

Implementation specifics

This command is part of the `rsct.core.rmc` fileset for the AIX operating system.

Location

`/opt/rsct/bin/stopsrc`

Examples

Suppose that you have a peer domain called `foo` with three defined nodes: `nodeA`, `nodeB`, and `nodeC`. `nodeA` has two Ethernet cards: `ent0` and `ent1`.

1. Suppose `nodeA` is online and `ent0` (on `nodeA`) is also online. To take `ent0` offline on `nodeA`, run this command on `nodeA`:

```
stopsrc -s 'Name == "ent0"' IBM.EthernetDevice
```

2. Suppose `nodeA` and `nodeB` are online, `ent0` (on `nodeA`) is also online, and you are currently logged on to `nodeB`. To take `ent0` offline on `nodeA`, run this command on `nodeB`:

```
stopsrc -s 'NodeName == "A" AND Name == "ent0"' IBM.EthernetDevice
```

3. Suppose `nodeA` and `nodeB` are online and file system `/filesys1` is defined and mounted on `nodeB`. To take `/filesys1` offline on `nodeB`, run this command on `nodeA`:

```
stopsrc -s 'NodeName == "B" AND Name == "/filesys1"' IBM.FileSystem
```

4. Suppose the resource handle for `ent0` on `nodeA` is:

```
0x406b 0x0001 0x00000000 0x0069564c 0x0dc1f272 0xb9de145e
```

To take `ent0` offline on `nodeA`, run this command on `nodeA`:

```
stopsrc -r "0x406b 0x0001 0x00000000 0x0069564c 0x0dc1f272 0xb9de145e"
```

stopsrc Command

Purpose

Stops a subsystem, a group of subsystems, or a subserver.

Syntax

To Stop a Subsystem

```
stopsrc [ -h Host ] [ -f | -c ] { -a | -g Group | -p SubsystemPID | -s Subsystem }
```

To Stop a Subserver

```
stopsrc [ -h Host ] [ -f ] -t Type [ -p SubsystemPID ] [ -P SubserverPID | -o Object ]
```

Description

The **stopsrc** command sends a request to the System Resource Controller (SRC) to stop a subsystem, a group of subsystems, or all subsystems. The **stopsrc** command sends the System Resource Controller a subsystem request packet that is forwarded to the subsystem for a stop subserver request.

In the absence of the **-f** (stop force) flag, a normal stop action is assumed. A normal stop requests that a subsystem or subserver complete all current processing, release resources when all application activity has been completed, and then end. No new requests for work should be accepted by the subsystem.

A forced stop requests that a subsystem or subserver end quickly, releasing all resources, but not wait for application activity to complete.

A cancel action stops the subsystem after the subsystem's resources are released and after a grace period. This grace period is specified in the subsystem object class. The cancel stop is used only for subsystem stops and is always sent to the subsystem as the **SIGTERM** signal. The subsystem should catch this signal, perform subsystem clean up operations, and end. If the subsystem does not end within the wait time period, specified in the subsystem object class, the subsystem is sent a **SIGKILL** signal to ensure that the subsystem stops.

If the subsystem uses sockets or message queues for communication, a packet is constructed and sent to the subsystem. If the subsystem uses signals for communication, the subsystem is sent the appropriate signal from the subsystem object class.

Flags

Item	Description
-a	Specifies that all subsystems are to be stopped.
-c	Specifies that the stop request is a canceled stop request. For a cancel stop request, a SIGTERM signal is sent to the subsystem. After the wait time contained in the subsystem object class has passed, if the subsystem has not yet ended, the subsystem is sent a SIGKILL signal.
-f	Specifies a forced stop request.
-g Group	Specifies that a group of subservers is to be stopped. The command is unsuccessful if the <i>Group</i> name is not contained in the subsystem object class.
-h Host	Specifies the foreign <i>Host</i> machine on which this stop action is requested. The local user must be running as "root". The remote system must be configured to accept remote System Resource Controller requests. That is, the srcmstr daemon (see /etc/inittab) must be started with the -r flag and the /etc/hosts.equiv or .rhosts file must be configured to allow remote requests.
-o Object	Specifies that a subserver <i>Object</i> value is to be passed to the subsystem as a character string.
-p SubsystemPID	Specifies a particular instance of the subsystem to stop, or a particular instance of the subsystem to which the stop subserver request is to be passed.
-P SubserverPID	Specifies that a subserver PID is to be passed to the subsystem as a character string.
-s Subsystem	Specifies a subsystem to be stopped. The <i>Subsystem</i> parameter can be the actual subsystem name or the synonym name for the subsystem. The stopsrc command stops all currently active instances of the subsystem. The command is unsuccessful if the <i>Subsystem</i> name is not contained in the subsystem object class.
-t Type	Specifies that a subserver is to be stopped. The stopsrc command is unsuccessful if the <i>Type</i> specified is not contained in the subserver object class.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To stop force a subsystem on a foreign host, enter:

```
stopsrc -h zork -s srctest -f
```

This forces a stop on all the instances of the `srctest` subsystem on the `zork` machine.

2. To stop cancel a subsystem group, enter:

```
stopsrc -g tcpip -c
```

This activates a stop cancel on all the subsystems in the `tcpip` group.

3. To stop a subserver, enter:

```
stopsrc -t tester -p 1234
```

This stops the `tester` subserver that belongs to the `srctest` subsystem with a subsystem PID of 1234.

4. To stop all subsystems, enter:

```
stopsrc -a
```

This stops all the active subsystems on the local machine.

Files

Item	Description
<code>/etc/objrepos/SRCsubsys</code>	Specifies the SRC Subsystem Configuration Object Class.
<code>/etc/objrepos/SRCsubsvr</code>	Specifies the SRC Subserver Configuration Object Class.
<code>/etc/services</code>	Defines the sockets and protocols used for Internet services.
<code>/dev/SRC</code>	Specifies the AF_UNIX socket file.
<code>/dev/.SRC-unix</code>	Specifies the location for temporary socket files.

stopvsd Command

Purpose

stopvsd – Makes a virtual shared disk unavailable.

Syntax

```
stopvsd {-a | vsd_name ...}
```

Description

The **stopvsd** command brings the specified virtual shared disks from the suspended state to the stopped state. This makes the virtual shared disks unavailable. All applications that have outstanding requests for the virtual shared disk see these requests terminate with error. Read and write requests return errors with **errno** set to **ENODEV**. If the virtual shared disk is in the stopped state, this command leaves it in the stopped state.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit vsd_mgmt
```

and select the **Stop a Virtual Shared Disk** option.

Under normal circumstances, you should not issue this command. The Recoverable virtual shared disk subsystem uses this command to manage shared disks in a controlled manner. If you issue this command, the results may be unpredictable.

Flags

-a

Specifies that all virtual shared disks in the suspended state are to be stopped.

Parameters

vsd_name

Specifies a virtual shared disk. If the virtual shared disk is not in the suspended state, you get an error message.

Security

You must have root authority to run this command.

Exit Status

0

Indicates the successful completion of the command.

nonzero

Indicates that an error occurred.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **starttrpdomain** command. To bring a particular node online in an existing peer domain, use the **starttrpnode** command. For more information on creating and administering an RSCT peer domain, refer to the *RSCT: Administration Guide*.

Under normal circumstances, you should not issue this command. The Recoverable virtual shared disk subsystem uses this command to manage shared disks in a controlled manner. If you issue this command, the results may be unpredictable.

Standard Output

Current RVSD subsystem run level.

Examples

To bring the virtual shared disk **vsd1vg1n1** from the suspended state to the stopped state, enter:

```
stopvsd vsd1vg1n1
```

Location

/opt/rsct/vsd/bin/stopvsd

stopwpar Command

Purpose

Deactivates an active workload partition.

Syntax

`/usr/sbin/stopwpar[ -h | -F ][ -r ][ -t seconds | -N ][ -v ] WparName`

Description

The `stopwpar` command deactivates a running workload partition. This includes stopping the following tasks:

- Stopping processes running within the workload partitions.
- Unloading the workload partition's WLM class, if any.
- Deactivating the workload partition's IP addresses, if any.
- Unmounting the workload partition's file systems, if any.
- Restarting the system workload partition.
- Removing the application workload partition.

The `stopwpar` command fails under the following circumstances:

- The specified workload partition does not exist.
- One or more processes cannot be stopped by the **kill** command (use the `-F` flag to force.)
- One or more file systems cannot be unmounted (use the `-F` flag to force.)

Flags

Item	Description
-F	Forces the workload partition to stop and signals the running processes more aggressively and unmounts the remote file systems. If the processes cannot be stopped, the workload partition remains in the Broken state and cannot be restarted.
-h	Uses a hard stop to signal the workload partition subsystems to end. The default timeout value is 60 seconds when using a hard stop.
-N	Specifies that the shutdown/halt to complete with no timeout.
-r	Restarts the workload partition after all stopping operations complete. This is equivalent to calling the <code>startwpar</code> command after the <code>stopwpar</code> command. This flag is not valid for application workload partitions.
-t <i>seconds</i>	Specifies the timeout length in number of seconds to wait for shutdown/halt to complete before the command fails and the program exits. The default is to fail after 600 seconds if the shutdown/halt has not completed.
-v	Specifies to show verbose output.

Parameters

Item	Description
<i>WparName</i>	Name of workload partition to stop. This parameter must be the last parameter on the command line.

Security

Access Control: Only the root user can run this command for system workload partitions. For application workload partitions, only the creator of the workload partition (or root) can run this command.

Examples

1. To stop the workload partition called *roy*, enter:

```
stopwpar roy
```

2. To discontinue the shutdown processing for the workload partition called *pinto* after 85 seconds, enter:

```
stopwpar -t 85 pinto
```

stpinet Method

Purpose

Disables the inet instance.

Syntax

```
stpinet [ -l "Interface ..." ] [ -t Time ]
```

Description

If **stpinet** is started with a list of network interfaces specified with the **-l** option, then this method only stops those IFs. Otherwise, **stpinet** informs users of the impending demise of TCP/IP, using the **wall** command, and invokes the **ifconfig** command to mark each configured IF as **down**. If no network interfaces are specified, the status flag of the inet instance is set to DEFINED.

Flags

Item	Description
-l "Interface ..."	Specifies the name of the interface to be disabled.
-t Time	Specifies the time in minutes until the inet instance is stopped.

Examples

The following example disables the inet instance `tr0` five minutes from the time the method is executed:

```
stpinet -l "tr0" -t 5
```

strace Command

Purpose

Prints STREAMS trace messages.

Syntax

```
strace [ mid sid level ] ...
```

Description

The **strace** command without parameters writes all STREAMS event trace messages from all drivers and modules to its standard output. These messages are obtained from the STREAMS **log** driver. If parameters are provided, they must be in triplets. Each triplet indicates that tracing messages are to be received from

the given module or driver, subID (usually indicating minor device), and priority level equal to or less than the given level. The `all` token may be used for any member to indicate no restriction for that attribute.

Parameters

Item	Description
<i>mid</i>	Specifies a STREAMS module ID number.
<i>sid</i>	Specifies a subID number.
<i>level</i>	Specifies a tracing priority level.

Output Format

The format of each trace message output is:

```
<seq> <time> <ticks> <level> <flags> <mid> <sid> <text>
```

Item	Description
<seq>	Trace sequence number
<time>	Time of message in <i>hh:mm:ss</i>
<ticks>	Time of message, in machine ticks, since system was started
<level>	Tracing priority level
<flags>	Has one of the following values: E Message is also in the error log F Indicates a fatal error N Mail was sent to the system administrator
<mid>	Module ID number of source
<sid>	SubID number of source
<text>	Formatted text of the trace message On multiprocessor systems, <text> is composed of two parts: • the number of the processor where the owner of the message has sent it, • the formatted text itself.

Once initiated, the **strace** command continues to execute until terminated by the user.

Note: Due to performance considerations, only one **strace** command is permitted to open the STREAMS log driver at a time. The log driver has a list of the triplets specified in the command invocation, and compares each potential trace message against this list to decide if it should be formatted and sent up to the **strace** process. Hence, long lists of triplets have a greater impact on overall STREAMS performance. Running the **strace** command has the most impact on the timing of the modules and drivers generating the trace messages that are sent to the **strace** process. If trace messages are generated faster than the **strace** process can handle them, some of the messages will be lost. This last case can be determined by examining the sequence numbers on the trace messages output.

Examples

1. To output all trace messages from the module or driver whose module ID is 41, enter:

```
strace 41 all all
```

2. To output those trace messages from driver or module ID 41 with sub-IDs 0, 1, or 2:

```
strace 41 0 1 41 1 1 41 2 0
```

Messages from sub-IDs 0 and 1 must have a tracing level less than or equal to 1. Those from sub-ID 2 must have a tracing level of 0.

strchg Command

Purpose

Changes stream configuration.

Syntax

To push modules onto a stream:

strchg -h *Module1* [, *Module2* ...]

To pop modules off a stream:

strchg -p [**-a** | **-u** *Module*]

To push and pop modules to conform to the configuration file:

strchg -f *File*

Description

The **strchg** command is used to alter the configuration of the stream associated with the user's standard input. The **strchg** command **pushes modules** on the stream, pops modules off of the stream, or both. Only the root user or owner of a STREAMS device can alter the configuration of that stream. If another user attempts to alter the configuration, the **strchg** command will not succeed.

Note: If modules are pushed in the wrong order, the stream might not function as expected.

Flags

Item	Description
-a	Pops all modules above the topmost driver off of a stream. The -p flag must be used in front of the -a flag.
-f <i>File</i>	Pushes and pops the necessary modules to conform the stream to the configuration given in the specified file. The -h , -p , and -f flags are mutually exclusive.
-h <i>Module1</i>	Pushes modules onto a stream. The modules are listed on the command line in the order they are to be pushed.
-p	Pops a module off of a stream. Used alone, the -p flag pops the topmost module from the stream.
-u <i>Module</i>	Pops all modules above the specified module off of a stream. The -p flag must be used in front of the -u flag. The -a and -u flags are mutually exclusive.

Parameters

Item	Description
<i>Module1</i>	Specifies the module to be pushed onto a stream. (Used by the -h flag.)
<i>Module</i>	Specifies the topmost module to remain on a stream. All modules above this module are popped off of the stream. (Used by the -u flag.)
<i>File</i>	Contains a list of modules representing the desired configuration of the stream. Each module name must appear on a separate line, where the first name represents the topmost module and the last name represents the module that is closest to the driver.

Return Values

On successful completion, the **strchg** command returns a value of 0. Otherwise, it returns a nonzero value and prints an error message indicating usage error, a bad module name, too many modules to push, failure of an **ioctl** operation on the stream, or failure to open the file specified by the *File* parameter.

Examples

1. To push the `ldterm` module on the stream, enter:

```
strchg -h ldterm
```

2. To pop the topmost module from the stream associated with the `/dev/term/24` device, enter:

```
strchg -p < /dev/term/24
```

The user must be the owner of this device or the root user.

3. If the `fileconf` file contains the following:

```
compat
ldterm
ptem
```

the following command configures the stream so that the `ptem` module is pushed over the driver, followed by the `ldterm` module, and the `compat` module is pushed closest to the stream head.

```
strchg -f fileconf
```

strclean Command

Purpose

Cleans up the STREAMS error logger.

Syntax

strclean [**-d**] [**-a** *Age*]

Description

The **strclean** command is used to clean up the STREAMS error-logger directory on a regular basis: for example, by using the **cron** daemon. By default, all files with names matching **error.*** in the **/var/adm/streams** directory that have not been modified in the last three days are removed.

Note: The **strclean** command is typically run using the **cron** daemon on a daily or weekly basis.

Flags

Item	Description
-a Age	Specifies the maximum age, in days, for a log file.
-d	Specifies a directory other than the default directory.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

The following example has the same result as running the **strclean** command with no parameters.

```
strclean -d /var/adm/streams -a 3
```

Files

Item	Description
/var/adm/streams/error.*	Contains the STREAMS error log.

strconf Command

Purpose

Queries stream configuration.

Syntax

strconf [**-t** | **-m** *module*]

Description

The **strconf** command is used to query the configuration of a stream. When used without any flags, it prints a list of all the modules in the stream as well as the topmost driver. The list is printed with one name per line, where the first name printed is the topmost module on the stream and the last item printed is the name of the driver.

Note: The **strconf** command only reads from standard input.

Flags

Item	Description
-m <i>Module</i>	Determines if the specified module is present on the stream. If the module is present, the strconf command prints the message yes and returns a value of 0. If it is not present, the strconf command prints the message no and returns a nonzero value. The -t and -m flags are mutually exclusive.
-t	Prints only the topmost module of the stream (if one exists).

Parameter

Item	Description
<i>Module</i>	Specifies the module for which to look.

Examples

1. For a stream that has only the `ldterm` module pushed above the `ports` driver, the **strconf** command (with no flags) would produce the following output:

```
ldterm
ports
```

2. Entering the following command asks if the `ldterm` module is on the stream:

```
strconf -m ldterm
```

The command produces the following output while returning an exit status of 0:

```
yes
```

strerr Daemon

Purpose

Receives error log messages from the STREAMS **log driver** .

Syntax

strerr

Description

The **strerr** daemon receives error log messages from the STREAMS log driver and appends them to a log file. The error log files produced reside in the directory **/var/adm/streams**, and are named **error.mm-dd**, where *mm* is the month and *dd* is the day of the messages contained in each log file.

The format of an error log message is:

```
<seq> <time> <ticks> <flags> <mid> <sud> <text>
```

These fields are defined as follows:

Item	Description
<seq>	Error sequence number
<time>	Time of message in <i>hh:mm:ss</i>
<ticks>	Time of message in machine ticks since boot priority level
<flags>	Has one of the following values: T The message was also sent to a tracing process F Indicates a fatal error N Send mail to the person who administers your system

Item	Description
<mid>	Module ID number of source
<sid>	Sub-ID number of source
<text>	Formatted text of the error message
	On multiprocessor systems, <text> is composed of two parts: • the number of the processor where the owner of the message has sent it, • the formatted text itself.

Messages that appear in the error log are intended to report exceptional conditions that require the attention of the person who administers your system. Those messages indicating the total failure of a **STREAMS driver or module** should have the **F** flag set. Those messages requiring the immediate attention of the administrator should have the **N** flag set, which causes the error logger to send the message to that person by way of the **mail** command. The priority level usually has no meaning in the error log, but does have meaning if the message is also sent to a tracer process.

Once initiated, the **strerr** daemon continues to execute until terminated by the user. Usually, the **strerr** daemon is executed asynchronously.

Note: Only one **strerr** daemon at a time is permitted to open the STREAMS log driver. If a module or driver is generating a large number of error messages, running the error logger causes a degradation in STREAMS performance. If a large number of messages are generated in a short time, the log driver may not be able to deliver some of the messages. This situation is indicated by gaps in the sequence numbering of the messages in the log files.

Files

Item	Description
<i>/var/adm/streams/error.mm-dd</i>	Error log file.

strinfo Command

Purpose

Displays administrative information about STREAMS activity.

Syntax

strinfo -m | -q

Description

The **strinfo** command displays information for debugging purposes about STREAMS, drivers and modules, or stream heads and the STREAMS run queue.

Flags

Item	Description
-m	Displays information on drivers and modules present in STREAMS.
-q	Displays informations on active stream heads, and on the run queue which holds the STREAMS module and driver service procedures.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To display information about STREAMS drivers and modules in use, enter:

```
strinfo -m
```

This produces a listing similar to the following:

```
Device: 'sad', dcookie 0xf, flags:0x4, str 0x19a69e8
Device: 'slog', dcookie 0x10, flags:0x4, str 0x19a6c18
Device: 'rs', dcookie 0x11, flags:0x2, str 0x19bcb00
Module: 'bufcall', flags:0x1, str 0x19a5c00
Module: 'ldterm', flags:0x0, str 0x19cc858
```

In this example dcookie indicates the major number, flags indicates the flags configuration, and str is the STREAMS table address.

2. To display information about active stream heads and the STREAMS run queue, enter:

```
strinfo -q
```

This produces a listing similar to the following:

```
Active Stream Heads
sth      sth_dev  sth_rq    sth_wq    sth_flag  rq->q_first
05a7ee00 00110001 05ad7000 05ad7074 00000818 00000000

STREAMS Service Queue
Queue 0x5ad7000 Flags 0x10
```

File

Item	Description
/usr/sbin/strinfo	Contains the strinfo command.

strings Command

Purpose

Finds the printable strings in a file.

Syntax

```
strings [ -a ] [ - ] [ -o ] [ -t Format ] [ -n Number ] [ -Number ] [ File ... ]
```

Description

The **strings** command looks for printable strings in a file. A string is any sequence of 4 or more printable characters that end with a new-line or a null character. The **strings** command is useful for identifying random object files.

Flags

Item	Description
-a or -	Searches the entire file, not just the data section, for printable strings. If this flag is omitted, the strings command only looks in the initialized data space of object files.
-n <i>Number</i>	Specifies a minimum string length other than the default of 4 characters. The maximum value of a string length is 4096. This flag is identical to the -Number flag.
-o	Lists each string preceded by its octal offset in the file. This flag is identical to the -t o flag.
-t <i>Format</i>	Lists each string preceded by its offset from the start of the file. The format is dependent on the character used as the <i>Format</i> variable. d Writes the offset in decimal. o Writes the offset in octal. x Writes the offset in hexadecimal. Note: When the -o and the -t <i>Format</i> flags are defined more than once on a command line, the last flag specified controls the behavior of the strings command.
-Number	Specifies a minimum string length other than the default of 4 characters. The maximum value of a string length is 4096. This flag is identical to the -n <i>Number</i> flag.
<i>File</i>	Binary or object file to be searched.

Exit Status

This command returns the following exit values:

Item	Description
0	Specifies that the command ran successfully.
>0	Specifies that an error occurred.

Examples

1. To search a file, enter:

```
strings strings
```

The **string** command displays:

```
@(#)56
1.17 com/cmd/scan/strings.c, cdmscan, bos320 5/7/92 10:21:20
Standard input
strings.cat
/usr/sbin/strings
Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
%7o
%7d
%7x
%7o
%7d
```

2. To search for strings at least 12 characters long, enter:

```
strings -12 strings
```

The **string** command displays:

```
1.17 com/cmd/scan/strings.c, cmdscan, bos320 5/7/92 10:21:20
Standard input
/usr/sbin/strings
Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
```

3. To search for strings at least 20 characters long and show the offset in hexadecimal, enter:

```
strings -t x -n 20 strings
```

The **string** command displays:

```
1017 1.17 com/cmd/scan/strings.c, cmdscan, bos320 5/7/92 10:21:20
108c Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
10d8 Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
1124 Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
1170 Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
11bc Usage: strings [-a | -] [-o] [-t format] [-n | -#] [file...]
```

strip Command

Purpose

Reduces the size of an Extended Common Object File Format (XCOFF) object file by removing information used by the binder and symbolic debug program.

Syntax

```
strip [ -V] [ -r [ -l] | -x [ -l] | -t | -H | -e | -E] [ -X {32|64|32_64}] [ -] File ...
```

Description

The **strip** command reduces the size of XCOFF object files. The **strip** command optionally removes the line number information, relocation information, the debug section, the typchk section, the comment section, file headers, and all or part of the symbol table from the XCOFF object files. Once you use this command, symbolic debugging of the file is difficult; therefore, you should normally use the **strip** command only on production modules that you have debugged and tested. Using the **strip** command reduces the storage overhead required by an object file.

For each object module, the **strip** command removes information as specified by the supplied options. For each archive file, the **strip** command removes the global symbol table from the archive.

You can restore a stripped symbol table to an archive or library file by using the **ar -s** command.

The **strip** command with no options removes the line number information, relocation information, symbol table, the debug section, and the typchk section, and the comment section.

Flags

Item	Description

-e	Sets the F_LOADONLY flag in the optional header of the object file. If the object file is placed in an archive, this flag indicates to the binder (ld command) that symbols in the object file should be ignored when linking with the archive.
-E	Resets (turns off) the F_LOADONLY bit in the optional header of the object file. (See -e flag).
-H	Removes the object file header, any optional header, and all section headers. Note: Symbol Table information is not removed.
-l	(Lowercase L) Strips the line number information from the object file.

Item	Description

-r	Removes all symbol table information except those entries for external and static symbols. Does not remove the relocation information. Also removes the debug and typchk sections. This option produces an object file that can still be used as input to the linkage editor (ld command).
-t	Removes most symbol table information but does not remove function symbols or line number information.
-V	Prints the version number of the strip command.
-x	Removes the symbol table information but does not remove static or external symbol information. The -x flag also removes relocation information, therefore linking to the file would not be possible.
-X <i>mode</i>	Specifies the type of object file strip should examine. The <i>mode</i> must be one of the following: 32 Processes only 32-bit object files 64 Processes only 64-bit object files 32_64 Processes both 32-bit and 64-bit object files The default is to process 32-bit object files (ignore 64-bit objects). The <i>mode</i> can also be set with the OBJECT_MODE environment variable. For example, OBJECT_MODE=64 causes strip to process any 64-bit objects and ignore 32-bit objects. The -X flag overrides the OBJECT_MODE variable.
--	(Double hyphen) Interprets all arguments following this flag as file names. This allows you to strip files whose names start with a hyphen.

Exit Status

This command returns the following exit values:

Item	Description

0	Successful completion.
>0	An error occurred.

Examples

1. To remove the symbol table and line number information from the **a.out** file, enter:

```
strip a.out
```

2. To remove the object file header of the **a.out** file, enter:

```
strip -H a.out
```

3. To remove both the 32-bit and 64-bit symbol tables from **lib.a**, enter:

```
strip -X 32_64 lib.a
```

Files

Item	Description
/usr/ccs/bin/strip	Contains the strip command.

stripnm Command

Purpose

Displays the symbol information of a specified object file.

Syntax

stripnm [**-x** | **-d**] [**-s**] [**-z**] *File*

Description

The **stripnm** command (when run without the **-s** flag) prints the symbol table of a specified object file to standard output. The file specified by the *File* parameter can be a single object file or an archive library of object files. If the file specified by the *File* parameter is an archive, a listing for each object file in the archive is produced. If the symbol table has been stripped from the object file, the **stripnm** command extracts symbol names from the traceback tables (even if the **-s** flag is not specified) and the loader section of the object file(s). If the traceback tables do not exist, an error message is displayed.

Each symbol name is preceeded by its address and one character representing the symbol type (similar to **nm** output). When used with **-z**, the output format is the same as it was before AIX 5.2, that is each symbol name is followed by its address (a series of blanks if the address is undefined) and the type of class and section type. The address field can be displayed as a decimal (the default value with **-z**, or when **-d** is used) or hexadecimal (the default value without **-z**, or if the **-x** flag is used).

Source file names are also collected and reported by the **stripnm** command. All the symbols following a source file name line belongs to the same source file, until the next source file name line is encountered. For stripped files, the source file name is reported as being the object file name.

When run using the **-s** flag, the **stripnm** command ignores the symbol table if present and always extracts routine names from the traceback tables and the loader section of the object file(s).

When no symbol table is present or the **-s** flag is used, the **stripnm** command also searches for glue code and pointer glue information. Both are sequences of instructions found in the text section of the object file.

The glue code for 32 bit applications is composed of the following sequences of instructions:

```
8182xxxx #   lwz r12,xxxx(r12) (xxxx is the TOC entry index)
90410014 #   stw r2,14(r1)
800c0000 #   lwz r0,0(r12)
```



```
804c0004 # lwz r2,4(r12)
7c0903a6 # mtctr r0
4e800420 # bctr
```

The loader section entry whose address matches the TOC entry pointed to by xxxx gives the function name for this sequence of glue code.

For 64 bit executables, the glue code sequences are as follows:

```
982xxxx # ld r12,xxxx(r2) (xxxx is the TOC entry index)
8410028 # std r2,28(r1)
80c0000 # ld r0,0(r12)
84c0008 # ld r2,8(r12)
c0903a6 # mtctr r0
e800420 # bctr
```

The pointer glue code for 32 bit applications is composed of the following sequence:

```
800b0000 # lwz r0,0(r11)
90410014 # stw r2,20(r1)
7c0903a6 # mtctr r0
804b0004 # lwz r2,4(r11)
816b0008 # lwz r11,8(r11)
4e80xx20 # bctr
```

For 64bit executables, the pointer glue code sequence is as follows:

```
e80b0000 # ld r0,0(r11)
f8410028 # std r2,20(r1)
7c0903a6 # mtctr r0
e84b0008 # ld r2,8(r11)
e96b0010 # ld r11,16(r11)
4e80xx20 # bctr
```

Pointer glue exists only in one copy and is always reported as symbol `._prtg1`.

The `stripnm` command searches the Text section from beginning to end for these sequences. If the command finds a sequence of instructions that matches, it is reported as glue code or pointer glue.

Source file symbols are generated artificially by **stripnm** for both glue code and pointer glue. For 32 bit executables, the source file is `glink.s` for all glue code entries, and `ptrgl.s`, for the pointer glue. For 64 bit executables, the source files are respectively `glink64.s` and `ptrgl_64.s`.

The **stripnm** command can also be used to search for symbol information in the `/unix` file. If the `/unix` file does not correspond to the currently running kernel, a warning message displays.

Flags

Item	Description
-d	Prints symbol address values in decimal format. This is the default with -z .
-s	Forces to ignore symbol table.
-x	Prints symbol address values in hexadecimal format. This is the default without -z .
-z	Uses the old format.

Examples

1. To list the symbols of the **a.out** object file, type:

```
stripnm a.out
```

2. To list the symbols address values, in decimal, from the **a.out** object file, type:

```
stripnm -d a.out
```

3. To list symbols from the object file from libc.a in the old format, but using hexadecimal addresses, type:

```
stripnm -xz libc.a
```

strload Command

Purpose

Loads and configures **Portable Streams Environment** (PSE).

Syntax

```
strload [ -u | -q ] [ -f File ] [ -d List ] [ -m List ]
```

Description

The **strload** command enables the system administrator to load and unload drivers and modules and to query the load status of PSE and its dependents.

By default, the **strload** command loads PSE according to the **/etc/pse.conf** file. The **-f** flag allows the administrator to use an alternate configuration file. The **-d** and **-m** flags are used to specify drivers and modules that are not present in the configuration files (such as when new drivers are being developed). The **-q** flag reports on the system load status (kernel existence) of the referenced drivers and modules.

Configuration File

The configuration file is a flat ASCII, line-oriented database. Comments are introduced by a # (pound sign), and continue until the end of the line. Blank lines are ignored. The form for each record is:

```
attributes filename [argument [node [minor ...] ] ]
```

Fields are separated by spaces, tabs, or both. A - (dash) can be specified as the field value, indicating that the default value is to be used. The fields are defined as follows:

Item	Description
attributes	Describes the extension to load. The acceptable values are:
d	Specifies a driver.
m	Specifies a module.
s	Creates the node as a standard (not cloned) device.
+	Specifies that the extension can be configured more than once. This value must be specified for all lines containing the extension file name.

Item	Description

filename	Specifies the object file containing the extension. If the command is issued with a "/" (slash) in the filename of the driver or module to be loaded, unloaded or queried, the strload command uses the value in the filename field explicitly. If there is no "/" in the filename entry, the strload command first looks for a copy of the driver or module in the current directory. If the driver or module is not in the current directory, strload looks for the driver or module in the /usr/lib/drivers/pse directory.

Note: It is recommended that the **strload** command be issued from the root directory (/). The **strload** command for load, unload, and query must always be issued from the same directory.

The kernel extension loader REQUIRES that the path names used be identical in load, unload and queries. This, coupled with the way the filename is determined by **strload**, could cause problems. Every byte in the path name used by the **strload** command must EXACTLY match every positionally corresponding byte in the path name used by the kernel extension loader because the kernel does a **strcmp()** on the filename when looking for matches. If the **strload** command is issued from a different directory to unload the module or driver, one of the following events occurs:

- If the **strload** command does not find a copy of the driver or module in the new current directory, **strload** attempts to unload the driver or module in the **/usr/lib/drivers/pse** directory. However, this path name may not be the same as the path name that the loader has logged for that driver or module. If the path name is not the same, the **strload** command fails.
- If the **strload** command finds another copy of the module or driver in the new current directory, then the path names are the same, and the loader correctly unloads the driver or module that was loaded. Thus, the **strload** command succeeds, but the results may not be as the user intended.

For example:

The following scenario (NOT recommended) causes "spx", also known as "A", to be unloaded. This is probably not the desired effect.

```
mkdir /tmp/foo /tmp/bar
cp /usr/lib/drivers/pse/spx /tmp/foo/A
cp /bin/ls /tmp/bar/A
cd /tmp/foo
strload -d A      # The loader knows the path and filename as
                  # "A" because "A" is found in the current
                  # directory

cd /tmp/bar
strload -q -d A   # Reports "yes" because there is "A" in the
                  # current directory. Note that the file "A"
                  # in /tmp/bar is NOT the same file "A" in
                  # /tmp/foo, but the loader does not care
                  # because it identifies the file by
                  # pathname.

strload -u -d A   # Unloads spx (also known as "A")!
```

The following is an error scenario:

```
mkdir /tmp/foo2 /tmp/bar2
cp /usr/lib/drivers/pse/spx /tmp/foo2/A
cd /tmp/foo2
strload -d A      # The loader knows the path and filename as
                  # "A" because "A" is found in the current
                  # directory.

cd /tmp/bar2
strload -q -d A   # Answers "no". There is no filename
                  # in /tmp/bar2 that matches "A", so strload
                  # prepends pathname "/usr/lib/drivers/pse" to
                  # "A". "/usr/lib/drivers/pse/A" is not found,
                  # so strload answers "no".

strload -u -d A   # Fails - "A" does not exist.
```

The following is an error scenario:

```
cd /usr/lib/drivers/pse
strload -d spx    # The loader knows the path and filename as
```

```

# "spx" because "spx" is found in the
# current directory.
cd /
strload -q -d spx # Answers "no". There is no filename in /
# that matches "spx", so strload prepends
# the pathname "/usr/lib/drivers/pse" to
# "spx". "/usr/lib/drivers/pse/spx" is found
# since it exists, so strload gives
# "/usr/lib/drivers/pse/spx" to the loader.
# The strcmp() fails since
# "/usr/lib/drivers/pse/spx" and "spx" do
# not match exactly.
strload -u -d spx # Fails - "spx" does not exist.

```

Item	Description

argument	Has no meaning for the strload command. This field is optional. It is passed to the extension when its configuration routine is called. Its interpretation is specific to that extension. The default argument is the value of the filename field.
node	Specifies the name of the node to create. This field is optional. It applies only to drivers and is used as the created node name when the driver is loaded. By default, the created node is /dev/filename .
minor	Specifies additional, non-clone nodes to create for this driver. This field is optional. The node names are created by appending the minor number to the cloned driver node name. No more than five minor numbers can be given (from 0 to 4), and a node is created for each one.

The **-d** and **-m** flags cause the configuration file to be ignored, unless it is explicitly named on the command line, as follows:

```
strload -f /tmp/my.conf -d newdriver
```

Note: The **-d** and **-m** flags do not override the configuration file. That is, if driver **dgb** is loaded by using the configuration file, the **-d** flag will attempt to reload it but will fail. The configuration file is processed before the **-d** and **-m** flags.

The *List* variable for the **-d** and **-m** flags is a comma-separated list of file names, each of which contains a single PSE driver or module. The configuration process proceeds as if a line of one of the following forms was found in the configuration file:

```
d filename
```

```
m filename
```

Flags

Item	Description

-d List	Lists PSE device drivers to load or unload. The <i>List</i> variable specifies a comma-separated list of driver object names.
-f File	Configures PSE according to the configuration information contained in the file indicated by the <i>File</i> variable. The default configuration file is /etc/pse.conf .
-m List	Lists PSE modules to load or unload. The <i>List</i> variable specifies a comma-separated list of module object names.
-q	Reports load status of extensions.
-u	Unloads extensions.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. Entering the following command loads PSE (if not already loaded), the dgb and ssb drivers from the `/usr/lib/drivers/pse/` directory, and the aoot module from the current directory, but does not use the configuration file:

```
root# strload -d dgb,ssb -m ./aoot
```

2. To unload the aoot module only, enter:

```
root# strload -u -m ./aoot
```

3. Entering the following command asks if the spx driver exists:

```
root# strload -q -d
spx
```

and produces the following output if not:

```
spx
: no
```

4. The following is an example configuration file:

```
#example configuration file
d      dgb                                #line 1
d      mux      -      -      0         #line 2
ds     foo                                #line 3
d+     xtiso    tcp    /dev/xti/tcp      #line 4
d+     xtiso    udp    /dev/xti/udp      #line 5
m      aoot                                #line 6
```

Line 1 loads the dgb driver extension as a cloned device named `/dev/dgb`. The argument passed to the dgb configuration routine is dgb.

Line 2 loads the mux driver extension as a cloned device named `/dev/mux` and also creates a standard device name `/dev/mux0` with a minor number of 0 (zero). (No more than five device names can be created with minor numbers from 0 to 4.)

Line 3 loads the foo driver extension as a standard device (not cloned) named `/dev/foo`. The minor number is 0.

Lines 4 and 5 load the xtiso driver extension, and configure it twice: once as tcp and once as udp. The clone nodes created are `/dev/xti/tcp` and `/dev/xti/udp`. The configuration routine of xtiso is called twice: once with the argument tcp, and once with udp.

Line 6 loads the aoot module extension. No node is created, and the configuration routine is passed the value aoot.

5. To load the streams **dlpi** driver, enter:

```
strload -f /etc/dlpi.conf
```

Files

Item	Description
<code>/usr/lib/drivers/pse/*</code>	Contains PSE kernel extensions.

Item	Description
/etc/pse.conf	Default PSE configuration file.
/usr/sbin/strload	Contains the strload command.

strreset Command

Purpose

Resets a stream.

Syntax

strreset [**-M** *Major*] [**-m** *Minor*]

Description

The **strreset** command resets an open stream by generating an M_FLUSH message to the stream head. You use it mainly to reset blocked streams. When it is impossible to reopen the stream, issue an I_FLUSH ioctl(), or equivalent command. This situation may happen with a process sleeping in a module's close routine, when signals can not be sent to the process (a zombie process exiting, for example).

Flags

Item	Description
-M <i>Major</i>	Specifies the major number for the special file associated with the stream to be reset.
-m <i>Minor</i>	Specifies the minor number for the special file associated with the stream to be reset.

Exit Status

This command returns the following exit values:

Ite	Description
m	
0	Successful completion.
>0	An error occurred.

Security

Access Control: You must have root authority to run this command.

Auditing Events: N/A

Files

Item	Description
/usr/sbin/strreset	Contains the strreset command.

str tune Command

Purpose

This command has several related functions:

- Get or set the streams tunable parameters.
- Define the objects to trace using the component trace.
- List the tunable values of the stream modules.
- List the tunable values of the active queues.

Syntax

str tune {**-n** *name* | **-q** *addr*} **-o** *tunable_name*[=*value*] **-o** *tunable_name*[=*value*] ...

str tune [**-n** *name* | **-q** *addr* [**-a**]] **-o** *trclevel*[=*value*]

str tune [**-M**]

str tune [**-Q**]

str tune [**-f** *tunefile*]

Description

There are no restrictions on the use of this command when it is used to display or list values, but when using this command to modify tunable values or to define objects to trace, you must have root authority.

Flags

-n *name*

Defines a stream module name or a device name.

-q *addr*

Defines an active queue address.

If the command sets tunables, it modifies the queue pair, or the only queue, depending on the synchronization level of the queue. If the synchronization level is not SQLVL_QUEUE, the synchronization level is also propagated to all queue pairs.

-o *tunable_name*

Defines the name of the tunable parameter. Possible values are:

- **hiwat**, which defines the high water mark for the flow control on a queue
- **lowat**, which defines the low water mark for the flow control on a queue
- **minpsz**, which defines the minimum packet size
- **maxpsz**, which defines the maximum packet size. A value of -1 indicates an infinite packet size.

The **str tune** command can initialize several tunables by listing the **-o** option several times.

value

If no new value is given, the command displays the value of the tunable. Only a user with root authority can modify a tunable parameter value.

-n *name*

Defines a stream module name. If the **-n** or **-q** flag is not present in the command, the command will display or modify the global variable containing the pse global trace level (**pse_trclevel**).

-q *addr*

Defines an active queue address. If the **-n** or **-q** flag is not present in the command, the command will display or modify the global variable containing the pse global trace level (**pse_trclevel**).

If the command sets the trace level, it modifies the queue pair, or the only queue, depending on the synchronization level of the queue. If the synchronization level is not SQLVL_QUEUE, the synchronization level is also propagated to all queue pairs.

-o *trclevel*

Displays or modifies the trace level. The **-o** flag cannot be listed more than once.

value

If no new value is given, the command displays the value of the tunable. Only a user with root authority can modify a tunable parameter value.

-a

Use this flag to force the **strtune** command to propagate the new value to all queues in the stream (from stream head to driver). If the synchronization level is not SQLVL_QUEUE, the synchronization level is also propagated to all queue pairs.

-M

Displays the name, idname, and associated tunable parameter (**minpsz**, **maxpsz**, **lowat**, **hiwat**, **trclevel**) values for each module.

-Q

Displays the name, idname, and associated tunable parameter (**minpsz**, **maxpsz**, **lowat**, **hiwat**, **trclevel**) values for each active queue.

-f *tunefile*

The *tunefile* variable holds the filepath to the file that contains the tunable parameter settings. Each line of the *tunefile* file is managed as one command; if there are any modification commands in the *tunefile*, the user must have root authority for those modifications to be implemented.

Exit Status

0

Successful completion.

>0

An error occurred.

Examples

1. To display the **hiwat** tunable value of **ldterm** module:

```
strtune -n ldterm -o hiwat
```

2. With root authority, to set the value of **hiwat** for the **ldterm** module to 8192:

```
strtune -n ldterm -o hiwat=8192
```

3. To run the following lines:

```
-n udp6 -o lowat=256  
-n dlpi -o hiwat=4096 -o lowat=128 -o minpsz=128
```

that are listed in the **/tmp/ff** file:

```
strtune -f /tmp/ff
```

This will result in the following commands being run:

```
strtune -n udp6 -o lowat=256  
strtune -n dlpi -o hiwat=4096 -o lowat=128 -o minpsz=128
```

File

src/bos/usr/sbin/strtune/strtune.c

Contains the **strtune** command.

struct Command

Purpose

Translates a FORTRAN program into a RATFOR program.

Syntax

struct [-s] [-i] [-a] [-b] [-n] [-t*Number*] [-c*Number*] [-e*Number*] [*File*]

Description

The **struct** command translates the FORTRAN program specified by *File* (standard input default) into a RATFOR program. Wherever possible, RATFOR control constructs replace the original FORTRAN. Statement numbers appear only where still necessary. Cosmetic changes are made, including changing Hollerith strings into quoted strings and relational operators into symbols (for example, **.GT.** into **>**). The output is appropriately indented.

The **struct** command knows FORTRAN 66 syntax, but not full FORTRAN 77. If an input FORTRAN program contains identifiers that are reserved words in RATFOR, the structured version of the program will not be a valid RATFOR program. The labels generated cannot go above 32767. If you get a **goto** statement without a target, try using the **-e** flag.

Flags

Item	Description
-a	Turn sequences of else-if statements into a non-RATFOR switch of the form: <pre>switch { case pred1: code case pred2: code case pred3: code default: code }</pre> The case predicates are tested in order. The code appropriate to only one case is executed. This generalized form of switch statement does not occur in RATFOR.
-b	Generates goto statements instead of multilevel break statements.
-c <i>Number</i>	Increments successive labels in the output program by the nonzero integer <i>Number</i> . The default is 1. Do not insert a space between -c and <i>Number</i> .
-e <i>Number</i>	If <i>Number</i> is 0 (default), places code within a loop only if it can lead to an iteration of the loop. Do not insert a space between -e and <i>Number</i> .
-i	Do not turn computed goto statements into switches. (RATFOR does not turn switches back into computed goto statements.)
-n	Generates goto statements instead of multilevel next statements.
-s	Input is accepted in standard format. Comments are specified by a c , C , or * in column 1, and continuation lines are specified by a nonzero, nonblank character in column 6. Input is in the form accepted by the f77 command.
-t <i>Number</i>	Makes the nonzero integer <i>Number</i> the lowest valued label in the output program. The default is 10. Do not insert a space between -t and <i>Number</i> .

If *Number* is nonzero, admits small code segments to a loop if otherwise the loop would have exits to several places including the segment, and the segment can be reached only from the loop. In this case, small is close to, but not equal to, the number of statements in the code segment. Values of *Number* under 10 are suggested.

Examples

To translate the `test.f` FORTRAN program into the `newtest.ratfor` RATFOR program, enter:

```
struct -s -i -n -t2 test.f > newtest.ratfor
```

Files

Item	Description
<code>/tmp/struct*</code>	Temporary files used during processing of the struct command.
<code>/usr/lib/struct/structure</code>	File that handles processing for the struct command.
<code>/usr/lib/struct/beautify</code>	File that handles processing for the struct command.
<code>/usr/ucb/struct</code>	Contains the struct command.

sttinnet Method

Purpose

Enables the inet instance.

Syntax

sttinnet [**-l** *Interface ...*]

Description

The **sttinnet** method enables the inet instance by calling the **ifconfig** command and sets the status flag of the inet instance to AVAILABLE.

Note: The **sttinnet** method is a programming tool and should not be executed from the command line.

Flags

Item	Description
-l <i>Interface ...</i>	Specifies which specific interface to enable. If no interfaces are specified, then all configured interfaces are started.

Examples

The following method enables the inet instance:

```
sttinnet -l tr0 -l tr1
```

stty-cxma Command

Purpose

Sets and reports the terminal options for a TTY configuration of the 128-port asynchronous subsystem.

Syntax

stty-cxma [**-a**] [**-g**] [*Option(s)*] [*ttyName*]

Description

If no flags or options are specified, the **stty-cxma** command reports all 128-port special driver settings and modem signals, as well as all standard parameters reported by the **stty** command for the tty device that is the current standard input.

The *ttyName* parameter can be specified to set or report options for a tty device for other than the standard input. The *ttyName* parameter can be a simple tty name, such as **tty0**, or can be prefixed by **/dev/**, such as **/dev/tty0**. This option may be used on a modem control line when no carrier is present.

Further options can be specified to change flow control settings, set transparent print options, force modem control lines, and display all tty settings. Unrecognized options are passed to the **stty** command for interpretation.

Flags

Item	Description

-a	Writes all the unique 128-port settings as well as all the standard tty settings reported by stty -a to standard output.

Item	Description

-g	Writes option settings to standard output in a form usable by another stty command.

Options

The following options specify transient actions to be performed immediately:

Item	Description
break	Sends a 250 MS break signal out on the tty line.
flush	Discards tty input and output immediately.
flushin	Discards tty input only.
flushout	Discards tty output only.

The actions specified by the following options are in effect until the device is closed. The next time the device is opened, default values are used.

Item	Description
dtr	Raises the DTR modem control line, unless DTR hardware flow control is selected.
-dtr	Drops the DTR modem control line, unless DTR hardware flow control is selected.
rts	Raises the RTS modem control line, unless RTS hardware flow control is selected.
-rts	Drops the RTS modem control line, unless RTS hardware flow control is selected.
startin	Releases flows control to resume stopped input.
startout	Restarts stopped output exactly as if an XON character was received.
stopin	Activates flow control to stop input.
stopout	Stops output exactly as if an XOFF character was received.

Item	Description
2200flow	Enables 2200 style flow control on the port. The 2200 terminals support an attached printer and use the following four flow control characters: 0xF8 terminal XON 0xF9 printer XON 0xFA terminal XOFF 0xFB printer XOFF
-2200flow	Disables 2200 style flow control on the port.
2200print	Runs flow control for the terminal and flow control for the transparent print device (as set by the 2200flow option) independently.
-2200print	Runs terminal and printer flow control (as set by the 2200flow option) together. So if either the terminal or the printer XOFF character is received, all output is paused until the matching XON character is received.
altpin	Switches the location of the DSR and DCD inputs on the modular connector, so that DCD is available when using an 8-pin RJ45 connector instead of the 10-pin RJ45 connector.
-altpin	Restores the availability of DSR when using the 10-pin RJ45 connector.
aixon	Enables auxiliary flow control, so that two unique characters are used for XON and XOFF. If both XOFF characters are received, transmission will not resume until both XON characters are received.
-aixon	Disables auxiliary flow control.
astartc c	Sets auxiliary XON flow control character. The character may be given as a decimal, octal, or hexadecimal number.
astopc c	Sets auxiliary XOFF flow control character. The character may be given as a decimal, octal, or hexadecimal number.
bufsize n	Sets the driver's estimate of the size of the transparent printer's input buffer. After a period of inactivity, the driver bursts this many characters to the transparent printer before reducing to the maximum CPS rate specified by the maxcps option rate selected above. The default value is 100 characters.
ctspace	Enables CTS hardware output flow control, so local transmission pauses when CTS drops.
-ctspace	Disables CTS hardware output flow control.
dcdpace	Enables DCD hardware output flow control, so local transmission pauses when DCD drops.
-dcdpace	Disables DCD hardware output flow control.
dsrpace	Enables DSR hardware output flow control, so local transmission pauses when DSR drops.
-dsrpace	Disables DSR hardware output flow control.
dtrpace	Enables DTR hardware input flow control, so DTR drops to pause remote transmission.
-dtrpace	Disables DTR hardware input flow control.

Item	Description

edelay <i>n</i>	Sets the rate at which the 128-port asynchronous adapter wakes up the driver on input. The adapter wakes the driver every <i>n</i> milliseconds. The default value is 100 milliseconds.

fastbaud	Alters the baud rate table, so 50 baud becomes 57600 baud.

-fastbaud	Restores the baud rate table, so 57500 baud becomes 50 baud.

Item	Description

fastcook	Performs cooked output processing on the 128-port asynchronous adapter to reduce host CPU usage and increase raw mode input performance.

-fastcook	Disables cooked output processing.

forcedcd	Disables carrier sense, so the tty may be opened and used even when the carrier is not present.

-forcedcd	Reenables carrier sense.

maxchar <i>n</i>	Sets the maximum number of transparent print characters the driver places in the output queue. Reducing this number increases system overhead; increasing this number delays operator keystroke echo times when the transparent printer is in use. The default value is 50 characters.

maxcps <i>n</i>	Sets the maximum CPS (characters per second) rate at which characters are output to the transparent print device. The rate chosen should be just below the average print speed. If the number is too low, printer speed is reduced. If the number is too high, the printer resorts to flow control, and user entry on the CRT is impaired accordingly. The default value is 100 CPS.

offstr <i>s</i>	Sets the CRT escape sequence to turn transparent print off. An arbitrary octal character <i>xxx</i> may be given as \xxx.

onstr <i>s</i>	Sets the CRT escape sequence to turn transparent print on. An arbitrary octal character <i>xxx</i> may be given as \xxx.

rtspace	Enables RTS hardware input flow control, so RTS drops to pause remote transmission.

-rtspace	Disables RTS hardware input flow control.

startc <i>c</i>	Sets the XON flow control character. The character may be given as a decimal, octal, or hexadecimal number.

stopc <i>c</i>	Sets the XOFF flow control character. The character may be given as a decimal, octal, or hexadecimal number.

term <i>t</i>	Sets the transparent printer on and off strings to values specified in the internal default table. Internal defaults are used for the following terminals: adm31 , ansi , dg200 , dg210 , hz1500 , mc5 , microterm , multiterm , pcterm , tvi , vp-a2 , vp-60 , vt52 , vt100 , vt220 , wyse30 , wyse50 , wyse60 , or wyse75 . If the terminal type is not found in the internal default table, the transparent print on and off strings are set to the values specified by the po and pf attributes in the termcap file.

Examples

1. To display all the unique 128-port settings as well as all the standard tty settings for a tty port configured on a 128-port asynchronous controller as /dev/tty0, enter:

```
stty-cxma -a tty0
```

2. To make DCD available when using an 8-pin RJ45 connector for a tty port configured on a 128-port asynchronous controller as /dev/tty3, enter:

```
stty-cxma altpin tty3
```

This command interchanges the location of the DSR and DCD inputs on the modular connector.

Files

Item	Description
<code>/usr/sbin/tty/stty-cxma</code>	Contains the stty-cxma command.

stty Command

Purpose

Sets, resets, and reports workstation operating parameters.

Syntax

```
stty [ -a ] [ -g ] [ Options ]
```

Description

The **stty** command sets certain I/O options for the device that is the current standard input. This command writes output to the device that is the current standard output.

This version of the operating system uses the standard X/Open Portability Guide Issue 4 interface to control the terminals, maintaining a compatibility with POSIX and BSD interfaces. The **stty** command supports both POSIX and BSD compliant options, but the usage of POSIX options is strongly recommended. A list of **obsolete BSD options**, with the corresponding POSIX options, is also provided.

When you redirect standard input from a tty device by typing:

```
stty -a </dev/ttyx
```

the **stty** command (POSIX) will hang while waiting for the **open()** of that tty until the RS-232 carrier detect signal has been asserted. Exceptions to this rule occur if the **cllocal** or **forcedcd** (128-port only) option is set.

Flags

Item	Description

-a	Writes the current state of all option settings to standard output.
-g	Writes option settings to standard output in a form usable by another stty command.

Options

The **stty** command supports following categories of options:

- [Control Modes](#)
- [Input Modes](#)
- [Output Modes](#)
- [Local Modes](#)
- [Hardware Flow Control Modes](#)
- [Control Character Assignments](#)

- [Combination Modes](#)
- [Window Size](#)

Control Modes

Control Modes	Description
clocal	Assumes a line without modem control.
-clocal	Assumes a line with modem control.
cread	Enables the receiver.
-cread	Disables the receiver.
cstopb	Selects 2 stop bits per character.
-cstopb	Selects 1 stop bit per character.
cs5, cs6, cs7, cs8	Selects character size.
hup, hupcl	Hangs up dial-up connection on the last close.
-hup, -hupcl	Does not hang up dial-up connection on the last close.
parenb	Enables parity generation and detection.
-parenb	Disables parity generation and detection.
parodd	Selects odd parity.
-parodd	Selects even parity.
0	Hangs up phone line immediately.
speed	Sets the workstation input and output speeds to the specified <i>speed</i> number of bits per second. All speeds are not supported by all hardware interfaces. Possible values for <i>speed</i> are: 50, 75, 110, 134, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 19.2, 38400, 38.4, exta, and extb. Note: exta, 19200, and 19.2 are synonyms; extb, 38400, and 38.4 are synonyms.
ispeed speed	Sets the workstation input speed to the specified <i>speed</i> number of bits per second. All speeds are not supported by all hardware interfaces, and all hardware interfaces do not support this option. Possible values for <i>speed</i> are the same as for the <i>speed</i> option.
ospeed speed	Sets the workstation output speed to the specified <i>speed</i> number of bits per second. All speeds are not supported by all hardware interfaces, and all hardware interfaces do not support this option. Possible values for <i>speed</i> are the same as for the <i>speed</i> option.

Input Modes

Input Modes	Description
brkint	Signals INTR on break.
-brkint	Does not signal INTR on break.
icrnl	Maps CR to NL on input.
-icrnl	Does not map CR to NL on input.
ignbrk	Ignores BREAK on input.
-ignbrk	Does not ignore BREAK on input.

Input Modes

	Description
igncr	Ignores CR on input.
-igncr	Does not ignore CR on input.
ignpar	Ignores parity errors.
-ignpar	Does not ignore parity errors.
inlcr	Maps NL to CR on input.
-inlcr	Does not map NL to CR on input.
inpck	Enables parity checking.
-inpck	Disables parity checking.
istrip	Strips input characters to 7 bits.
-istrip	Does not strip input characters to 7 bits.
iuc lc	Maps uppercase alphabetic characters to lowercase.
-iuc lc	Does not map uppercase alphabetic characters to lowercase.
ixany	Allows any character to restart output.
-ixany	Allows only the START (the Ctrl-Q key sequence) to restart output.
ixoff	Sends START/STOP characters when the input queue is nearly empty/full.
-ixoff	Does not send START/STOP characters.
ixon	Enables START/STOP output control. Once START/STOP output control has been enabled, you can pause output to the workstation by pressing the Ctrl-S key sequence and resume output by pressing the Ctrl-Q key sequence.
-ixon	Disables START/STOP output control.
imaxbel	Echoes the BEL character and discards the last input character if input overflows.
-imaxbel	Discards all input if input overflows.
parmrk	Marks parity errors.
-parmrk	Does not mark parity errors.

Output Modes

Output Modes	Description
bs0, bs1	Selects style of delay for backspaces (bs0 signifies no delay).
cr0, cr1, cr2, cr3	Selects style of delay for CR characters (cr0 signifies no delay).
ff0, ff1	Selects style of delay for form feeds (ff0 signifies no delay).
nl0, nl1	Selects style of delay for NL characters (nl0 signifies no delay).
ofill	Uses fill characters for delays.
-ofill	Uses timing for delays.
ocrnl	Maps CR characters to NL characters.
-ocrnl	Does not map CR characters to NL characters.
olcuc	Maps lowercase alphabetic characters to uppercase on output.

Output Modes

	Description
-olcuc	Does not map lowercase alphabetic characters to uppercase on output.
onlcr	Maps NL characters to CR-NL characters.
-onlcr	Does not map NL characters to CR-NL characters.
onlret	On the terminal, NL performs the CR function.
-onlret	On the terminal, NL does not perform the CR function.
onocr	Does not output CR characters at column zero.
-onocr	Outputs CR characters at column zero.
opost	Processes output.
-opost	Does not process output; that is, ignores all other output options.
ofdel	Uses DEL characters for fill characters.
-ofdel	Uses NUL characters for fill characters.
tab0, tab1, tab2	Selects style of delay for horizontal tabs (tab0 signifies no delay).
tab3	Expands tab character to variable number of spaces.
vt0, vt1	Selects style of delay for vertical tabs (vt0 signifies no delay).

Local Modes

Local Modes

	Description
echo	Echoes every character typed.
-echo	Does not echo characters.
echoctl	Echoes control characters as ^X (Ctrl-X), where X is the character given by adding 100 octal to the code of the control character.
-echoctl	Does not echo control characters as ^X (Ctrl-X).
echoe	Echoes the ERASE character as the "backspace space backspace" string. Note: This mode does not keep track of column position, so you can get unexpected results when erasing such things as tabs and escape sequences.
-echoe	Does not echo the ERASE character, just backspace.
echok	Echoes a NL character after a KILL character.
-echok	Does not echo a NL character after a KILL character.
echoke	Echoes the KILL character by erasing each character on the output line.
-echoke	Just echoes the KILL character.
echonl	Echoes the NL character.
-echonl	Does not echo the NL character.
echopr	Echoes erased characters backwards with / (slash) and \ (backslash).
-echopr	Does not echo erased characters backwards with / (slash) and \ (backslash).
icanon	Enables canonical input (canonical input allows input-line editing with the ERASE and KILL characters). See the discussion about canonical mode input in Line Discipline Module (ldterm) in <i>Communications Programming Concepts</i> .

Local Modes	Description
-icanon	Disables canonical input.
ixten	Specifies that implementation-defined functions shall be recognized from the input data. Recognition of the following control characters requires ixten to be set: eol2 , dsusp , reprint , discard , werase , lnext . The functions associated with these modes also require ixten to be set: imaxbel , echoke , echoprt , and echoctl .
-ixten	Specifies that implementation-defined functions shall not be recognized from the input data.
isig	Enables the checking of characters against the special control characters INTR, SUSP and QUIT.
-isig	Disables the checking of characters against the special control characters INTR, SUSP and QUIT.
noflsh	Does not clear buffers after INTR, SUSP, or QUIT control characters.
-noflsh	Clears buffers after INTR, SUSP, or QUIT control characters.
pending	Causes any input that is pending after a switch from raw to canonical mode to be re-input the next time a read operation becomes pending or the next time input arrives. Pending is an internal state bit.
-pending	No text is pending.
tostop	Signals SIGTOU for background output.
-tostop	Does not signal SIGTOU for background output.
xcase	Echoes uppercase characters on input, and displays uppercase characters on output with a preceding \ (backslash).
-xcase	Does not echo uppercase characters on input.

Hardware Flow Control Modes

These options are extensions to the X/Open Portability Guide Issue 4 standard.

Item	Description
cdxon	Enables CD hardware flow control mode on output.
-cdxon	Disables CD hardware flow control mode on output.
ctsxon	Enables CTS hardware flow control mode on output.
-ctsxon	Disables CTS hardware flow control mode on output.
dtrxoff	Enables DTR hardware flow control mode on input.
-dtrxoff	Disables DTR hardware flow control mode on input.
rtsxoff	Enables RTS hardware flow control mode on input.
-rtsxoff	Disables RTS hardware flow control mode on input.

Control Assignments

To assign a control character to a character string, type:

```
stty Control String
```

where the *Control* parameter may be the intr, quit, erase, kill, eof, eol, eol2, start, stop, susp, dsusp, reprint, discard, werase, lnext, min, or time character. (Use the min and time characters with the **-icanon** option.)

Note: The values for min and time are interpreted as integer values, not as character values.

The *String* parameter may be any single character such as c. An example of this control assignment is:

```
stty stop c
```

Another way of assigning control characters is to enter a character sequence composed of a \[^] (backslash, caret) followed by a single character. If the single character after the ^ (caret) is one of the characters listed in the ^c (caret c) column of the following table, the corresponding control character value will be set. For example, to assign the DEL control character by using the ? (question mark) character, type the string \[^]? (backslash, caret, question mark), as in:

```
stty erase \^?
```

caret Control Characters in stty	
[^] c	Value
a, A	<SOH>
b, B	<STX>
c, C	<ETX>
d, D	<EOT>
e, E	<ENQ>
f, F	<ACK>
g, G	<BEL>
h, H	<BS>
i, I	<HT>
j, J	<LF>
k, K	<VT>
l, L	<FF>
m, M	<CR>
n, N	<SO>
o, O	<SI>
p, P	<DLE>
q, Q	<DC1>
r, R	<DC2>
s, S	<DC3>
t, T	<DC4>
u, U	<NAK>
v, V	<SYN>
w, W	<ETB>
x, X	<CAN>
y, Y	
z, Z	<SUB>

caret Control Characters in stty (<i>continued</i>)	
^c	Value
[	<ESC>
\	<FS>
]	<GS>
^	<RS>
_	<US>
?	<DEL>
@	<NUL>

Combination Modes

cooked

See the **-raw** option.

ek

Sets ERASE and KILL characters to the Ctrl-H and Ctrl-U key sequences, respectively.

evenp

Enables **parenb** and **cs7**.

-evenp

Disables **parenb** and sets **cs8**.

lcase, LCASE

Sets **xcase**, **iuc lc**, and **olc uc**. Used for workstations with uppercase characters only.

-lcase, -LCASE

Sets **-xcase**, **-iuc lc**, and **-olc uc**.

nl

Sets **-icrnl** and **-onlcr**.

-nl

Sets **icrnl**, **onlcr**, **-inlcr**, **-igncr**, **-ocrnl**, and **-onlret**.

oddp

Enables **parenb**, **cs7**, and **parodd**.

-oddp

Disables **parenb** and sets **cs8**.

parity

See the **evenp** option.

-parity

See the **-evenp** option.

sane

Resets parameters to reasonable values.

raw

Allows raw mode input (no input processing, such as erase, kill, or interrupt); parity bit passed back.

-raw

Allows canonical input mode.

tabs

Preserves tabs.

-tabs, tab3

Replaces tabs with spaces when printing.

Window size

Description

cols *n*, columns *n*

The terminal (window) size is recorded as having *n* columns.

rows *n*

The terminal (window) size is recorded as having *n* rows.

size

Prints the terminal (window) sizes to standard output (first rows and then columns).

Obsolete Options

The following BSD options are supported by the **stty** command. For each of them, the recommended POSIX option is given.

Item	Description
all	Use the stty -a command to display all current settings.
crt	Use the sane option to reset parameters to reasonable values.
crtbs	Use the -echoe option.
crterase	Use the echoe option.
-crterase	Use the -echoe option.
crtkill	Use the echoke option.
-crtkill	Use the echok and -echoke options.
ctlecho	Use the echoctl option.
-ctlecho	Use the -echoctl option.
decctlq	Use the -ixany option.
-decctlq	Use the ixany option.
even	Use the evenp option.
-even	Use the -evenp option.
everything	Use the stty -a command to display all current settings.
litout	Use the -opost option.
-litout	Use the opost option.
odd	Use the oddp option.
-odd	Use the -oddp option.
pass8	Use the -istrip option.
-pass8	Use the istrip option.
prterase	Use the echopr option.
speed	Use the stty command to display current settings.
tandem	Use the ixoff option.
-tandem	Use the -ixoff option.

Examples

1. To display a short listing of your workstation configuration, type:

```
stty
```

This lists settings that differ from the defaults.

2. To display a full listing of your workstation configuration, type:

```
stty -a
```

3. To enable a key sequence that stops listings from scrolling off the screen, type:

```
stty ixon ixany
```

This sets **ixon** mode, which lets you stop runaway listing by pressing the Ctrl-S key sequence. The **ixany** flag allows you to resume the listing by pressing any key. The normal workstation configuration includes the **ixon** and **ixany** flags, which allows you to stop a listing with the Ctrl-S key sequence that only the Ctrl-Q key sequence will restart.

4. To reset the configuration after it has been messed up, type:

Ctrl-J stty sane Ctrl-J

Press the Ctrl-J key sequence before and after the command instead of the Enter key. The system usually recognizes the Ctrl-J key sequence when the parameters that control Enter key processing are messed up.

Sometimes the information displayed on the screen may look strange, or the system will not respond when you press the Enter key. This can happen when you use the **stty** command with parameters that are incompatible or that do things you don't understand. It can also happen when a screen-oriented application ends abnormally and does not have a chance to reset the workstation configuration.

Entering the **stty sane** command sets a reasonable configuration, but it may differ slightly from your normal configuration.

5. To save and restore the terminal's configuration:

```
OLDCONFIG=`stty -g`      # save configuration
stty -echo               # do not display password
echo "Enter password: \c"
read PASSWD              # get the password
stty $OLDCONFIG          # restore configuration
```

This command saves the workstation's configuration, turns off echoing, reads a password, and restores the original configuration.

Entering the **stty -echo** command turns off echoing, which means that the password does not appear on the screen when you type it at the keyboard. This action has nothing to do with the **echo** command, which displays a message on the screen.

File

Item	Description
<code>/usr/bin/stty</code>	Contains the stty command.

style Command

Purpose

Analyzes surface characteristics of a document.

Syntax

style [-a] [-e] [-lNumber] [-ml] [-mm] [-p] [-P] [-rNumber] *File ...*

Description

The **style** command analyzes the surface characteristics of the writing style of an English-language document. It reports on readability, sentence length and structure, word length and usage, verb type, and sentence openers. Because the **style** command runs the **deroff** command before looking at the text, header files that contain appropriate formatting information should be included as part of the input.

Note: The use of nonstandard formatting macros may cause incorrect sentence breaks.

Flags

Item	Description
-a	Prints all sentences with their length and readability index.
-e	Prints all sentences that begin with an expletive such as "There are".
-lNumber	Prints all sentences longer than the number of words specified by the parameter <i>Number</i> .

Item	Description
-ml	Causes the deroff command to skip lists; use -ml if a document contains many lists of sentence fragments.
-mm	Overrides the default ms macro package.
-p	Prints all sentences that contain a passive verb.
-P	Prints parts of speech of the words in the document.
-rNumber	Prints all sentences whose readability index is greater than <i>Number</i> .

su Command

Purpose

Changes the user ID associated with a session.

Syntax

su [-] [*Name* [*Argument ...*]]

Description

The **su** command changes user credentials to those of the root user or to the user specified by the *Name* parameter, and initiates a new session. The user name might include a Distributed Computing Environment (DCE) cell specification.

Note: The root user is not required to satisfy the DCE authentication when switching to a DCE user. In this case, the user's DCE credentials are not required.

Any arguments, such as flags or parameters, that are specified by the *Arguments* parameter must relate to the login shell defined for the user specified by the *Name* parameter. These arguments are passed to the specified user's login shell. For example, if the login shell for user Fred is **/usr/bin/csh**, you can include any of the flags for the **csh** command, such as the **-f** flag. When the **su** command runs, it passes the **-f** flag to the **csh** command. When the **csh** command runs, the **-f** flag omits the **.cshrc** startup script.

Note: If the *domainlessgroups* attribute is set in the **/etc/secvars.cfg** file and if the user belongs to the Lightweight Directory Access Protocol (LDAP) domain or files domain, all the group IDs are fetched from the LDAP domain and the files domain.

The following functions are performed by the **su** command:

Item	Description
account checking	Validates the user account to be certain it exists, that it is enabled for the su command, that the current user is in a group permitted to switch to this account with the su command, and that it can be used from the current controlling terminal.
user authentication	Validates the user's identity, using the system-defined primary authentication methods for the user. If a password has expired, the user must supply a new password.
credentials establishment	Establishes initial user credentials, using the values in the user database. These credentials define the user's access rights and accountability on the system.

Item	Description
session initiation	If the - flag is specified, the su command initializes the user environment from the values in the user database and the /etc/environment file. When the - flag is not used, the su command does not change the directory.

These functions are performed in the sequence shown. If one function is unsuccessful, the succeeding functions are not done. Refer to the **ckuseracct**, **ckuserID**, **authenticate**, **setpcred**, and **setpenv** subroutines for the semantics of these functions.

To restore the previous session, type **exit** or press the Ctrl-D key sequence. This action ends the shell called by the **su** command and returns you to the previous shell, user ID, and environment.

If the **su** command is run from the **/usr/bin/tsh** shell, the trusted shell, you exit from that shell. The **su** command does not change the security characteristics of the controlling terminal.

Each time the **su** command is executed, an entry is made in the **/var/adm/sulog** file. The **/var/adm/sulog** file records the following information: date, time, system name, and login name. The **/var/adm/sulog** file also records whether or not the login attempt was successful: a **+** (plus sign) indicates a successful login, and a **-** (minus sign) indicates an unsuccessful login.

Note: Successful use of the **su** command resets the **unsuccessful_login_count** attribute in the **/etc/security/lastlog** file only if the user's **rlogin** and **login** attributes are both set to **false** in **/etc/security/user**. Otherwise, the **su** command doesn't reset the **unsuccessful_login_count**, because the administrator often uses the **su** command to fix user account problems. The user is able to reset the attribute through a local or remote login.

Flags

Item	Description

-	Specifies that the process environment is to be set as if the user had logged in to the system using the login command. Nothing in the current environment is propagated to the new shell.

Note: This behavior is intended for compatibility with alternate UNIX shell environments where flag options are allowed ahead of the Name parameter.

Security

The **su** command is a PAM-enabled application with a service name of **su**. System-wide configuration to use PAM for authentication is set by modifying the value of the **auth_type** attribute, in the **usw** stanza of **/etc/security/login.cfg**, to **PAM_AUTH** as the root user.

The authentication mechanisms used when PAM is enabled depend on the configuration for the **su** service in **/etc/pam.conf**. The **su** command requires **/etc/pam.conf** entries for the **auth**, **account**, **password**, and **session** module types. In order for the **su** command to exhibit a similar behavior through PAM authentication as seen in standard AIX authentication, the **pam_allowroot** module must be used as sufficient and called before **pam_aix** in both the **auth** and **account** **su** service stacks. Listed below is a recommended configuration in **/etc/pam.conf** for the **su** service:

```
#
# AIX su configuration
#
su auth sufficient /usr/lib/security/pam_allowroot
su auth required /usr/lib/security/pam_aix

su account sufficient /usr/lib/security/pam_allowroot
su account required /usr/lib/security/pam_aix

su session required /usr/lib/security/pam_aix

su password required /usr/lib/security/pam_aix
```


Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

To get the full functionality of the command, besides the **accessauths**, the role should also have the **aix.security.su** authorization.

Examples

1. To obtain root user authority, enter one of the following commands:

```
su
```

This command runs a subshell with the effective user ID and privileges of the root user. You will be asked for the root password. Press End-of-File, Ctrl+D key sequence, to end the subshell and return to your original shell session and privileges.

```
su --
```

This command runs a subshell with the effective user ID and privileges of the root user. Enter the root password, when prompted. Press End-of-File, Ctrl+D key sequence, to end the subshell and return to your original shell session and privileges.

2. To obtain the privileges of the jim user, enter the following command:

```
su jim
```

This command runs a subshell with the effective user ID and privileges of jim.

3. To set up the environment as if you had logged in as the jim user, enter:

```
su - jim
```

This starts a subshell using jim's login environment.

4. To run the backup command with root user authority and then return to your original shell, enter:

```
su root "-c /usr/sbin/backup -9 -u"
```

This command runs the **backup** command with root user authority within root's default shell. You must give the correct root password when queried for the command to execute.

5. Enter one of the following commands to change the user credentials of the current session to root user:

```
su -
```

```
su - root
```

```
su - --
```

The preceding commands start a subshell by using the root user's login environment.

Files

Item	Description
/usr/bin/su	Contains the su command.
/etc/environment	Contains user environment values.
/etc/group	Contains the basic group attributes.
/etc/passwd	Contains the basic user attributes.
/etc/security/user	Contains the extended attributes of users.

Item	Description
<code>/etc/security/envIRON</code>	Contains the environment attributes of users.
<code>/etc/security/limits</code>	Contains the process resource limits of users.
<code>/etc/security/passwd</code>	Contains password information.
<code>/var/adm/sulog</code>	Contains information about login attempts.

subj Command

Purpose

Generates a list of subjects from a document.

Syntax

subj [*File* ...]

Description

The **subj** command searches one or more English-language files for subjects that might be appropriate in a subject-page index and prints the list of subjects on the standard output. The document should contain formatting commands (from the **nroff**, **troff**, and **mm** commands, among others) to make the best use of the **subj** command.

The **subj** command selects sequences of capitalized words as subjects, except for the first word in each sentence. Thus, if a sentence begins with a proper noun, the capitalization rule does not select this word as a subject. However, since each sentence is expected to begin on a new line, the first word of a sentence that begins in the middle of a line may be erroneously selected. Also, the **subj** command selects modifier-noun sequences from the abstract, headings, and topic sentences (the first sentence in each paragraph). Thus, occasionally a word is incorrectly categorized as a noun or adjective.

The output of the **subj** command may not be appropriate for your needs and should be edited accordingly.

Parameters

Item	Description

<i>File</i>	Specifies the English-language files that the subj command searches for appropriate subjects for indexing.

sum Command

Purpose

Displays the checksum and block count of a file.

Syntax

sum [*-i*] [*-r* | *-o*] [*File* ...]

Description

The **sum** command reads the file specified by the *File* parameter and calculates a checksum and the number of 1024-byte blocks in that file. If no options are specified, a byte-by-byte algorithm, such as the BSD 4.3 default algorithm, is used. If no files are named, the standard input is read. The checksum and number of 1024-byte blocks are written to standard output. The **sum** command is generally used to

determine if a file that has been copied or communicated over transmission lines is an exact copy of the original.

Flags

Item	Description

-i	Allows the user to compute the checksum without including header information, if the input file is a binary file. If the input file is not a binary file, the checksum includes header information.
-o	Uses the word-by-word algorithm to compute the checksum. The sum command with the -o flag is compatible with the Version 2 sum command in terms of the checksum, but not the number of blocks.
-r	Uses a byte-by-byte algorithm to compute the checksum. Using the -r flag is the same as using no options.

Note: The default is no longer the word-by-word computation algorithm; it is the BSD 4.3 default algorithm.

Exit Status

This command returns the following exit values:

Item	Description

0	Successful completion.
>0	An error occurred.

Examples

To display the checksum of, and the number of 1024-byte blocks in, the **file1** and **file2** files, type:

```
sum file1 file2
```

If the checksum of the **file1** file is 32830, the checksum of the **file2** file is 32481, and the **file1** file contains one block, and the **file2** contains four blocks, the **sum** command displays:

```
32830      1      file1
32481      4      file2
```

Files

Item	Description
/usr/bin/sum	Contains the sum command.

suma Command

Purpose

Creates a task to automate the download of technology levels and service packs from a fix server.

Syntax

To create, edit, or schedule a Service Update Management Assistant (SUMA) task:

```
suma { { [ -x ] [ -w ] } | -s CronSched } [ -a Field=Value ]... [ TaskID ]
```

To list SUMA tasks:

```
suma -l [ TaskID ]...
```

To list or edit the default SUMA task:

```
suma -D [ -a Field=Value ]...
```

To list or edit the SUMA global configuration settings:

```
suma -c [ -a Field=Value ]...
```

To unschedule a SUMA task:

```
suma -u TaskID
```

To delete a SUMA task:

```
suma -d TaskID
```

Description

The **suma** command can be used to perform the following operations on a SUMA task or policy:

- Create
- Edit
- List
- Schedule
- Unschedule
- Delete

A unique Task ID represents the task on which the specified operation is performed. To create or edit cases on a SUMA task, if the *TaskID* is not specified, the create operation is assumed by default, and a unique *TaskID* is generated. For the **-l** flag, if the *TaskID* is not specified, a list of all SUMA tasks are displayed. For the **-c** flag, if the **-a** flag is not specified, the SUMA global configuration settings are listed.

Flags

Item	Description
-c	Lists or edits the SUMA global configuration settings. The -a flag allows one or more configuration settings to be updated to the specified value. When used without the -a flag, all SUMA configuration settings are listed. Following are the configuration settings that can be edited with the -a flag: FIXSERVER_PROTOCOL Specifies that the transfer uses https (secure) protocol when communicating with the fix server. The https protocol is the only supported protocol and cannot be changed. The default value is https. The valid value is https. DOWNLOAD_PROTOCOL Specifies whether the transfer uses http, or https (secure) protocol when downloading file sets. The http protocol takes advantage of multi-threaded performance and uses the download director protocol (ddp). The https protocol is single-threaded. The default value is http. The valid values are http, and https. DL_TIMEOUT_SEC Specifies the time in seconds to wait for a response from the fix server during a download operation. The default value is 180. The valid values are whole numbers greater than zero. HTTP_PROXY and HTTPS_PROXY Proxy server and port to use for the HTTP or HTTPS transfers. The SUMA command shares the proxy connectivity settings with the Electronic Service Agent. The HTTP or HTTPS proxy service configuration can be set up through the SMIT Create/Change Service Configuration menus (use fastpath smitty srv_conn) that allow the server specifications such as IP address, port number, and an optional user ID and password. SUMA no longer supports the settings of the HTTP_PROXY and HTTPS_PROXY parameters. The default value is blank (disabled). The valid value is blank.

Item	Description
-c (Continued)	SCREEN_VERBOSE Specifies a verbosity level for logging information to stdout and stderr. Used when the suma command is run from the command line or from the SMIT interface. It is not applicable for scheduled tasks that are run from cron. The default value is LVL_INFO. Following are the valid values: • LVL_OFF: No information is displayed or logged. • LVL_ERROR: Displays error messages and other highly important messages. • LVL_WARNING: Displays warning messages in addition to LVL_ERROR messages. • LVL_INFO: Displays informational messages in addition to LVL_WARNING messages. • LVL_VERBOSE: Displays verbose informational messages in addition to LVL_INFO messages. • LVL_DEBUG: Displays debug output. This setting is for debugging purposes and must not be used for normal operations. NOTIFY_VERBOSE Specifies a verbosity level for the information that is sent in an email notification. Applies only to scheduled tasks run from cron. The default values LVL_INFO. The valid values are LVL_OFF, LVL_ERROR, LVL_WARNING, LVL_INFO, LVL_VERBOSE, and LVL_DEBUG (refer to the SCREEN_VERBOSE setting for value descriptions) LOGFILE_VERBOSE Specifies a verbosity level for the information that is logged to the log file (/var/adm/ras/suma.log). Note: An LVL_OFF setting logs information to the download log file (/var/adm/ras/suma_d1.log). The default value is LVL_VERBOSE. The valid values are LVL_OFF, LVL_ERROR, LVL_WARNING, LVL_INFO, LVL_VERBOSE, and LVL_DEBUG (refer to the SCREEN_VERBOSE setting for value descriptions) MAXLOGSIZE_MB The maximum size (in MB) that a log file is allowed to reach. The default value is 1. The valid values are whole numbers greater than zero. REMOVE_CONFLICTING_UPDATES Specifies whether lppmgr command must remove conflicting updates that have the same level as base images (lppmgr -u flag) when run during a clean action. The default value is yes. The valid values are yes, and no. REMOVE_DUP_BASE_LEVELS Specifies whether lppmgr command must remove duplicate base levels (lppmgr -b flag) when run during a clean action. The default value is yes. The valid values are yes, and no.
-c (Continued)	REMOVE_SUPERSEDE Specifies whether lppmgr must remove superseded file set updates (lppmgr -x flag) when run during a clean action. The default value is yes. The valid values are yes, and no. TMPDIR Specifies the directory to store temporary files. The default value is /var/suma/tmp. The valid value is any directory that currently exists.
-d	Deletes the SUMA task that is associated with the specified <i>TaskID</i> and any schedules for this task that were created with the -s flag.

Item	Description
-D	Lists or edits the default SUMA task. The -a flag allows one or more <i>Fields</i> of the default task to be updated to the specified <i>Value</i> . When used without the -a flag, the default SUMA task is listed.
-l	Lists SUMA tasks. When used without a <i>TaskID</i> , all SUMA tasks are listed. The <i>TaskID</i> can be used to specify one or more task IDs to list.
-s CronSched	Schedules a SUMA task. If specified when a new task is being created, a save is implied (-w flag functions). The <i>CronSched</i> is a list of five space-separated entries (minute, hour, day, month, weekday) contained in quotation marks. The valid values for these entries are as follows (see the crontab man page for more details): • Minute: 0 - 59 • Hour: 0 - 23 • Day: 1 - 31 • Month: 1 - 12 • Weekday: 0 - 6 (for Sunday - Saturday)
-u	Unscheduled a SUMA task. This flag removes any scheduling information for the specified <i>TaskID</i> .
-w	Writes or saves a SUMA task. If used instead of the -s flag, the task is saved, allowing scheduling information to be added later. If used with the -x flag, the task is run immediately and also saved.
-x	Specifies that a SUMA task must be run immediately and not scheduled. If used without the -w flag, the task is not saved for future use.
-a Field=Value ...	Assigns the specified <i>Value</i> to the specified <i>Field</i> . Following are the supported <i>Fields</i> and <i>Values</i> for the create or edit operation on a SUMA task:

RqType

When suma command is run with an RqType of Latest, the RqType is the only required field. See example 1 for the default values that are used in this case. Other RqType values (TL, SP, ML, PTF) require specification of *Field=Value* information.

ML

Specifies a request to download a specific maintenance or technology level. An example is 5300-11.

TL

Specifies a request to download a specific technology level. An example is 6100-03.

PTF

Specifies a request to download a PTF. An example is U813941. Only certain PTFs might be downloaded as an individual file set. For example, PTFs containing `bos.rte.install`, `bos.alt_disk_install.rte`, or PTFs that come out in between Service Packs. Otherwise, the TL or SP must be downloaded.

SP

Specifies a request to download a specific service pack. An example is 6100-02-04.

Latest

Specifies a request to download the latest fixes. This RqType value returns the latest service pack of the TL specified in *FilterML*.

Item	Description
-a (<i>Continued</i>)	RqName The specific name of the item requested (for example, 6100-03 or 6100-04-03). The RqName field must be blank when RqType equals Latest. Repeats Specifies whether the task is executed once and does not remain on the system, repeats until the item is found, or repeats forever. The Repeats field applies only to scheduled tasks run from cron that have an Action of Download, Clean, or Metadata. If run from the command line or if Action is Preview, this field is ignored, and no task is removed. y Sets up a repeating task, and requires that the task is assigned a <i>CronSched</i> with the -s flag. When the RqType equals TL, SP, PTF, or ML, the task is removed when the item is found. When RqType equals Latest, the task is set up to repeat forever. n Specifies that the task is executed once and does not remain on the system.

Item	Description
-a <i>(Continued)</i>	DisplayName Indicates the display name for this SUMA task (for example, "Download TL 6100-04 when available"). This option is used to view existing SUMA tasks in SMIT. Action Preview Specifies that a download preview is performed. No file sets are downloaded. Download Specifies that file sets are downloaded into the DLTarget based on the policy. Clean Specifies that file sets are downloaded into the DLTarget based on the policy, followed by a clean operation. The lppmgr command is used to clean file sets that are not needed from the DLTarget. The three configurable lppmgr flag options that are listed in the SUMA global configuration settings are: • <u>REMOVE_CONFLICTING_UPDATES</u> • <u>REMOVE_DUP_BASE_LEVELS</u> • <u>REMOVE_SUPERSEDE</u> Metadata Specifies that metadata files are downloaded instead of file set updates. The following RqType values are supported: TL Downloads metadata for a specific technology level. SP Downloads metadata for a specific service pack. Latest Downloads metadata for all service packs for the technology level that is specified for the FilterML flag. DLTarget Contains the directory location where the downloaded files are stored. If this field is not specified, it is given the value <code>/usr/sys/inst.images</code> and the files are stored in a directory based on the image type; for example <code>/usr/sys/inst.images/installp/ppc</code>, or <code>/usr/sys/inst.images/RPMS/ppc</code>. NotifyEmail Contains one or more email addresses (multiple addresses must be comma-separated) that are sent a notification email after a file set download or preview. A notification is sent only if the task is scheduled for execution at a future time (<i>CronSched</i> is specified).

Item	Description
-a (Continued)	FilterDir Specifies the name of a fix repository directory to filter against so that duplicate fixes are not downloaded. This option allows a directory other than the <code>DLTarget</code> to be filtered against. For example, you might filter against a <code>NIM lpp_source</code> without having to download into this directory. If left blank, the <code>DLTarget</code> is used. If the value of the <code>FilterDir</code> field ends with <code>/installp/ppc</code> fix repository directory, the suma command filters the fixes within the specified directory. If the value of the <code>FilterDir</code> field does not end with <code>/installp/ppc</code> fix repository directory, the suma command attempts to search and filter the fixes within the <code>/installp/ppc</code> directory. If the fix repository directory that is specified by the <code>FilterDir</code> field has the <code>/installp/ppc</code> directory, the filter process is successful. If the fix repository directory that is specified by the <code>FilterDir</code> field does not have the <code>/installp/ppc</code> directory, the filter process fails. FilterML Specifies a technology level to filter against; for example, 6100-03. If not specified, the value that is returned by <code>oslevel -r</code> on the local system is used. MaxDLSize The maximum allowable amount of data to be downloaded by any single policy execution, in MB. If it is determined that the download operation exceeds this size, no download occurs. A value of "unlimited" or -1 can be specified to indicate no higher limit on the amount of data to be downloaded. Extend Specifying <code>y</code> automatically extends the filesystem where the <code>DLTarget</code> resides. If <code>n</code> is specified and more space is required for the download, no download occurs.

Item	Description

-a (Continued)

MaxFSSize

Specifies the maximum allowable size to which the DLTARGET filesystem can be extended, in MB. If it is determined that the download operation exceeds this limit, no download occurs. A value of unlimited or -1 can be specified to indicate no higher limit on the size of the filesystem. The filesystem can be expanded until physical disk space is exhausted if no higher limit is set on the size of the filesystem.

The actual size of the DLTARGET filesystem after expansion might be slightly larger than the requested size. The expansion process of the DLTARGET filesystem uses the physical partition (PP) size of the volume group (VG). The required filesystem size is rounded up to the next multiple of a PP when expanding the filesystem. The **suma** command multiplies the total download size of the technology levels and service packs by a fudge factor of 1.15 to calculate the required filesystem memory for the download of the technology levels and service packs.

You can calculate the value for the MaxFSize option by using the following formula:

$$Y = X * 1.15 / (1024 * 1024)$$

where Y is the required filesystem size in MB and X is the total bytes of updates downloaded.

If the available free memory (in MB) in the filesystem is denoted by Z, and the value of Y is greater than Z, more memory is required in the filesystem to download the technology levels and service packs. The extra memory that is required in the filesystem can be calculated as the difference of the values of Y and Z in MB. If the total size of the filesystem is denoted by T (in MB), then the filesystem is expanded to the value of F, where F is calculated by using the following formula:

$$F = T + (Y - Z)$$

Setting the MaxFSize option to a value greater than the value of F ensures that the filesystem is expanded to a size that enables download of fixes to be completed successfully.

Parameters

Item	Description

<i>TaskID</i>	Specifies a unique numeric identifier that is associated with a task. This ID is assigned when a task is created.

Exit Status

Item	Description

0	The command completed successfully.
>0	An error occurred.

Examples

1. To list the SUMA global configuration settings, run the following command:

```
suma -c
```

Output similar to the following output is displayed:

```
FIXSERVER_PROTOCOL=https
DOWNLOAD_PROTOCOL=http
DL_TIMEOUT_SEC=180
DL_RETRY=1
HTTP_PROXY=
HTTPS_PROXY=
SCREEN_VERBOSE=LVL_INFO
NOTIFY_VERBOSE=LVL_INFO
LOGFILE_VERBOSE=LVL_VERBOSE
MAXLOGSIZE_MB=1
REMOVE_CONFLICTING_UPDATES=yes
REMOVE_DUP_BASE_LEVELS=yes
REMOVE_SUPERSEDE=yes
TMPDIR=/var/suma/tmp
```

2. To edit the SUMA global configuration setting to change the maximum log file size to 2 MB, run the following command:

```
suma -c -a MAXLOGSIZE_MB=2
```

3. To list the SUMA task defaults, run the following command:

```
suma -D
```

Output similar to the following output is displayed:

```
DisplayName=
Action=Download
RqType=Latest
RqName=
Repeats=y
DLTarget=/usr/sys/inst.images
NotifyEmail=root
FilterDir=/usr/sys/inst.images
FilterML=7300-02
MaxDLSize=-1
Extend=y
MaxFSSize=-1
```

4. To create and schedule a task that downloads the latest fixes monthly (for example, on the 15th of every month at 2:30 AM.), run the following command:

```
suma -s "30 2 15 * *" -a RqType=Latest \
-a DisplayName="Latest fixes - 15th Monthly"
```

Note: A task ID is returned for this newly created task. This example assumes that some of the SUMA task defaults are used, as displayed in the `suma -D` example. For example, when the task default of `DLTarget=/usr/sys/inst.images`, the installp images are downloaded into the `/usr/sys/inst.images/installp/ppc` directory.

5. To view SUMA scheduling information that is set up by running a `suma -s CronSched` command, run the following command:

```
crontab -l root
```

6. To create and schedule a task that checks for a specific TL once a week (for example, every Thursday at 3 AM.), downloads it when it becomes available, and sends email notifications to users on a remote system, run the following command:

```
suma -s "0 3 * * 4" -a RqType=TL -a RqName=6100-04 \
-a NotifyEmail="bob.smith@host2,ann@host2"
```

Note: To make a weekly check for a TL, the `Repeats` field needs to be set to `y`. In this case, after the TL is found, the task is deleted. If `Repeats=n`, only a single check occurs before deleting the task.

7. To create and schedule a task that checks for critical fixes monthly (for example, on the 20th of every month at 4:30 AM.), run the following command:

```
suma -s "30 4 20 * *" -a RqType=Latest -a RqName= \
-a RqLevel=latest -a Repeats=y
```

Note: By setting Repeats=y, this task 'repeats forever' and is not deleted after a successful download.

8. To create and schedule a task that downloads the entire AIX Version 7.1 with the 5300-11 Recommended Maintenance package into the /lppsrc/5311 directory on Monday at 11:00 PM, and runs an **lppmgr** command clean operation after the download operation to remove any superseded updates, duplicate base levels, and conflicting updates, run the following command:

```
suma -s "0 23 * * 1" -a Action=Clean -a RqType=ML -a RqName=5300-11 \
-a DLTarget=/lppsrc/5311
```

Note: Before running a task that specifies Action=Clean, you can run `suma -c` to verify the SUMA global configuration settings that are used when you run **lppmgr** command. In this case, having REMOVE_SUPERSEDE, REMOVE_DUP_BASE_LEVELS, and REMOVE_CONFLICTING_UPDATES all set to **yes** results in the action described earlier.

9. To create and schedule a task that downloads the entire AIX Version 7.1 with the 5300-11 Recommended Maintenance package into the /tmp/lppsrc/5311 directory on Monday at 11:00 PM, filtering against any updates already contained in /lppsrc, run the following command:

```
suma -s "0 23 * * 1" -a RqType=ML -a RqName=5300-11 \
-a DLTarget=/tmp/lppsrc/5311 -a FilterDir=/lppsrc -a FilterSysFile=/dev/null
```

Note: After the task is successfully completed, the task is removed because RqType=TL is a repeat until found task. However, if Repeats=n, only a single check for the 5300-03 TL is made, and if the TL is not found on the fix server, the task is deleted because it is set up not to repeat.

10. To immediately execute a task that performs a preview to check whether an SP exists on the fix server, and to create and save this task for later scheduling if the SP does not yet exist, run the following command:

```
suma -x -w -a Action=Preview -a RqType=SP -a RqName=6100-04-02
```

Note: A task ID is returned for this newly created task.

11. To immediately execute the newly created task from the earlier example (assume task ID 23 was returned) and attempt to download the SP and save the Action=Download setting for task ID 23, run the following command:

```
suma -x -w -a Action=Download 23
```

Note: Because this task is being run from the command line, and not scheduled through cron, the Repeats field are ignored and this task is not deleted regardless of whether the SP is found.

12. To schedule task ID 23 to repeatedly check for a specific SP once a week (for example, every Thursday at 3 AM.), and download it when it becomes available, run the following command:

```
suma -s "0 3 * * 4" -a Repeats=y 23
```

Note: This task is deleted when the SP is found.

13. To unschedule a task that removes its scheduling information from the crontab file in the /var/spool/cron/crontabs directory, run the following command:

```
suma -u 23
```

14. To delete a task that also removes its scheduling information if it exists, run the following command:

```
suma -d 23
```

15. To list multiple SUMA tasks, where 4 and 23 represent task IDs, run the following command:

```
suma -l 4 23
```

16. To list all SUMA tasks, run the following command:

```
suma -l
```

17. To create and schedule a task that checks monthly (for example, on the 15th of every month at 2:30 AM.) for the latest service pack on the specified `FilterML`, and download any that are not already in the `/tmp/latest` repository, run the following command:

```
suma -s "30 2 15 * *" -a RqType=Latest -a FilterML=6100-02 \  
-a DLTarget=/tmp/latest -a FilterDir=/tmp/latest
```

Note: A task ID is returned for this newly created task.

Location

`/usr/suma/bin/suma`

Files

Item	Description
<code>/usr/suma/bin/suma</code>	Contains the suma command.
<code>/usr/sbin/suma</code>	Link to <code>/usr/suma/bin/suma</code> .
<code>/var/adm/ras/suma.log</code>	Contains detailed results from running the suma command.
<code>/var/adm/ras/suma_dl.log</code>	Contains a list of files that are downloaded.
<code>/var/spool/cron/crontabs</code>	Directory that contains the crontab file for scheduling.

suspendvsd Command

Purpose

suspendvsd – Deactivates an available virtual shared disk.

Syntax

```
suspendvsd {-a | vsd_name...}
```

Description

The **suspendvsd** command brings the specified virtual shared disks from the active state to the suspended state. They remain available. Read and write requests which were active while the virtual shared disk was active are suspended and held. Subsequent read and write operations are also held. If the virtual shared disk is in the suspended state, this command leaves it in the suspended state.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit vsd_mgmt
```

and select the **Suspend a Virtual Shared Disk** option.

Under normal circumstances, you should not issue this command. The Recoverable virtual shared disk subsystem uses this command to manage shared disks in a controlled manner. If you issue this command, the results may be unpredictable.

Flags

-a

Specifies that all the virtual shared disks in the active state are to be suspended.

Parameters

vsd_name

Specifies a virtual shared disk. If the virtual shared disk is not in the active state, you get an error message.

Security

You must have root authority to run this command.

Exit Status

0

Indicates the successful completion of the command.

nonzero

Indicates that an error occurred.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **startdomain** command. To bring a particular node online in an existing peer domain, use the **startnode** command. For more information on creating and administering an RSCT peer domain, refer to the *RSCT: Administration Guide*.

Under normal circumstances, you should not issue this command. The Recoverable virtual shared disk subsystem uses this command to manage shared disks in a controlled manner. If you issue this command, the results may be unpredictable.

Standard Output

Current RVSD subsystem run level.

Examples

To bring the virtual shared disk **vsd1vg1n1** from the active state to the suspended state, enter:

```
suspendvsd vsd1vg1n1
```

Location

/opt/rsct/vsd/bin/suspendvsd

svmon Command

Purpose

Captures and analyzes a snapshot of virtual memory.

Syntax

Command report

svmon -C *commands* [-O *options*] [-t *count*] [-i *interval* [*numintervals*]] [-@ [**ALL | *wparnames*]]**

Detailed segment report

svmon -D sids [**-O** options] [**-i** interval [numintervals]]

Global report

svmon -G [**-O** options] [**-i** interval [numintervals]] [**-@** [**ALL** | wparnames]]

Process report

svmon -P [pids] [**-O** options] [[**-t** count] [**-i** interval [numintervals]]] [**-@** [**ALL** | wparnames]]

Segment report

svmon -S [sids] [**-O** options] [**-t** count] [**-i** interval [numintervals]] [**-@** [**ALL** | wparnames]]

User report

svmon -U [lognames] [**-O** options] [**-t** count] [**-i** interval [numintervals]] [**-@** [**ALL** | wparnames]]

Workload management class report

svmon -W [classnames] [**-O** options] [**-t** count] [**-i** interval [numintervals]] [**-@** [**ALL** | wparnames]]

Workload management tier report

svmon -T [tiers] [**-O** options] [**-a** supclassname] [**-t** count] [**-i** interval [numintervals]] [**-@** [**ALL** | wparnames]]

XML report

svmon X [**-o** filename] [**-i** interval [numintervals]] [**-c** < comment >] [**-O** options]

Description

The **svmon** command displays information about the current state of memory. However, the displayed information does not constitute a true snapshot of memory because the **svmon** command runs at user level with interrupts enabled.

If you specify no flag, the **svmon** command, by default, reports real memory at the system level.

You can see memory consumption details and generate the following types of reports. To see more information about a type of report, select one of the following links:

- [Command report](#)
- [Detailed segment report](#)
- [Global report](#)
- [Process report](#)
- [Segment report](#)
- [User report](#)
- [Workload management class report](#)
- [Workload management tier report](#)
- [XML report](#)

The output of these reports can be in compact format or long format. To generate compact format report, specify the **-O** flag. If you do not specify the **-O** flag, the report is in long format.

Command report

The command report displays the statistics of memory use for the specified command. To print this report, specify the **-C** flag. The command report can be in compact format or in long format:

Item	Description
Compact report	A one line summary for each command. To set compact report as the default format, specify the -O flag.
Long report	A multiple lines report for each command that contains a summary, a size-per-page report, and the details of the segments. To set long report as the default format, do not specify the -O flag.

Detailed segment report

The detailed segment report displays detailed information about the primary segments that are specified. To print the detailed segment report, specify the **-D** flag.

The detailed segment report is in long report format only.

Global report

The global report displays the statistics of the real memory and paging space that are in use for the whole system. If you do not specify any flag, the global report is the default format of report that the **svmon** command generates.

To print the global report, specify the **-G** flag.

The global report can be in compact format or long format:

Item	Description
Compact report	A report on only the main metrics of the system. This report is one line with a maximum of 160 characters.
Long report	A summary of memory, page size, and affinity domain. The report is multiple lines, which is the default format of global report. By default, the following metrics are displayed: • The memory metric displays the memory consumption of the machine. • The Page Size metric displays the memory consumption of the Page Size. • The Affinity Domain metric reports the memory affinity by affinity domain.

Note: Pinned memory pages in the Global report of the **svmon** command includes kernel locked pages when the kernel lock (**vmm_klock_mode** option) is enabled. For more information about the kernel lock option, refer to the **vmo -h vmm_klock_mode** command documentation.

Process report

The process report displays the memory use for the specified active process. If you do not specify a list of processes, the **svmon** command displays the memory use statistics for all active processes.

To print the process report, specify the **-P** flag.

The process report can be in compact format or long format:

Item	Description
Compact report	A one line report for each process. To set the compact report as the default format, specify the -O flag.
Long report	A multiple lines summary for each process. To set the long report as the default format, do not specify the -O flag. This report contains a summary for each process, a per-page-size report, and the details of the segments.

Note: The **svmon** command does not show the decrease in the count for the memory usage when the application releases the memory. When the memory is released from the application, it goes back to the memory free list of the per-process. The **svmon** command accounts for the memory that is released as the allocated memory for that application.

Segment report

The segment report displays the statistics of memory use for the specified segments. To display the statistics for all of the defined segments, do not specify any list.

To print the segment report, specify the **-S** flag.

The segment report includes metrics for each specified segment. The report contains several lines of metrics for each segment.

User report

The user report displays the statistics of memory use for the specified users (login names). To display the statistics for all of the users, do not specify any list of login names.

To print the user report, specify the **-U** flag.

The user report can be in compact format or long format:

Item	Description
Compact report	A one line report for each user. To set the compact report as the default format, specify the -O flag.
Long report	A multiple lines summary for each user. To set the long report as the default format, do not specify the -O flag. This report contains a summary for each user, a per-page-size report, and the details of the segments.

Workload management class report

The workload management class report displays statistics of memory use for the specified workload management classes. To display the statistics for all of the defined classes, do not specify any class.

To print the workload management class report, specify the **-W** flag.

Restriction: This report is available only when the Workload Manager is running. If the Workload Manager is not running, the following message is displayed and no statistics are reported:

```
WLM must be started
```

If the Workload Manager is running in passive mode, the **svmon** command displays the following message before displaying the statistics:

```
WLM is running in passive mode
```

The workload management class report can be in compact format or long format:

Item	Description
Compact report	A one line report for each class. To set the compact report as the default format, specify the -O flag.
Long report	A multiple lines summary for each class. To set the long report as the default format, do not specify the -O flag. This report contains a summary for each class, a per-page size report, and the details of the segments.

Workload management tier report

The workload management tier report displays information about the tiers, such as the tier number, the superclass name, and the total number of pages in real memory from segments belonging to the tier.

To print the tier report, specify the **-T** flag. Only the long report format is supported.

Restriction: This report is available only when the Workload Manager is running. If the Workload Manager is not running, the following message is displayed and no statistics are reported:

```
WLM must be started
```

If the Workload Manager is running in passive mode, the **svmon** command displays the following message before displaying the statistics:

```
WLM is running in passive mode
```

XML report

You can use the **svmon** command with an **-X** flag to generate a report in XML format. The XML report contains data of the global environment, the processes, the segments, the users, the workload management classes, and the commands running on the system.

The report is by default printed to the standard output. To print the output to a file named *filename*, specify the **-O filename** flag. The extension of the output file will be **.svm**.

The **.svm** file uses an XML Schema Definition (XSD) that the **/usr/lib/perf/svmon_schema.xsd** file defines. You can use the XML data in the XML reports to build custom applications because the schema is self-documented.

In the XML report, if you do not specify the **-O affinity** argument, or set it to the off value, only the domain affinity at system level is reported.

Flags

If no command line flag is given, then the **-G** flag is the default.

Item	Description
-@ [ALL <i>wparnames</i>]	Displays report for the workload partitions. The -@ ALL option specifies to display the report for all of the WPAR starting with the global report, and to process all of the available WPAR, sorting them by the name. When you specify a list of WPAR names in the <i>wparnames</i> parameter, the WPAR information is displayed in a header, and the report is displayed without adding WPAR information. All information displayed is restricted to the WPAR that was processed and has meaning only inside the WPAR. For example, the <i>pid</i> displayed is virtual <i>pid</i>, which is the <i>pid</i> inside the WPAR. The same rule applies to the <i>svmon</i> options. Each WPAR name in the list is processed in the given order and each <i>svmon</i> report is separated by the WPARname header. When you do not specify a list, the <i>svmon</i> command adds WPAR information to existing reports. The pid section and segments section of the report contain the WPAR name when one is available. Virtual <i>pid</i> information might also be displayed. When all of the keywords are used, the <i>svmon</i> command processes all of the available WPAR, sorting them by the WPAR name. Note: The -@ flag is not supported when executed within a workload partition.
-a <i>supclassname</i>	Restricts the scope to the subclasses of the <i>supclassname</i> parameter (in the Tier report that is returned with the -T flag).
-c < <i>comment</i> >	Adds a comment, specified by the <i>comment</i> parameter, into the XML report. Use the -c flag with the -X flag.
-C <i>commands</i>	Displays memory use statistics for the processes running the commands that are specified by the <i>commands</i> parameter.
-D <i>sids</i>	Displays memory use statistics for the segments that the <i>sids</i> parameter specifies, and a detail status of all of the frames of each segment.
-G	Displays a global report.
-i <i>interval</i> [<i>numintervals</i>]	Displays statistics repetitively. The svmon command collects and prints statistics in the interval that the <i>interval</i> parameter specifies. The <i>numintervals</i> parameter specifies the number of repetitions. If the <i>numintervals</i> parameter is not specified, the svmon command runs until you interrupt it (Ctrl+C). Tip: The observed interval might be larger than the specified interval because it might take a few seconds to collect statistics for some options.
-o <i>filename</i>	Specifies the output file with the <i>filename</i> parameter for XML reports. Use this flag with the -X flag.
-O <i>options</i>	Changes the content and presentation of the reports that the svmon command generates. You can specify values to the <i>options</i> parameter to modify the output. Tip: To overwrite the default values that are defined previously by the -O options flag, you can define the .svmonrc configuration file in the directory where the svmon command is launched.

Item	Description
-P [<i>pids</i>]	Displays the memory-usage statistics for the processes that the <i>pids</i> parameter specifies.
-S [<i>sids</i>]	Displays the memory-usage statistics for segments that the <i>sids</i> parameter specifies. The <i>sids</i> parameter is a hexadecimal value. The segment IDs (SIDs) that are specified must be of primary segments. If you do not specify a list of SIDs, the statistics of memory use are displayed for all of the defined segments.
-t <i>count</i>	Displays the top object in the <i>count</i> parameter to be printed.
-T [<i>tiers</i>]	Displays the memory-usage statistics of all of the classes of the tier numbers that the <i>tiers</i> parameter specifies. If you do not specify a list of tiers, the statistics of memory use are displayed for all of the defined tiers.
-U [<i>lognames</i>]	Displays the memory-usage statistics for the login name that the <i>lognames</i> parameter specifies. If you do not specify a list of login identifiers, the statistics of the memory use are displayed for all of the defined login identifiers.
-W [<i>classnames</i>]	Displays the memory-usage statistics for the Workload Manager class that the <i>classnames</i> parameter specifies. If you do not specify a list of class names, the statistics of memory usage are displayed for all of the defined class names.
-X	Generates the XML report.

Parameters

Item	Description
<i>commands</i>	Specifies the commands to be reported in the command report (-C). The value of the <i>commands</i> parameter is a string. You can specify more than one command. The value of the <i>commands</i> parameter is the exact base name of an executable file.

Item*options***Description**

Specifies the content and presentation of each report. Use this parameter with the **-O** flag.

The values of the *options* parameter must be separated by commas, or enclosed in quotation marks (" ") and separated by commas or spaces. The following values are valid to the *options* parameter.

Tip: The **scope** specifies the reports that support the value.

- **activeuser** = [on | off]

The **activeuser** argument specifies that the **svmon** command displays only the active user.

- **Default value:** off
- **Scope:** User report (**-U**)

You can specify the following values to the **activeuser** option:

on
Displays only the active user.

off
Displays all of the user.

- **affinity** = [on | detail | off]

The **affinity** argument specifies that the **svmon** command displays the memory affinity at process level or segment level.

- **Default value:** off
- **Scope:** Global report (**-G**), process report (**-P**), and segment report (**-S**)

You can specify the following values to the **affinity** option:

on
Displays memory affinity at process level

detail
Displays memory affinity at segment level

off
Does not display the memory affinity

In the XML report, if you do not specify the **-O affinity** argument, or set it to the off value, only the domain affinity at system level is reported.

Note:

1. Use the **-O affinity = detail** argument with caution.
2. The **summary** argument with the value of *longreal* or *longname* is not supported with the **affinity** argument.

- **commandline** = [on | off]

The **commandline** argument specifies that the **svmon** command displays the command that is used for the current report.

- **Default value:** off
- **Scope:** All reports

You can specify the following values to the **commandline** option:

on
Displays the command that is used for the current report

off
Does not display the command that is used for the current report

options

(*Continued description of the valid values for the options parameter.*)

- **file_mem_scan** = [on | off]

If the segment information for some files, such as remote files is not updated by the file system, by default, value of the **svmon** command does not collect the segment information for those files. By turning **file_mem_scan=on**, the **svmon** command scans the entire system's segment table to gather segment information of those files.

- **Default value:** off
- **Scope:** Command report (**-C**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**), global report when affinity is on (**-G -O affinity = on**)

You can specify the following values for the **file_mem_scan** option:

on
Displays the report with client segments for all files including the files for which the segment information is not updated by the file system.

off
Displays the report with client segments for all files excluding the files for which the segment information is not updated by the file system.

Note: If you use the value of **file_mem_scan = on**, the performance might be impacted based on the number of files opened while running the command and the number of segments in the system.

Item

options

Description

(Continued description of the valid values for the options parameter).

- **filename** = [on | off]

The **filename** argument specifies that the **svmon** command displays the file names of each file segment.

- **Default value:** off

- **Scope:** Command report (**-C**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **filename** option:

on

Displays the file names of each file segment

off

Does not display the file name of each file segment

Note: Use the **filename** argument with caution.

- **filtercat** = [off exclusive kernel shared unused unattached]

The **filtercat** argument specifies that the **svmon** command filters the segments by category.

- **Default value:** off

- **Scope:** Command report (**-C**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **filtercat** option to filter the segments by category:

kernel

Filters the kernel segments.

exclusive

Filters the exclusive segments. The exclusive segments are used by only one process, except the shared-memory segments that are always reported as either shared or unattached.

shared

Filters the shared segments. The shared segments are used by more than one process, or shared-memory segments used by at least one process.

unused

Filters the unused segments. The unused segments are not used by any processes.

unattached

Filters the unused shared-memory segments. The unattached segments are shared-memory segments that are not used by any process.

off

Deactivates the filter. The **off** option is the same as the command **-O filtercat = "kernel exclusive shared unused"**.

Note: The **filtercat** option changes the value of the reported basic metrics in the summary header because it adds or removes segments from the report.

Item*options***Description***(Continued description of the valid values for the options parameter).*

- **filterpgsz** = [off s m L S]

The **filterpgsz** argument specifies that the **svmon** command filters the segments by page size.

- **Default value:** off

- **Scope:** Command report (**-C**), detailed segment report (**-D**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **filterpgsz** option to filter the segments by page size:

- off** Filters the segments that are 4 KB (small) in page size
- m** Filters the segments that are 64 KB (medium) in page size
- L** Filters the segments that are 16 MB (large) in page size
- S** Filters the segments that are 16 GB (supreme) in page size
- off** Deactivates the **filterpgsz** option

Note: The **filterpgsz** argument changes the values of the reported metrics in the summary header, because it adds or removes segments from the report.

To filter segments of different page sizes, you can specify various parameters in the form of *<min_size><max_size>*.

For example, to filter the segments with small page size and the segments with small and medium page sizes, enter the following command:

```
svmon -O filterpgsz="sm s"
```

- **filterprop** = [off notempty data text]

The **filterprop** argument specifies that the **svmon** command filters the segments report by property.

- **Default value:** off

- **Scope:** Command report (**-C**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **filterprop** option to filter the segments by property:

- notempty** Filters the segments with value that is in use and is not equal to zero
- data** Filters the data segments, which are computational
- text** Filters the text segments, which are not computational
- off** Deactivates the **filterprop** option

Note: The **filterprop** argument changes the value of the reported basic metrics in the summary header because it adds or removes segments from the report.

Item*options***Description***(Continued description of the valid values for the options parameter).*

- **filtertype** = [off working persistent client]

The **filtertype** argument specifies that the **svmon** command filters the segments by type.

- **Default value:** off

- **Scope:** Command report (**-C**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **filtertype** option to filter the segments by type:

working

Filters the working segments

persistent

Filters the persistent segments, such as the segments on journaled file system (JFS)

client

Filters the client segments, such as the segments on enhance journaled file system (JFS2) or network file system (NFS)

off

Deactivates the **filtertype** option, which is the same as the **-O filtertype = "working persistent client"** command

Note: The **filtertype** argument changes the value of the reported basic metrics in the summary header, because it adds or removes segments from the report.

- **format** = [80 | 160 | nolimit]

The **format** argument specifies the maximum width, in characters, for the output of the **svmon** command.

- **Default value:** 80

- **Scope:** Command report (**-C**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **format** option:

80

Limits the width of the output to 80 characters. In a process report, some fields are truncated. In a segment report, some fields are displayed on separate lines.

160

Limits the width of the output to 160 characters. In a process report, some fields are truncated. In a segment report, some fields are displayed on separate lines.

nolimit

Does not limit the width in character. Does not truncate fields or display them in separate lines. Some columns of the report might be shifted.

Tip: You can use the **summary** argument to force the value of the **format** option to 160 characters.

- **frame** = [on | off]

The **frame** argument specifies that the **svmon** command displays the information per frame.

- **Default value:** off

- **Scope:** Detailed segment report (**-D**)

You can specify the following values to the **frame** option:

on

Displays the information per frame

off

Displays the report automatically

Item

options

Description

(Continued description of the valid values for the options parameter).

- **mapping** = [on | off]

The **mapping** argument specifies that the **svmon** command displays the source segments that are associated with the segments that are created by the **mmap** subroutine (also known as the **mmap** segments). When the source segments do not pertain to the process address space and the **mapping = on** value is specified, the source segments are integrated into the report and are flagged with an asterisk (*).

- **Default value:** off

- **Scope:** Command report (**-C**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **mapping** option:

- on**

Displays the source segments that are associated to the segments created by the **mmap** subroutine

- off**

Does not display the source segments that are associated with the segments created by the **mmap** subroutine

Note: The **mapping** argument changes the values of the reported metrics in the summary header because it adds or removes segments from the report.

- **maxbufsize=size[KB | MB | GB]**

The **maxbufsize** argument modifies the memory buffer size to store the data related to the segment identifiers.

When you run the **svmon** command with the **-P** flag and when the process has more segments, the svm (snapshot of virtual memory) process might fail because of less buffer size. In such cases, use the **-O maxbufsize** flag to increase the buffer size. The **maxbufsize** argument overrides the default buffer size value and uses the specified value. In AIX 7.2.4, or later, the default buffer size is 2 MB. In AIX 7.2.3, or earlier, the default buffer size 512 KB.

- **mpss** = [on | off]

The **mpss** argument breaks down the value of the mixed page size segment into individual page sizes.

- **Default value:** off

- **Scope:** Command report (**-C**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **mpss** option:

- on**

Breaks down the value of the mixed page size segment into individual page sizes

- off**

Does not break down the value of the mixed page size segment

- **overwrite** = [on | off]

The **overwrite** argument overwrites the XML file that the **svmon** command produced.

- **Default value:** on

- **Scope:** XML report (**-X**)

You can specify the following values to the **overwrite** option:

- on**

Overwrites the XML file that the **svmon** command generated

- off**

Does not overwrite the XML file

Item**Description**

options

(Continued description of the valid values for the options parameter).

- **pgsz** = [on | off]

The **pgsz** argument specifies that the **svmon** command displays the sections per page size.

- **Default value:** off

- **Scope:** Command report (**-C**), process report (**-P**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **pgsz** option:

on

Displays the sections per page size

off

Displays the report automatically

- **pidlist** = [on | number | off]

The **pidlist** argument specifies that the **svmon** command displays a list of process IDs (PIDs) or the number of different PIDs for each segment.

- **Default value:** off

- **Scope:** Command report (**-C**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **filename** option:

on

Displays a list of process IDs for each segment.

For special segments, a label is displayed instead a list of process IDs. The following labels are displayed:

- **System segment:** Labels the segments that are flagged as system segments

- **Unused segment:** Labels the segments that are not used by any existing processes. For example, persistent segments that are relative to the files that are no longer in use.

- **Unattached segment:** Labels the shared-memory segments that are not used by any existing processes.

- **Shared-library text:** Labels the segments that contain a shared library. The shared library can be used by most of the processes. This label prevents the display of a long list of processes.

number

Displays the number of different process IDs for each segment.

off

Does not displays the list or number of process IDs for each segment.

options

(Continued description of the valid values for the options parameter).

- **process** = [on | off]

The **process** argument specifies that the **svmon** command displays the list of the processes that belong to the entity.

- **Default value:** off

- **Scope:** Command report (**-C**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **process** option:

on

Displays the list of the processes that belong to the entity

off

Does not display the list of processes that belong to the entity

- **range** = [on | off]

The **range** argument specifies that the **svmon** command displays the ranges of pages within the segments that have been allocated.

- **Default value:** off

- **Scope:** Command report (**-C**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **range** option:

on

Displays the ranges of pages within the allocated segments

off

Does not display the ranges of pages within the allocated segments

Item**Description**

- **segment** = [on | category | off]

The **segment** argument specifies that the **svmon** command displays the segment statistics for entities.

- **Default value:** off

- **Scope:** Command report (**-C**), process report (**-P**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **segment** option:

on

Displays a unique segment list. The segments are sorted by the values of the **sortseg** argument.

category

Groups the segments in three categories: system, exclusive, and shared. The segments in each category are sorted by the values of the **sortseg** argument.

off

Does not display the segment lists.

- **shmid** = [on | off]

The **shmid** argument displays the shared-memory ID that is associated with a shared-memory segment.

Restriction: The **shmid** argument cannot work with a workload partition.

- **Default value:** off

- **Scope:** Command report (**-C**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **shmid** option:

on

Displays the shared-memory ID associated to a shared-memory segment

off

Does not display the shared-memory ID associated to a shared-memory segment

Note: Use the **shmid** argument with caution.

Item*options***Description***(Continued description of the valid values for the options parameter).*

- **sortentity** = [inuse | pin | pgsp | virtual]

The **sortentity** argument specifies the method for the **svmon** command in sorting the reports.

- **Default value:** inuse

- **Scope:** Command report (**-C**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **sortentity** option to sort the reports:

inuse

Sorts the reports in decreasing order of real memory consumption

pin

Sorts the reports in decreasing order of pinned memory consumption

pgsp

Sorts the reports in decreasing order of paging space consumption

virtual

Sorts the reports in decreasing order of virtual memory consumption

- **sortseg** = [inuse | pin | pgsp | virtual]

The **sortseg** argument specifies the method for the **svmon** command in sorting the segment reports.

- **Default value:** inuse

- **Scope:** Command report (**-C**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **sortseg** option to sort the segment reports:

inuse

Sorts the segments in decreasing order of real memory consumption

pin

Sorts the segments in decreasing order of pinned memory consumption

pgsp

Sorts the segments in decreasing order of paging space consumption

virtual

Sorts the segments in decreasing order of virtual memory consumption

- **subclass** = [on | off]

The **subclass** specifies that the **svmon** command displays the statistics of memory use for the subclass of the workload management classes.

- **Default value:** off

- **Scope:** Workload management tier report (**-T**) and workload management class report (**-W**)

You can specify the following values to the **subclass** options:

on

Displays the statistics of memory use of the workload management classes' subclasses

off

Does not display the statistics of memory use of the workload management classes' subclasses

Item**Description***options*

(Continued description of the valid values for the options parameter).

- **summary** = [basic | longreal | ame | longame]

The **summary** argument specifies the format to display the summary for the **svmon** command.

- **Default value:** basic
- **Scope:** Command report (**-C**), global report (**-G**), process report (**-P**), user report (**-U**), and workload management class report (**-W**) summary = [ame | longame] is available only with global report (**-G**).

You can specify the following values to the **summary** option:

basic

Displays the basic headers for the **svmon** command

longreal

Displays the real memory information in a long format (160 columns per line).

Note: The **summary** argument with the value of longreal is supported along with the **-G** flag only.

ame

Displays the Active Memory Expansion information (in an Active Memory Expansion enabled system).

longame

Displays the Active Memory Expansion information (in an Active Memory Expansion enabled system) in a long format.

- **svmonalloc** = [on | off]

The **svmonalloc** argument specifies that the **svmon** command displays the maximum size of the memory that it dynamically allocated during its processing.

- **Default value:** off
- **Scope:** All reports

You can specify the following values to the **svmonalloc** options:

on

Displays the maximum size of the allocated memory

off

Does not display the maximum size of the allocated memory

- **threadaffinity** = [on | off]

The **threadaffinity** argument specifies that the **svmon** command displays the home SRADIDs (Scheduler Resource Allocation Domain Identifier) and the thread SRAD (Scheduler Resource Allocation Domain) affinity statistics for the threads of a process.

- **Default value:** off
- **Scope:** Process report (**-P**)

You can specify the following values to the **threadaffinity** option:

on

Displays the home SRADIDs and thread SRAD affinity statistics for the threads of a process.

off

Does not display the home SRADIDs and thread SRAD affinity statistics for the threads of a process.

- **timestamp** = [on | off]

The **timestamp** argument specifies that the **svmon** command displays the timestamp at the beginning of the report.

- **Default value:** off
- **Scope:** Command report (**-C**), process report (**-P**), segment report (**-S**), workload management tier report (**-T**), user report (**-U**), and workload management class report (**-W**)

You can specify the following values to the **timestamp** option:

on

Displays the time stamp at the beginning of the report

off

Does not display the time stamp at the beginning of the report

Item	Description
<i>options</i>	(Continued description of the valid values for the <i>options</i> parameter). • tmem = [on off] - The tmem argument specifies the svmon command to append the true memory details. – Default value: on – Scope: Global report (-G). You can specify the following values to the tmem option. on Displays the true memory information at the end of the report off Does not display the true memory information. Note: The summary argument must have the value of ame. • unit = [auto page KB MB GB TB] - The unit argument modifies the metrics unit of the report. – Default value: page – Scope: Command report (-C), process report (-P), segment report (-S), workload management tier report (-T), user report (-U), and workload management class report (-W) You can specify the following values to the unit option: auto Expresses the values in the most appropriate unit with at most three significant digits. The unit used in the report is specified for each metric. page Expresses the values in 4 KB page units. The unit used in the report is specified in the report header. KB Expresses the values in kilobytes (KB) MB Expresses the values in megabytes (MB) GB Expresses the values in gigabytes (GB) TB Expresses the values in terabytes (TB) Tip: To overwrite the default values that are defined previously by the -O options flag, you can define the .svmonrc configuration file in the directory where the svmon command is launched.
<i>count</i>	Specifies the top object to be printed. Use the <i>count</i> parameter with the -T flag.
<i>interval</i>	Specifies the interval for the svmon command to collect and print statistics. Use the <i>interval</i> parameter with the -i flag.
<i>numintervals</i>	Specifies the number of repetitions for the svmon command to collect and print statistics when the <i>interval</i> parameter is specified. Use the <i>numintervals</i> parameter with the -i interval option. If the <i>numintervals</i> parameter is not specified, the svmon command runs until you interrupt it (Ctrl+C).
ALL	Specifies that the -@ flag displays the report for all of the WPAR starting with the global report, and then process all of the available WPAR, sorting them by the WPAR name.
<i>wparnames</i>	Specifies the workload partitions whose information is to be displayed. When you specify the -@ wparnames option, all of the information displayed is restricted to the WPAR that the <i>wparnames</i> parameter specifies, and has meaning only inside the WPAR. Each WPAR name in the list is processed in the given order and each svmon report is separated by the WPARname header.
<i>sids</i>	Specifies the segment IDs (SIDs). The SIDs must be primary segments.
<i>pids</i>	Specifies the process IDs (PIDs). The value of the <i>pids</i> parameter is a decimal value. If you do not supply any list of process IDs (PIDs), the statistics of memory use are displayed for all active processes. Use the <i>pids</i> parameter with the -P flag.
<i>lognames</i>	Specifies the login names. The value of the <i>lognames</i> parameter is a string. It is an exact login name. If you do not specify any lists of login identifiers, the statistics of the memory use are displayed for all of the defined login identifiers. Use the <i>lognames</i> parameter with the -U flag.
<i>classnames</i>	Specifies the Workload Manager class. The value of the <i>classnames</i> parameter is a string. It is the exact name of a class. For a subclass, the name should be in the form <i>superclassname.subclassname</i> .
<i>tiers</i>	Specifies a tier number for the classes. If you do not specify a list of tiers, the statistics of memory use are displayed for all of the defined tiers. Use the <i>tiers</i> parameter with the -T flag.
<i>supclassname</i>	Specifies the name of the superclass that the subclasses are restricted to. You cannot specify a list of classes for this flag.

Item	Description
<i>filename</i>	Specifies the name of the output file. It is an alpha-numeric string. The suffix of the output file name is .svm . It is automatically added to the file name if you do not specify the suffix. Use the <i>filename</i> parameter with the -o flag and the -X flag.
<i>comment</i>	Specifies the string to add in the <CollectionHeader><Comment> tag of the XML report. Use the <i>comment</i> parameter with the -X flag and the -c flag.

Security

Any user can run the **svmon** command. If the user is not a root user, the view will be limited to the user's own processes.

If RBAC is activated and the **aix.system.stat** role that is attributed to the user, the user can see the same view that the root user does.

Examples

1. To display global statistics in a one line format every minute for 30 minutes, enter the following command:

```
# svmon -G -O summary=longreal -i 60 30
```

2. To display global statics with automatic unit selection, a time stamp, per page size data, and detailed affinity information, enter the following command:

```
# svmon -G -O unit=auto,timestamp=on,pgsz=on,affinity=detail
```

3. To display global statistics for the system and all of its WPAR in a compact format, enter the following command:

```
# svmon -G -O summary=longreal -@ ALL
```

4. To display the memory consumption in megabytes (MB) of all processes in a compact report, enter the following command:

```
# svmon -P -O summary=basic,unit=MB
```

5. To display the memory consumption of all processes according to the number of virtual pages, and sort the segments for each process by the number of pages in the paging space, enter the following command:

```
# svmon -P -O segment=on,sortentity=virtual,sortseg=pgsp
```

6. To display the memory consumption of process 123456 in full detail, enter the following command:

```
# svmon -P 123456 -O
segment=on,pidlist=on,range=on,mapping=on,shmid=on,filename=on,affinity=detail
```

7. To display the top 10 system segments sorted by the number of pages in real memory, enter the following command:

```
# svmon -S -t 10 -O filtercat=kernel,sortseg=inuse
```

8. To display all of the segments that are not attached to a process, enter the following command:

```
# svmon -S -O filtercat=unattached
```

9. To display only 16 MB segments with their address ranges, enter the following command:

```
# svmon -S -O filterpgsz=L -O range=on
```


10. In the global WPAR, to display the WPAR name that each segment belongs to, enter the following command:

```
# svmon -S -@
```

11. To display the memory consumption of all Oracle processes in a compact report for only the shared segments, enter the following command:

```
# svmon -C oracle -O summary=basic,filtercat=shared
```

12. To display the top 10 users running the processes that consume the most memory every minute, enter the following command:

```
# svmon -U -t 10 -O summary=basic -i 60
```

13. To display the memory use for the Mysupclass superclass with its subclasses, enter the following command:

```
# svmon -W Mysupclass -O subclass=on
```

14. To display the memory use for the 0 tier subclasses of the Mysupclass superclass, enter the following command:

```
# svmon -T 0 -a Mysupclass
```

15. To display the frames that belong to the 36cfb segment with frame level details, enter the following command:

```
# svmon -D 36cfb -O frame=on
```

16. To generate an XML report in the **lpar01.svm** file, enter the following command:

```
# svmon -X -o lpar01.svm  
# svmon -X -o lpar01
```

17. To generate an XML report with affinity domain details, enter the following command:

```
# svmon -X -o lpar_affinity -O affinity=on
```

18. To generate an XML report with affinity domain details at the segment level, enter the following command:

```
# svmon -X -o lpar_affinitydet -O affinity=detail
```

19. To display global statistics with memory compression details along with true memory snapshot at the end, enter the following command:

```
# svmon -G -O summary=ame
```

20. To display global statistics with memory compression details with true memory details turned-off, enter the following command

```
# svmon -G -O summary=ame,tmem=off
```

21. To display global statistics with Active Memory Expansion details (in an Active Memory Expansion enabled system) in a one line format, enter the following command

```
# svmon -G -O summary=longame
```

22. To display the home SRADIDs and thread SRAD affinity statistics for the threads of a process, enter:

```
# svmon -P 1 -O threadaffinity=on
```

swap Command

Purpose

Provides a paging space administrative interface.

Syntax

swap [**-a** *device*] [**-d** *device*] [**-s**] [**-l**]

Description

The functions provided by the swap command are display of characteristics, addition of paging space and removal of paging space.

Flags

Item	Description
-a <i>device</i>	Activates the paging space. Performs the same function the swapon command.
-d <i>device</i>	Deactivates the paging space. Performs the same function as the swapoff command.
-l	Lists the status of paging space areas in a list form. The output has 4 columns, containing the following information: device Path name of the page space. maj/min The major/minor device number for the device. total Total size in megabytes for the area. free Amount of available space.
-s	Prints summary information about total paging space usage and availability. The following information is displayed in the output (amounts of paging space are listed in 4K byte blocks). allocated Total amount of paging space area currently allocated. used Total amount of paging space area currently being used. available Total amount of free paging space. These numbers include paging spaces from all configured areas as listed by the -l option on active paging space. Note: There is a paging space limit of 64 GB per device.

Exit Status

- 0**
The command completed successfully.
- >0**
An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To print summary information on total paging space, enter:

```
swap -s
```

2. To list the status of the paging space areas in a list form, enter:

```
swap -l
```

3. To activate a particular paging space device paging01, enter:

```
swap -a /dev/paging01
```

Files

Item	Description
/usr/sbin/swap	Contains the System V swap command.

swapoff Command

Purpose

Deactivates one or more paging spaces.

Syntax

swapoff *DeviceName* { *DeviceName* ... }

Description

The **swapoff** command deactivates one or more paging spaces. The paging spaces are specified by *DeviceName*.

Note: There is a paging space limit of 64 GB per device.

To be deactivated:

- The paging space must have been previously activated through the **swapon** command.
- There must exist enough space in the remaining paging spaces. The remaining paging device should have enough space to accommodate the current system-wide paging space usage and the **npswarn** value.

Note: This command is not supported when executed within a workload partition.

Exit Status

Item	Description
Value	Description
0	Deactivation is successful, the paging state is set to the INACTIVE state.

Item	Description

1	The following message displays:

```
swapoff: Cannot deactivate paging space DeviceName
```

2	There is not enough space in the remaining paging spaces, the deactivation is not done and the following message displays:

```
"swapoff: Cannot deactivate paging space DeviceName :  
There is not enough space in the file system."
```

3	An I/O error occurred on user pages of a paging space, the following message displays:

```
swapoff: Deactivation of paging space DeviceName suspended:  
I/O errors encountered on user backing pages.
```

The recommended action is:

- Check the error log.
- Deactivate the paging space for the next reboot using the **chps** command.
- Reboot the system.

4	An I/O error occurred on system pages of a paging space, the following message displays:

```
swapoff: Deactivation of paging space DeviceName suspended:  
I/O errors encountered on system backing pages. The system may crash.
```

The recommended action is:

- Check the error log.
- Deactivate the paging space for the next reboot using the **chps** command.
- Reboot the system.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

swapon Command

Purpose

Activates a paging space.

Syntax

swapon -a | *devicename*

Description

The **swapon** command activates a paging space. It is used during early system initialization to make the initial paging space available. During a later phase of system initialization, the **swapon -a** command is used to make other devices available so that paging and swapping activity is interleaved across several devices. If the option **auto=yes** then the **swapon -a** command makes all devices specified in the **/etc/swapspace**s available that aren't explicitly excluded from being automatically swapped on by

their stanza. Calls to the **swapon** command normally occur in the system multiuser initialization **/etc/rc** file.

The *devicename* parameter specifies a specific device to be made available. The second form gives individual block devices as given in the system swap configuration table. The call makes this space and other defined spaces available to the system for paging and swap allocation. The system swap configuration table is the set of all devices specified in the **/etc/swapspaces** file.

Note: The maximum number of active paging spaces is 16. In addition, there is a paging space limit of 64 GB per device.

Note: This command is not supported when executed within a workload partition.

Flags

Item	Description

-a	Causes all devices present in the /etc/swapspaces file to be made available.

Security

The Role Based Access Control (RBAC) Environment: This command implements and can perform privileged operations. Only privileged users can execute such privileged operations.

To review the list of privileges and the authorizations associated with this command, refer to the **/etc/security/privcmds** database.

Examples

1. To cause all devices present in the **/etc/swapspaces** file to be made available, enter:

```
swapon -a
```

All devices present in the **/etc/swapspaces** file are now available.

2. To cause the **/dev/paging03** and **/dev/paging04** devices to be available for paging and swapping, enter:

```
swapon /dev/paging03 /dev/paging04
```

The **/dev/paging03** and **/dev/paging04** devices are now available.

Files

Item	Description
/etc/rc	System multiuser initialization
/dev/paging	Device entries for paging/swap space
/etc/swapspaces	Contains a list of swap devices.

swcons Command

Purpose

Redirects, temporarily, the system console output to a specified device or file.

Syntax

```
swcons [ -p Log_File ] [ -s Log_Size ] [ -t Tag_Verbosity ] [ -v Log_Verbosity ] PathName
```

Description

The **swcons** command temporarily switches the system console output to a different target during system operation. This command only switches system informational-, error-, and intervention-required message output to the specified destination. The **swcons** command does not affect the operation of the system console device that is providing a login by way of the **getty** command.

The device or file specified when using this command remains the target for console output until changed by another **swcons** command, until the next start of the system, or until the console driver detects an error when accessing the designated device or file. If an open or write error is detected on the device or file specified by the **swcons** command, the console device driver switches all output back to the device or file that provided console support when the system was last started.

The *PathName* parameter must be a fully qualified path name to a device or file that is to receive system console message output. If the *PathName* parameter specifies a file that does not exist, the **swcons** command creates the file. If the file does exist, the **swcons** command appends any new console message output to the contents of the file.

Attention: Use of the **swcons** command to switch console output to an NFS mounted file system or a diskless/dataless client might cause the operating system to hang.

Flags

Item	Description
-p <i>Log_File</i>	Specifies the full path name to use for the console output log file.
-s <i>Log_Size</i>	Specifies the size, in bytes, of the console output log file.
-t <i>Tag_Verbosity</i>	Specifies the verbosity level for console output tagging. Zero disables tagging; 1 through 9 enable tagging. For additional information about console output logging and tagging, see the console Special File in the <i>Files Reference</i> book.
-v <i>Log_Verbosity</i>	Specifies the verbosity level for console output logging. Zero disables logging; 1 through 9 enable logging.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To change the system console message output to a file called `console.out` in the `/tmp` directory, enter:

```
swcons /tmp/console.out
```

2. To change the system console message output to a terminal with the logical name `tty3`, enter:

```
swcons /dev/tty3
```

3. To change the system-console message output back to the device or file that supported the console output at system start time, enter:

```
swcons
```

Files

Item	Description
<code>/dev/console</code>	Specifies the special file for system console access.
<code>/usr/sbin/swcons</code>	Contains the swcons command file.

swrole Command

Purpose

Switches to a specified role session.

Syntax

```
swrole { ALL | Role [ ,Role ] ... } [ Argument ... ]
```

Description

The **swrole** command creates a new role session with the roles that is specified by the *Role* parameter. The *Role* parameter must be composed of the names of roles in the **roles** attribute of the user. Before creating a new role session, the **swrole** command performs authentication according to the **auth_mode** attribute of the **chrole** command for the specified roles. If any of the specified roles requires authentication, the user must be successfully authenticated for the action to be performed. If none of the specified roles require authentication, no authentication is requested.

The **swrole** command creates a new role session with the specified roles added to the active role set of the session. The **ALL** keyword specifies that a role session is created with all the roles that are assigned to the user. Role sessions are limited to eight roles per session. If a user has more than eight roles, only the first eight roles are assigned to the role session when the **ALL** keyword is specified. Creation of a new role session preserves the user environment for the current session.

Any argument, such as a flag or a parameter, which is specified by the *Arguments* parameter, must relate to the login shell that is defined for the user. The arguments are passed to the login shell that is created for the role session. For example, if the login shell for a user is `/usr/bin/ksh`, any of the flags that are allowed for the **ksh** command can be specified.

To restore the previous session, type `exit` or press the Ctrl-D. The action ends the shell created by the **swrole** command and returns the user to the previous shell and environment.

Each time the **swrole** command is run, an entry is made in the `/var/adm/rolelog` file. The `/var/adm/rolelog` file records the following information: date, time, system name, login name and role name. The `/var/adm/rolelog` file also records whether or not the role initiation attempt is successful: a plus sign (+) indicates a successful role initiation, and a minus sign (-) indicates an unsuccessful role initiation.

The **swrole** command is functional only when the system is operating in enhanced Role Based Access Control (RBAC) mode. If the system is not in enhanced RBAC mode, the command displays an error message and returns failure.

Examples

1. To assume the RoleAdmin and FSAdmin roles as a user who has been assigned the roles, enter the following command:

```
swrole RoleAdmin,FSAdmin
```

2. To run the **backup** command as a role that has the appropriate authorization, enter the following command:

```
swrole FSAdmin "-c /usr/sbin/backup -9 -u"
```

swts Command

Purpose

Switches a thin server to a different COSI.

Syntax

swts **-c** *Image* [**-n** | **-t** *Time*] [**-v**] *ThinServer*

Description

The **swts** command switches a thin server to a different Common Operating System Image (COSI). If specified with the **-t** flag, the thin server switches to a new common image at the time specified by the *Time* parameter. The value for *Time* must be a valid cron tab entry. Refer to the **crontab** command for creating valid cron time entries.

The **swts** command can be run on either a NIM master or a thin server. When a thin server is switched to a new common image, files in the `/inst_root` directory for the thin server will be synced with the new common image.

Flags

Item	Description
-c <i>Image</i>	Specifies the common image that the thin server switches to.
-n	Specifies option to allow a thin server to switch to a new common OS image that was setup by the NIM administrator with the -c flag. The user running from the thin server will only need to execute the swts command without any argument to switch the common OS images.
-t <i>Time</i>	Specifies a cron entry that allows thin servers to be switched over at a more convenient time.
-v	Enables verbose debug output when the swts command runs.

Exit Status

Item	Description
0	The command completed successfully.
>0	An error occurred.

Security

Access Control: You must have root authority to run the **swts** command.

Examples

1. To switch the `cosi1` common image of a thin server named `lobo` to a common image named `cosi2`, enter:

```
swts -c cosi2 lobo
```


The lobo thin server is re-initialized and cosi2 is its new operating system.

2. To switch the cosi1 common image of a thin server named lobo to a common image named cosi2 at midnight on Sunday, December 25, enter:

```
swts -c cosi2 -t "0 0 25 12 0" lobo
```

The lobo thin server will continue to use the cosi1 common image until midnight on Sunday, December 25, when it switches to cosi2.

Location

/usr/sbin/swts

Files

Item

/etc/niminfo

Description

Contains variables used by NIM.

sync Command

Purpose

Updates the i-node table and writes buffered files to the hard disk.

Syntax

sync

Description

The **sync** command runs the **sync** subroutine. If the system must be stopped, run the **sync** command to ensure file system integrity. The **sync** command writes all unwritten system buffers to disk including modified i-nodes, delayed block I/O, and read-write mapped files.

Note: The writing, although scheduled, is not necessarily complete upon return from the **sync** subroutine.

synclvodm Command

Purpose

Rebuilds the logical volume control block, the device configuration database, and the device special files.

Syntax

synclvodm [**-c** | **-D** | **-F** | **-k** | **-K** | **-P** | **-R** | **-v**] *VolumeGroup LogicalVolume ...*

Description

The **synclvodm** command rebuilds the logical volume control block, the device configuration database, and the device special files (for the volume group and logical volumes), so that they are synchronized with the volume group descriptor areas on the physical volumes.

During normal operations, the device configuration database remains consistent with the logical volume manager information in the logical volume control blocks and the volume group descriptor areas on the physical volumes. If for some reason the device configuration database is not consistent with Logical Volume Manager information, the **synclvodm** command can be used to resynchronize the database. The

volume group must be active for the resynchronization to occur (see **varyonvg**). If logical volume names are specified, only the information related to those logical volumes is updated. If logical volume names are not specified, every logical volume in the volume group is updated.

Attention: Do not remove the **/dev** entries for volume groups or logical volumes. Do not change the device configuration database entries for volume groups or logical volumes using the object data manager.

Note: To use this command, you must either have root user authority or be a member of the **system** group.

Flags

Item	Description

-c	Treats naming conflicts as fatal errors. If this flag is not specified, the command generates a warning message for any naming conflicts, and automatically renames the logical volume by default. A logical volume naming conflict occurs when the logical volume name is already in use by another device. A volume group naming conflict occurs when the volume group major number cannot be reserved in the device configuration database.
-D	Does not remove or recreate the logical volume minor numbers and device special files. If not specified, the command removes and recreates the logical volume minor numbers and device special files by default.
-F	Does not synchronize the device configuration database entries for the physical volumes in the volume group. If this flag is not specified, the command removes the device configuration database entries for all physical volumes in the volume group, and recreates those entries based on the information in the volume group descriptor area by default.
-k	Takes the volume group lock when the synclvodm command is running. If this flag is not specified, the volume group lock is taken only if the parent process does not have the lock.
-K	Does not take the volume group lock when the synclvodm command is running. Use this flag when the caller is a shell script, and is managing the volume group lock in the shell script with the putlvodm -k and -K flags. The default behavior is to take the volume group lock unless the parent process has the lock.
-P	Preserves the permission bits for the special files of logical volume device. The -P flag overrides the -D flag. The -P flag is ignored for original type volume groups. If this flag is not set, the ownership of the logical volume special file is set to root, and the group is set to system.
-R	Restores the user, group, and permissions for the logical volume device special files to the values previously set by the mklv and chlv commands using the -U , -G , and -P flags. The -R flag is ignored for original type volume groups, or when the -D flag is specified.
-v	Displays the output from the synclvodm command in verbose mode.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

To synchronize the device configuration database with the logical volume manager information for rootvg, enter the following:

```
synclvodm rootvg
```

Files

Item	Description
<code>/usr/sbin/synclvodm</code>	Contains the synclvodm command.

syncroot Command

Purpose

Synchronizes a non-shared portion of installed software with a shared part.

Syntax

```
/usr/sbin/syncroot [[ -a ] [ -i ] ] [ -F ] [ -r ] ] [ -p ] [ -v ] [ -X ]
```

Flags

Item	Description
<code>-a</code>	Performs additional installation only. Does not downlevel the installp file sets (that is, uninstall, reject, force overwrite). Not valid with the <code>-r</code> flag.
<code>-i</code>	Only updates installp file sets. Not valid with the <code>-r</code> flag.
<code>-F</code>	Forces copy RPM files. Not valid with the <code>-i</code> flag.
<code>-r</code>	Only updates RPM files. Not valid with the <code>-i</code> flag.
<code>-p</code>	Previews operation. Do not actually performs the synchronization.
<code>-v</code>	Specifies the verbose mode.
<code>-X</code>	Expands file systems if necessary and possible.

Notes:

- If you are logged into a version 6 workload partition, on a version 7 global system, and run the **syncroot** command, the operation will fail with the following error:

```
syncroot: Processing root part installation status.  
Your global system is at a higher version than the WPAR.  
Please log out of the WPAR and execute the migwpar command.  
syncroot: Returns Status = FAILURE
```

- For shared workload partitions (WPARs), if the installation history of the source system is different from the installation history of the target system, the **restwpar** command and the **syncroot** command might fail for few filesets. You might see a failure message that is similar to the following example for the **syncroot** command, at the end of the **restwpar** operation:

```
syncroot: Error synchronizing installp software  
syncroot: Returns Status = FAILURE
```

You must restore or migrate the shared WPAR to a logical partition (LPAR) that has the installation history similar to the installation history of the source LPAR.

Security

Access Control: Only the root user can run this command.

Examples

1. To update all **installp** filesets in the root part, enter:

```
# syncroot -i
```

2. To perform an update of all **RPM** files and expand space automatically (if needed and possible), enter:

```
# syncroot -r -X
```

syncvg Command

Purpose

Synchronizes logical volume copies that are not current.

Syntax

```
syncvg [ -f ] [ -i ] [ -H ] [ -P NumParallelLps ] { -l | -p | -v } Name ... { [ -a { all | pid1,pid2,... } ] [ -r { all | pid1,pid2,... } ] [ -t { all | pid1,pid2,... } ] [ -n vgName ] [ -T SyncRate [ -d { all | pid1,pid2,... } ] ] [ -q ] [ -Q ] }
```

Description

The **syncvg** command synchronizes the physical partitions, which are copies of the original physical partition, that are not current. The **syncvg** command can be used with logical volumes, physical volumes, or volume groups, with the *Name* parameter representing the logical volume name, physical volume name, or volume group name. The synchronization process can be time consuming, depending on the hardware characteristics and the amount of data.

When the **-f** flag is used, a good physical copy is chosen and propagated to all other copies of the logical partition, whether or not they are stale. Using this flag is necessary in cases where the logical volume does not have the mirror write consistency recovery.

Unless disabled, the copies within a volume group are synchronized automatically when the volume group is activated by the **varyonvg** command.

Note: For the **syncvg** command to be successful, at least one good copy of the logical volume should be accessible, and the physical volumes that contains this copy should be in ACTIVE state. If the **-f** option is used, the above condition applies to all mirror copies.

If the **-P** option is not specified, **syncvg** will check for the *NUM_PARALLEL_LPS* environment variable. The value of *NUM_PARALLEL_LPS* will be used to set the number of logical partitions to be synchronized in parallel.

Flags

Item	Description
-a { <i>all</i> <i>pid1,pid2,...</i> }	Pauses one or more sync operations. The following parameters can be passed to this option: all Pause all sync operations. pid1,pid2,... A comma separated list of process ID (PID) to pause.
-f	Specifies a good physical copy is chosen and propagated to all other copies of the logical partition, whether or not they are stale.

Item	Description
-H	Postpones writes for this volume group on other active concurrent cluster nodes until this sync operation is complete. When using the -H flag, the -P flag does not require that all the nodes on the cluster support the -P flag. This flag is ignored if the volume group is not varied on in concurrent mode.
-i	Reads the names from standard input.
-l	Specifies that the <i>Name</i> parameter represents a logical volume device name.
-n <i>vgName</i>	Manages sync operations for a specific volume group. This option is only valid with the -a , -r , -t , -q and -Q options. vgName Volume group name.
-p	Specifies that the <i>Name</i> parameter represents a physical volume device name.
-P <i>NumParallelLps</i>	Numbers of logical partitions to be synchronized in parallel. The valid range for <i>NumParallelLps</i> is 1 to 32. <i>NumParallelLps</i> must be tailored to the machine, disks in the volume group, system resources, and volume group mode. When a volume group is varied on in concurrent mode, all other cluster nodes that have this volume group varied must be at least AIX 4.3.0, otherwise syncvg will ignore this option and continue. Note: See Description above for more information.
[-q]	Queries sync operations. A verbose list of sync operation Process Identifiers (PIDs) is returned. This flag also outputs the sync rate for each sync operation. If the <i>SyncRate</i> option is not specified using the -T flag, this flag displays the current sync rate of the sync operation.
[-Q]	Queries sync operations. A comma separated list of sync operation PIDs is returned. This flag also returns the sync rate for each sync operation. If the <i>SyncRate</i> option is not specified using the -T flag, this flag displays the current sync rate of the sync operation.
{-r all pid1,pid2,... }	Resumes one or more sync. The following parameters can be passed to this option: all Resumes all sync operations. pid1,pid2,... A comma separated list of PIDs to resume.
{-t all pid1,pid2,... }	Terminates one or more sync. The following parameters can be passed to this option: all Terminates all sync operations. pid1,pid2,... A comma separated list of PIDs to terminate.

Item	Description
[-T SyncRate [-d { all pid1,pid2,... }]]	Throttles the sync rate of the current sync operation or throttles one or more sync operations that are in progress. The following parameters can be passed to this option: SyncRate Specifies the sync rate, in MB/sec, to throttle. The syncvg command synchronizes one Logical Track Group (LTG) at a time. This parameter must be specified in multiples of the LTG size of the volume group. If the SyncRate parameter is not specified in the multiples of the LTG size, the syncvg command rounds up to the nearest LTG size of the volume group. If you do not specify the -d flag, the syncvg command throttles the sync rate of the current sync operation. -d all Throttles the sync rate for all the sync operations that are in progress. -d pid1,pid2,... A comma-separated list of PIDs that throttle the sync rate.
-v	Specifies that the <i>Name</i> parameter represents a volume group device name.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To synchronize the copies on physical volumes hdisk4 and hdisk5, enter:

```
syncvg -p hdisk4 hdisk5
```

2. To synchronize the copies on volume groups vg04 and vg05, enter:

```
syncvg -v vg04 vg05
```

3. To display the synchronization status, enter:

```
syncvg -q
An output that is similar to the following example is displayed.
```

VG Name	Status	Sync Rate	PID	Command
tv2	SYNCING	128M	8323316	/bin/ksh /usr/sbin/syncvg -l tv2lv1
tv2	SYNCING	1M	7536758	/bin/ksh /usr/sbin/syncvg -l tv2lv3
tv2	SYNCING	256M	6815782	/bin/ksh /usr/sbin/syncvg -l tv2lv2
tv1	SYNCING	2G	7995416	/bin/ksh /usr/sbin/syncvg -l tv1lv2
tv1	SYNCING	5M	2949162	/bin/ksh /usr/sbin/syncvg -l tv1lv3
tv1	SYNCING	1G	7274582	/bin/ksh /usr/sbin/syncvg -l tv1lv1

4. To pause the **syncvg** command and then display the synchronization status, enter:

```
syncvg -a all
syncvg -q
An output that is similar to the following example is displayed.
```

VG Name	Status	Sync Rate	PID	Command
tv2	PAUSE	128M	8323316	/bin/ksh /usr/sbin/syncvg -l tv2lv1
tv2	PAUSE	1M	7536758	/bin/ksh /usr/sbin/syncvg -l tv2lv3
tv2	PAUSE	256M	6815782	/bin/ksh /usr/sbin/syncvg -l tv2lv2
tv1	PAUSE	2G	7995416	/bin/ksh /usr/sbin/syncvg -l tv1lv2
tv1	PAUSE	5M	2949162	/bin/ksh /usr/sbin/syncvg -l tv1lv3
tv1	PAUSE	1G	7274582	/bin/ksh /usr/sbin/syncvg -l tv1lv1

5. To synchronize the current **syncvg** operation with a sync rate of 512 MB/sec on a volume group named vg00, enter:

```
syncvg -T 512 -v vg00
```

Files

Item	Description
/usr/sbin/syncvg	Contains the syncvg command.
/tmp	Directory where the temporary files are stored and while the command is running.

syncwpar Command

Purpose

Synchronizes software between a global system and a workload partition.

Syntax

Shared WPAR synchronization

```
/usr/sbin/syncwpar [[ -a ] [ -i ] ] [ -F ] [ -r ] ] [ -p ] [ -v ] [ -X ] { -A | -f wparnamesfile | wparname }
```

Detached WPAR synchronization

```
/usr/sbin/syncwpar -D [ -d device ] [ -p ] [ -v ] [ -X ] { -A | -f wparnamesfile wparname }
```

Detached WPAR interim fix operations

```
/usr/sbin/syncwpar -D { -E <path to fix> | -R <ifix label> } { -A | -f wparnamesfile | wparname }
```

Versioned WPAR device data synchronization

```
/usr/sbin/syncwpar -c wparname
```

Description

The **syncwpar** command synchronizes the software that is installed in the global shared parts (usually the **/usr** and **/opt**) with the workload partition *root* part.

If you specify the **-D** flag, the **syncwpar** command recovers the system software that is in a detached workload partition (WPAR) with writable **/usr** directory, and that has diverged from the system software in the global environment. If you do not specify the **-D** flag, the **syncwpar** command runs only on shared WPAR that have a read-only **/usr** directory.

Note: The **syncwpar** command cannot be used to synchronize software levels in AIX 5.2 or AIX 5.3 versioned WPAR. Software in versioned WPAR is independent from the software in the global environment.

The **syncwpar** command operates on a single WPAR when you specify the *wparname* parameter, on a list of WPAR when you specify the *wparname* parameter with the **-f wparnamesfile** parameter, or on all system WPAR when you specify the **-A** flag.

Restriction: Running the **syncwpar** command on application workload partitions is restricted.

Note: If you run the **syncwpar** command to sync a version 6 workload partition, on a version 7 global system, the **syncwpar** command will call the **migwpar** command and migrates the workload partition.

Flags

Item	Description
-a	Performs additional installation only. Does not downlevel the <code>installp</code> filesets (that is, uninstall, reject, force overwrite). Not valid with the <code>-r</code> flag.
-c	Synchronizes the predefined storage device data in a specified versioned workload partition. The <code>-d</code> device flag is not required to synchronize the device data. Synchronization of the device data helps to resolve problems while configuring a storage device in a WPAR.
-D	Synchronizes software in the detached system workload partitions that have a writable <code>/usr</code> directory. The default is to synchronize software in only shared system workload partitions that have a read-only <code>/usr</code> directory.
-i	Only updates the <code>installp</code> filesets. Not valid with the <code>-r</code> flag.
-F	Forces the RPM files to be copied. Not valid with the <code>-i</code> flag.
-r	Updates only the RPM files. Not valid with the <code>-i</code> flag.
-p	Previews the operation. Does not actually perform the synchronization.
-v	Specifies the verbose mode.
-X	Expands file systems if necessary and possible.
-A	Synchronizes all of the available system workload partitions with the global system.
-f	Specifies the file containing a list of workload partitions in the <i>wparnamesfile</i> parameter.
-d	Synchronizes the software in a detached WPAR <i>wpar</i>, using the specific software installation directories. The <code>-d</code> flag is valid only when used with the <code>-D</code> flag. - When the <code>-d</code> flag is specified, the images in the directory are used to apply the base installation or updates to the detached WPARs. It is important that the install or update images in the specified location are the same as the ones that were last used to install or update the global system so that the resulting software levels match. - When the <code>-d</code> flag is not specified, the synchronization rejects or commits the levels of software in the detached WPARs.
-R	Removes the specified interim fix from the WPARs. The argument is the label of the <i>ifix</i> parameter that must be removed. The flag is valid only for detached system workload partitions.
-E	Installs the specified interim fix into the detached system workload partitions. The argument is the full path to the <i>ifix</i> parameter. The flag is valid only for detached system workload partitions.

Parameters

Item	Description
<i>wparnamesfile</i>	Specifies the file that contains a list of workload partition names.
<i>wparname</i>	Specifies the name of a workload partition.
<i>device</i>	Specifies the name of a device.

Security

Access Control: Only the root user can run this command.

Examples

1. To synchronize all of the software on workload partition `mywpar`, enter the following command:

```
syncwpar mywpar
```

2. To synchronize all WPAR, enter the following command in the global environment:

```
# syncwpar -A
```

3. To synchronize WPAR that is named `mywpar` and to expand the file system automatically, enter the following command:

```
# syncwpar -X mywpar
```

4. To synchronize software in the detached WPAR named `privatewpar` using the `/mysw` software installation directory, enter the following command:

```
# syncwpar -D -d /mysw privatewpar
```

5. To install the **myfix.epkg.Z** interim fix to all the detached system workload partitions, enter the following command:

```
# syncwpar -D -E /tmp/myfix.epkg.Z -A
```

6. To remove an interim fix with the label **myfix** from all the detached system workload partitions, enter the following command:

```
# syncwpar -D -R myfix -A
```

syscall Command

Purpose

Performs a specified subroutine call.

Syntax

syscall [-n] *Name* [*Argument1* ... *ArgumentN*] [; *Name* [*Argument1* ... *ArgumentN*]] ...

Description

The **syscall** command executes a system call interface program, which performs the subroutine call specified by the *Name* parameter. If you specify the **-n** flag, the **syscall** command performs the call *n* times. Arguments specified by the *Argument* parameter are passed to the subroutine without error checking. The *Argument* parameter can be expressed in the following formats:

Item	Description
<code>0x nnn</code>	Hexadecimal constant <i>nnn</i> .
<code>0 nnn</code>	Octal constant <i>nnn</i> .
<code>nnn</code>	Decimal constant <i>nnn</i> .
<code>+nnn</code>	Decimal constant <i>nnn</i> .
<code>-nnn</code>	Decimal constant <i>nnn</i> .

Item	Description
<code>"string"</code>	The character string <code>"string"</code> .
<code>'string'</code>	The character string <code>"string"</code> .
<code>\string</code>	The character string <code>"string"</code> .
<code>#string</code>	The length of the character string <code>"string"</code> .
<code>&&n</code>	The address of the n th argument to this subroutine. ($n=0$ is the subroutine name.)
<code>&n</code>	The address of the n th byte in an internal 10KB buffer.
<code>\$n</code>	The result of the n th subroutine. ($n=0$ is the first subroutine.)
<code>string</code>	Anything else is a literal character string.

The **syscall** command prints a message and exits for unknown subroutines and for subroutines that return a value of -1.

Note: The **syscall** command understands the **sleep** subroutine as a special case subroutine.

Flags

Item	Description
-n	Specifies the number of times the syscall command performs the specified subroutine.
;	Separates multiple subroutines (up to a maximum of 20) issued by the same invocation of the syscall command.

Examples

To simulate the C program fragment:

```
output=open("x", 401, 0755);
```

```
write(output, "hello", strlen("hello"));
```

enter:

```
syscall open x 401 0755 \; write \ $0 hello \#hello
```

Note: Special shell characters must be escaped.

Files

Item	Description
<code>/usr/bin/syscall</code>	Contains the syscall command.

sysck Command

Purpose

Checks the inventory information during installation and update procedures.

Syntax

sysck { **-i** | **-u** } [**-R** *RootPath*] [**-N**] [**-v**] [**-s** *SaveFile*] [**-O** { **r** | **s** | **u** }] **-f** *File* *ProductName*
{ **tcback** *Flags* }

All of the **tcback** command flags are valid with this command.

Description

Note: All of the **tcback** command flags are valid with the **sysck** command. This feature provides compatibility with Version 3.1.

The **sysck** command checks file definitions against the extracted files from the installation and update media and updates the Software Vital Product Data (SWVPD) database. The **sysck** command does not recognize the following special characters in file names: grave accent (`), quotation marks (' '), backslash (\), caret (^), parentheses ((,)), vertical bar (|), braces ({, }), brackets ([,]), greater than and less than symbols (<, >), colon (:), and comma (,). If a file name contains one of these characters, the **sysck** command fails.

The **sysck** command is primarily used during the installation and update of software products.

When invoked with the **-i** flag, the **sysck** command checks the attributes of an extracted file with its file definitions, updates the SWVPD, and attempts to fix some errors if they exist.

The *File* parameter is the name of the stanza file that contains the file definitions. An example of such a file is the **/etc/security/sysck.cfg** file, although the **syschk** command does not use this file. The **sysck** command checks the size, links, symlinks, owner, group, and mode attributes of a file for which the type attribute is set to **FILE**. When invoked with the **-v** flag as well as the **-i** flag, **sysck** also checks the checksum value of a file.

The **sysck** command updates the file name, product name, type, checksum, and size of each file in the SWVPD database.

To fix errors, the **sysck** command resets the attribute of the installed or updated file to the defined value in the *File* stanza file, except for some attributes as described in "Fixing Errors".

When invoked with the **-u** flag, the **sysck** command removes the entry from the SWVPD database for each file that is part of the software product *ProductName*. The **sysck** command also deletes any hard links and symbolic links for each file, as defined in the SWVPD database.

Flags

Item	Description
-f <i>File</i>	Specifies the name of the stanza file that contains the file definitions.
-i	Checks for the correct installation of a software product's files. Updates the SWVPD database with the file definitions, and attempts to fix some errors if found.
-N	Specifies that the SWVPD database should not be updated.
-O { r s u }	Specifies which part of the SWVPD is to be updated, as follows: r Specifies the root part of the SWVPD. s Specifies the /usr/share part of the SWVPD. u Specifies the /usr part of the SWVPD (default).
Item	Description
-R <i>RootPath</i>	Use <i>RootPath</i> as root instead of / .

Item	Description
-s <i>SaveFile</i>	Takes a snapshot of what is currently in the VPD and saves it in stanza format to the file specified by <i>SaveFile</i> . Called with the -u option. No action is taken in the database with this flag. Must be used with the -f option. For example:
<pre>sysck -i -s /tmp/save.inv -f /tmp/real.inv bos.rte.shell</pre>	
-u	Deletes file entries from the SWVPD and deletes hard links and symbolic links.
-v	Verifies that the checksum is correct.
<i>ProductName</i>	Specifies the installable software product or option that is being checked.

Environment Variables

Item	Description
INUTREE	The environment variable INUTREE has only the following four valid values: NULL Same as INUTREE not being set. M Specifies the root part of the SWVPD. S Specifies the /usr/share part of the SWVPD. U Specifies the /usr part of the SWVPD (default). INUTREE can be used instead of the -O Tree flag.
INUNOVPD	The environment variable INUNOVPD can be null or can be set to 1. If it is set to 1 then sysck does not update the SWVPD. INUNOVPD can be used instead of the -N flag.
INUVERIFY	If the environment variable INUVERIFY is set to 1 sysck verifies that the checksum attributes in the stanza file are correct. INUVERIFY can be used instead of the -v flag.

File Definitions

Item	Description
acl	The access control list for the file. If the value is blank, the acl attribute is removed. If no value is specified, the command computes a value, according to the format described in Access Control Lists. This attribute should grant x (execute) access only to the root user and members of the security group. The command should setuid to the root user and have the trusted computing base attribute.
class	The logical group of the file. A value must be specified because it cannot be computed. The value is <i>ClassName</i> [<i>ClassName</i>].
checksum	The checksum of the file. If the value is blank, the checksum attribute is removed. If no value is specified, the command computes a value, according to the format given in the sum command. The value is the output of the sum -r command, including spaces.
group	The file group. If the value is blank, the group attribute is removed. If no value is specified, the command computes a value, which can be a group ID or a group name.
mode	The file mode. If the value is blank, the mode attribute is removed. If no value is specified, the command computes a value, which can be an octal number or a string (rwX), and have the TCB , SUID , SGID , and SVTX attributes.

Item	Description
owner	The file owner. If the value is blank, the owner attribute is removed. If no value is specified, the command computes a value, which can be a user ID or a user name.
size	The size of the file in bytes. If the value is blank, the size attribute is removed. A VOLATILE value in the size field indicates that the file size will change (so no checksum value can be given). A NOSIZE value indicates that the file has 0 length. If no value is specified, the command computes a value, which is a decimal number.
target	Allows symbolic links and hard links to exist as separate stanzas in the inventory. The target file definition refers to the full path name of the source of the link, for example: <pre>/etc/foo --> /usr/bar</pre> The target is /usr/bar.
type	The type of file. This value cannot be blank. If no value is specified, the command computes a value, which can be the FILE , DIRECTORY , FIFO , BLK_DEV , CHAR_DEV , LINK , MPX_DEV , and SYMLINK keywords.
xacl	An addition to the extended-access control list. A value must be specified as a single entry in an extended-access control list because the value cannot be computed. This attribute is valid only if the -i flag is used. For information about the format, see the acl file definition above.

Fixing Errors

To fix errors, the **sysck** command resets the attribute of the installed or updated file to the defined value defined in the *File* stanza file except for the following attributes, for which the **sysck** command acts as described:

Item	Description
links	Creates any missing hard links. If a link exists to another file that is not listed in this definition, the link is deleted.
program	If this attribute is included in the <i>File</i> stanza file, sysck invokes the program. A message is printed if an error occurs, but no additional action is taken.
symlinks	Creates any missing symbolic links. If a link exists to another file that is not listed in this definition, the link is deleted.

Security

Privilege Control: Only the root user can run this command.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. A product that uses the **installp** command to install ships an inventory file in its image. To add the definitions to the inventory database and check permissions, links, checksums, etc., enter:

```
sysck -i -f dude.rte.inventory dude.rte
```

where `dude.rte.inventory` would look like the following:

```
/usr/bin/dude.exec:
class = apply,inventory,dude.rte
```

```
owner = bin
group = bin
mode = 555
type = FILE
size = 2744
checksum = "04720      3"
```

2. To remove any links to files for a product that has been removed from the system and remove the files from the inventory database, enter:

```
sysck -u -f dude.rte.inventory dude.rte
```

Files

Item	Description
/etc/objrepos/inventory	Specifies names and locations of files in a software product on the root.
/usr/lib/objrepos/inventory	Specifies names and locations of files in a software product on the /usr file system.
/usr/share/lib/objrepos/inventory	Specifies names and locations of files in a software product on the /usr/share file system.

syscorepath Command

Purpose

Specifies a single system-wide directory where all core files of any processes will be dumped.

Syntax

```
syscorepath [ -p DirectoryName ] [ -g ] [ -c ]
```

Description

The **syscorepath** command enables a system administrator to set up a single system-wide directory in which to dump core files from any processes. This can ease administrative tasks in managing file-system space and provides a single, known directory in which to find core files. By default, the core file is created in the working directory of the process being core-dumped.

The directory should have read and write privileges for all users on the system. If a user does not have permission to write in the directory, a core file will not be created. Core files will be given unique names based on the process ID and time, so a core file will be named **core.pid.ddhhmmss**, where *pid* is the process ID, *dd* is the day of the month, *hh* is the hour in 24-hour format, *mm* is minutes, and *ss* is seconds.

Note: The settings made by the **syscorepath** command do not persist across system reboots. However, the settings made by the **chcore** command persist across system reboots.

Flags

Item	Description
-c	Unsets the current directory specified as the repository for core files. Subsequent core files will be created in the working directory of the process.
-g	Displays current directory specified as the repository for core files.

Item	Description
-p <i>DirectoryName</i>	Specifies the directory to use as a repository for core files. <i>DirectoryName</i> must be a valid directory name.

Exit Status

Item	Description
0	The command completed successfully.
>0	An error occurred.

Standard Errors

EPERM	User does not have permission.
ENOTDIR	Specified <i>DirectoryName</i> is not a directory.
ENAMETOOLONG	Specified <i>DirectoryName</i> is too long.

Security

Only the root user can run this command.

Examples

1. To set **/core** as the repository for core files, type:

```
syscorepath -p /core
```

2. To display the current repository for core files, type:

```
syscorepath -g
```

3. To unset the directory used as the repository for core files, type:

```
syscorepath -c
```

Files

Item	Description
/usr/bin/syscorepath	Contains the syscorepath command.

sysdumpdev Command

Purpose

Displays and modifies the information and settings that are related to traditional system dump and firmware-assisted system dump.

Syntax

sysdumpdev **-P** { **-p** *device* | **-s** *device* } [**-q**] [**-i**]

sysdumpdev [**-p** *device* | **-s** *device*] [**-q**]

sysdumpdev [**-d** *directory* | **-D** *directory* | **-e** | **-I** | [**-k** | **-K**] | **-l** | [**-n** | **-N**] | **-p** *device* | **-q** | **-s** *device* | **-z**]

sysdumpdev [**-i**]

sysdumpdev **-L** { **-v** | **-S** *device* }

sysdumpdev [**-t** { *traditional* | *fw-assisted* }] [**-f** { *disallow*, *allow*, *require* }]

Description

The **sysdumpdev** command changes the primary or secondary dump device designation in a system that is running. The primary and secondary dump devices are designated in a system configuration object. The new device designations are in effect until you run the **sysdumpdev** command again, or you restart the system.

If you run the **sysdumpdev** command with the **no** flag, the **sysdumpdev** command identifies the current attributes of the primary and secondary dump devices and writes the attribute values to the ODM object class and NVRAM. The default primary dump device is **/dev/hd6**. The default secondary dump device is **/dev/sysdumpnull**. If the system has 4 GB or more of memory, then the default dump device is **/dev/ig_dumplv**, and **/dev/ig_dumplv** is a dedicated dump device. AIX Version 7.1, and later, extends firmware assisted dump capabilities to make it as the default system dump method if it is supported by the platform.

Note:

- A mirrored paging space might be used as a dump device.
- Do not use a diskette drive as your dump device.
- If you use a paging device, only use **hd6**, the primary paging device. The AIX operating system supports using any paging device in the root volume group (**rootvg**) as the secondary dump device.
- If you use a removable device such as a tape or DVD, be aware that the dump does not span volumes. Thus, the dump must fit on a single volume.
- You can configure an iSCSI software initiator device in the root volume group (**rootvg**) as the dump device for a firmware-assisted system dump, for AIX Version 6.1 with the 6100-01 Technology Level.
- Remote dumps for thin servers are supported for AIX 6.1. You must define the relative dump resource on the NIM master to see the dump resource on the NIM client as an iSCSI disk that can only be used to configure the primary dump device. Only firmware-assisted system dump can be configured on an iSCSI disk device.
- For AIX Version 6.1 with the 6100-06 Technology Level, you can configure a firmware-assisted dump of kernel memory.
- >| When you configure a dump device for a logical volume that is encrypted and locked, you might see an error message. |<
- >| You cannot configure a dump device for a logical volume if the encryption or decryption of the logical volume is in progress. |<

For AIX 6.1 and later versions, all dumps are compressed. You should use the **savecore** command to copy dumps from the dump device to a file.

The **sysdumpdev** command supports firmware-assisted system dump for the following features:

- Return of dump size estimation
- Display of information about most recent dump
- Detection of a new dump

The **sysdumpdev** command also provides the dump type including the traditional dump type or the *fw-assisted* dump type.

The **-t** flag specifies the type of dump. Its possible values are *traditional* and *fw-assisted*.

The **-f** flag specifies the full memory system dump mode. This mode is relevant only for the firmware-assisted system dump. In this mode, the dump is performed independently of the operating system. All of the partition memory is saved to the dump.

Running sysdumpdev in Non-rootvg Volume Groups

You can use a dump-logical volume outside the root volume group, if it is not a permanent dump device and for a traditional system dump only. For example, if the **-P** flag is not specified. However, if you choose a paging space, the dump device cannot be copied unless it is in rootvg. If the dump device must be copied, only rootvg is active before paging is started.

The primary dump devices must always be in the root volume group for permanent dump devices. The secondary device might be outside the root volume group unless it is a paging space.

Flags

Item	Description
-d <i>directory</i>	Specifies the <i>directory</i> the dump is copied to at system boot. If the copy fails at boot time, you can use the -d flag to ignore the system dump.
-D <i>directory</i>	Specifies the <i>directory</i> the dump is copied to at system boot. If the copy fails at boot time, you can use the -D flag to copy the dump to an external media. Note: When using the -d <i>directory</i> or -D <i>directory</i> flags, the following error conditions are detected: • <i>directory</i> does not exist. • <i>directory</i> is not in the local journaled file system. • <i>directory</i> is not in the rootvg volume group.
-e	Estimates the size of the dump (in bytes) for the current running system. The size that is shown is the estimated size of the compressed dump.
Item	Description
-f { <i>disallow</i> <i>allow_kernel</i> <i>require_kernel</i> <i>allow_full</i> <i>require_full</i> }	Specifies whether firmware-assisted system dump does allow, require or forbid the dump of either the kernel memory or the full memory. In kernel memory or full memory mode, the dump is performed independently of the operating system. All of the kernel relevant memory is saved to a kernel memory system dump. All of the partition memory is saved to a full memory system dump. The -f flag has the following variables: • The <i>disallow</i> variable specifies that neither the full memory system dump mode nor the kernel memory system dump mode is allowed. It is the selective memory mode. • The <i>allow_full</i> variable specifies that the full memory system dump mode is allowed but is performed only when operating system cannot properly handle the dump request. • The <i>require_full</i> variable specifies that the full memory system dump mode is allowed and is always performed. When the full memory dump is allowed, the dump size estimation specified with the -e flag corresponds to the memory size with the applied compression factor.
-i	Indicates that the sysdumpdev command was called from a system function. This flag is only used by system utilities. The -i flag will not make the requested change if the effected value has already been modified by other than an automatic IBM function; that is, the -i flag will not override a previous change.
-I	Resets the indications of previous changes. After the -I flag is specified, changes are allowed with the -i flag.

Item	Description
-k	If your machine has a key mode switch, it is required to be in the service position before a dump can be forced with the dump key sequences.
-K	If your machine has a key mode switch, the reset button or the dump key sequences will force a dump with the key in the normal position, or on a machine without a key mode switch. Note: On a machine without a key mode switch, a dump can not be forced with the key sequence without this value set.
-l	Lists the current value of the primary and secondary dump devices, copy directory, and forcecopy attribute. The -l flag also displays the current dump type. The following list indicates the possible values that are displayed: • fw-assisted: The preferred dump type is firmware-assisted system dump. • fw-assisted (suspend): The preferred dump type is firmware-assisted system dump, but the primary dump device is either not configured or it does not support firmware-assisted system dump. In the latter case, a traditional system dump is triggered. • traditional: Only the traditional system dump is available after the sysdumpdev -t traditional command. It might also because the firmware-assisted system dump is not supported on this system. To support firmware-assisted system dump, there must be sufficient memory when the system starts up, and POWER6 or later hardware and the supported firmware must be installed.
> > -n	Disables the Nest Accelerators (NX) GZIP dump compression. The system dump is compressed without using the NX GZIP. < <
> > -N	Enables the Nest Accelerators (NX) GZIP accelerated dump compression. NX GZIP dump compression is only applicable to firmware-assisted type system dumps. < <
-L	Displays statistical information about the most recent system dump. This includes date and time of last dump, number of bytes written, and completion status. The -L flag shows both the compressed size and the uncompressed size of the dump. The compressed size is the size of what was actually written to the dump device. If no previous dump was recorded in nonvolatile memory, this flag scans the dump devices for the existing dump. Note: 1. The dump sizes shown might not reflect the exact size of the dump on the media. There can be a small difference because of disk and copy block sizes. 2. If the dump has failed due to an I/O error, the major and minor device numbers will be those for the failing device.
-P	Makes permanent the dump device specified by -p or -s flags. The -P flag can only be used with the -p or -s flags.
-p device	Temporarily changes the primary dump device to the specified device. The device can be a logical volume, writable DVD, or a tape device or an iSCSI disk configured by NIM for remote dump.
-q	Suppresses all messages to standard output. If this flag is used with the -l , -z , or -L flag, the -q flag will be ignored.
-s device	Device Temporarily changes the secondary dump device to the specified device. The same devices valid for the -p flag are valid here.

Item	Description
-S <i>device</i>	Scans a specific dump device for a valid compressed dump. The dump must be from an AIX release with parallel dump support. This flag can be used only with the -L flag.
-t { <i>traditional</i> <i>fw-assisted</i> }	Specifies the type of dump to perform. The -t flag has the following variables: - The <i>traditional</i> variable specifies that the traditional system dump is performed. In this dump type, the dump data is saved before the system reboot. Under any of the following circumstances, you can only specify the <i>traditional</i> variable: - Firmware-assisted system dump is not supported. - Memory is not sufficient when the system starts. - POWER6 or later hardware is not installed. You cannot use the traditional system dump on an iSCSI software initiator dump device. - The <i>fw-assisted</i> variable specifies that the firmware-assisted system dump is performed. In this dump type, the dump data is saved in parallel with the system reboot. If the system starts in a low memory configuration, you must explicitly enable the full memory dump using the -f flag, especially in iSCSI software initiator configuration where firmware-assisted system dump cannot fall back on the traditional system dump if the full memory dump is not allowed. If you specify the <i>fw-assisted</i> variable but the primary dump device is either not configured or it does not support firmware-assisted system dump, a traditional system dump is triggered. When the firmware-assisted system dump type is not allowed at configuration time, or is not enforced at dump request time, a traditional system dump is performed. In addition, because the scratch area is only reserved at initialization, a configuration change from traditional system dump to firmware-assisted system dump is not effective until the system is rebooted.
-v	When the dump status is not 0, this option will display available dump debug information. The debug data, when available, is used by service to diagnose dump failures. This flag can only be used with the -L flag.
-z	Determines if a new system dump is present. If one is present, a string containing the size of the dump in bytes and the name of the dump device will be written to standard output. If a new system dump does not exist, nothing is returned. After the sysdumpdev -z command is run on an existing system dump, the dump will no longer be considered recent.

If no flags are used with the **sysdumpdev** command, the default dump devices are used.

Security

Access Control: Only the root user can run this command.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Error Codes

Note: A nonzero dump status indicates a failed dump. The following values are the possible dump status values and their corresponding light-emitting diode (LED) values:

Dump status	Description	LED value
0	Dump completed successfully	0C0
-1	No dump device defined	0C8
-2	Dump device too small	0C4
-3	Dump crashed or did not start	0C5
-4	I/O error	0C1

Examples

1. To display current dump device settings, enter the following command:

```
sysdumpdev -l
```

For information about the types of dump that this command shows, see the **-l** flag description under the Flags section.

2. To designate logical volume hd7 as the primary dump device, enter the following command:

```
sysdumpdev -p /dev/hd7
```

3. To designate tape device rmt0 as the secondary dump device, enter the following command:

```
sysdumpdev -s /dev/rmt0
```

4. To display information from the previous dump invocation, enter the following command:

```
sysdumpdev -L
```

5. To permanently change the database object for the primary dump device to /dev/newdisk1, enter the following command:

```
sysdumpdev -P -p /dev/newdisk1
```

6. To determine if a new system dump exists, enter the following command:

```
sysdumpdev -z
```

If a system dump has occurred recently, an output that is similar to the following is displayed:

```
4537344 /dev/hd7
```

7. To specify the directory that a dump is copied to after a system crash, if the dump device is /dev/hd6, enter the following command:

```
sysdumpdev -d /tmp/dump
```

This attempts to copy the dump from /dev/hd6 to /tmp/dump after a system crash. If there is an error during the copy, the system continues to boot and the dump is lost.

8. To specify the directory that a dump is copied to after a system crash, if the dump device is /dev/hd6, enter the following command:

```
sysdumpdev -D /tmp/dump
```

This attempts to copy the dump from /dev/hd6 to the /tmp/dump directory after a crash. If the copy fails, you are prompted with a menu. You can copy the dump manually to some external media through this menu.

9. To scan a dump device for a dump, enter the following command:

```
sysdumpdev -L -S /dev/hd6
```

sysdumpstart Command

Purpose

Provides a command line interface to start a kernel dump to the primary or secondary dump device.

Syntax

sysdumpstart [-p] [-t *traditional* | -f { *disallow* | *require_kernel* | *require_full* }]

sysdumpstart [-s] [-t *traditional*]

Description

The **sysdumpstart** command provides a command line interface to start a kernel dump to the primary or secondary dump device. When the dump completes, the system halts. Use the **kdb** command to examine a kernel dump. Use the **sysdumpdev** command to reassign the dump device.

During a kernel dump, the following values can be displayed on the three-digit terminal display as follows:

Item	Description

0c0	Indicates that the dump completed successfully.

0c1	Indicates that an I/O occurred during the dump.

0c2	Indicates that the dump is in progress.

0c4	Indicates that the dump is too small.

0c5	Indicates a dump internal error .

0c8	Indicates that the dump was disabled. In this case, no dump device was designated in the system configuration object for dump devices. The sysdumpstart command halts, and the system continues running.

0c9	Indicates that a dump is in progress.

0ca	Indicates that a firmware-assisted system dump is not finished yet. System startup resumes after the dump completes.

0cb	Indicates that a dump is in progress.

0cc	Indicates that the system switched to the secondary dump device after attempting a dump to the primary device.

You could also use the System Management Interface Tool (SMIT) **smit sysdumpstart** fast path to run this command.

You can specify the **-t traditional** flag that allows to force a traditional system dump when the firmware-assisted system dump is configured.

Restriction:

- If traditional system dump is the current configuration, the **sysdumpstart** command cannot start a firmware-assisted system dump.
- If firmware-assisted system dump is the current configuration with an iSCSI software initiator dump device, the **sysdumpstart** command cannot start a traditional system dump.

You can specify the **-f** flag that allows to override the current full memory dump configuration.

Flags

Item	Description
<code>-f{ <i>disallow</i> <i>require_kernel</i> <i>require_full</i> }</code>	Specifies if neither the kernel memory dump nor the full memory dump is allowed. If allowed, this flag specifies where the kernel memory dump or full memory dump is required. The <code>-f</code> flag has the following keywords: • Specify the <i>disallow</i> keyword to start a firmware-assisted system dump of selective memory. • Specify the <i>require_kernel</i> keyword to start a firmware-assisted system dump of kernel memory. • Specify the <i>require_full</i> keyword to start a firmware-assisted system dump of full memory.
<code>-p</code>	Initiates a system dump and writes the results to the primary dump device.
<code>-s</code>	Initiates a system dump and writes the results to the secondary dump device.
<code>-t <i>traditional</i></code>	Forces a traditional system dump independently to the current configuration.

Security

Access Control: Only the root user can run this command.

Examples

1. To start a kernel dump to the primary dump device, enter the following command:

```
sysdumpstart -p
```

2. To start a kernel dump to the secondary dump device, enter the following command:

```
sysdumpstart -s
```

sysline Command

Purpose

Displays system status on the status line of a terminal.

Syntax

```
/usr/bin/sysline [ -b ] [ -c ] [ -d ] [ -e ] [ -h ] [ -i ] [ -j ] [ -l ] [ -m ] [ -p ] [ -q ] [ -r ] [ -s ] [ -w ] [ -D ] [ -H  
Remote ] [ +N ]
```

Description

The **sysline** command runs in the background and periodically displays system status information on the status line of the terminal. Not all terminals contain a status line. If no flags are specified, the **sysline** command displays the following status items:

- Time of day
- Current number of processes which may be run
- Number of users (followed by a u)
- Number of executable processes (followed by an r)
- Number of suspended processes (followed by an s)
- Number of users who have logged on and off since the last status report

Finally, if new mail has arrived, a summary of it is printed. If there is unread mail in your mailbox, an asterisk appears after the display of the number of users. The display is normally in reverse video (if your terminal supports this in the status line) and is right-justified to reduce distraction. Every fifth display is done in normal video to give the screen a chance to rest.

If you have a file named **.who** in your home directory, then the contents of that file is printed first. One common use of this feature is to alias the **chdir**, **pushd**, and **popd** commands to place the current directory stack in **/.who** after it changes the new directory.

If you have a file named **.syslinelock** in your home directory, then the **sysline** command will not update its statistics and write on your screen, it will just go to sleep for a minute. This is useful if you want to momentarily disable **sysline**. Note that it may take a few seconds from the time the lock file is created until you are guaranteed that **sysline** will not write on the screen.

Flags

Item	Description
-b	Beeps once every half hour and twice every hour.
-c	Clears the status line for five seconds before each redisplay.
-D	Prints out the current day/date before the time.
-d	Prints status line data in human readable format, debug mode.
-e	Prints out only the information. Suppresses the control commands necessary to put the information on the bottom line. This option is useful for putting the output of the sysline command onto the mode line of an emacs window.
-H Remote	Prints the load average on the remote host <i>Remote</i> . If the host is down, or is not sending <i>rwhod</i> packets, then the down time is printed instead. If the prefix ucb is present, then it is removed.
-h	Prints out the host machine's name after the time.
-i	Prints out the process ID of the sysline command process onto standard output upon startup. With this information you can send the alarm signal to the sysline process to cause it to update immediately. The sysline command writes to the standard error, so you can redirect the standard output into a file to catch the process ID.
-j	Left-justifies the sysline command output on terminals capable of cursor movement on the status line.
-l	Suppresses the printing of names of people who log in and out.
-m	Suppresses mail check.
+N	Updates the status line every <i>N</i> seconds. The default is 60 seconds.
-p	Suppresses the report of the number of processes that are executable and suspended.
-q	Suppresses the printout diagnostic messages if something goes wrong when starting up.
-r	Suppresses reverse video display.
-s	Prints the short form of a line by left-justifying iff (if and only if) escapes are not allowed in the status line. Some terminals (the Televideos and Freedom 100 for example) do not allow cursor movements (or other "intelligent" operations) in the status line. For these terminals, the sysline command normally uses blanks to cause right-justification. This flag disables the adding of blanks.
-w	Prints the status on the current line of the terminal, suitable for use inside a one line window (Window mode).

Examples

To display the day and date, the number of processes which may be run, the number of users, and to clear the screen five seconds before it updates, enter:

```
sysline -Dcr
```

Note: This will only work on screens which have status line capabilities.

Files

Item	Description
/etc/utmp	Contains the names of users who are logged in.
/dev/kmem	Contains the process table.
/var/spool/rwho/whod.*	Contains who/Uptime information for remote hosts.
\${HOME}/.who	Specifies information to print on the bottom line.
\${HOME}/.syslinelock	Specifies that when it exists, sysline does not print.

syslogd Daemon

Purpose

Logs system messages.

Syntax

```
syslogd [-a] [ -d ] [ -s ] [ -f ConfigurationFile ] [ -m MarkInterval ] [ -r ] [ -R ] [ -n ] [ -N ] [ -p LogName ]  
[ -M all ] [ -A AdditionalLog ] [-e]
```

Description

The **syslogd** daemon reads a datagram socket and sends each message line to a destination described by the **/etc/syslog.conf** [configuration file](#). The **syslogd** daemon reads the configuration file when it is activated and when it receives a hangup signal.

The **syslogd** daemon creates the **/etc/syslog.pid** file, which contains a single line with the command process ID used to end or reconfigure the **syslogd** daemon.

A terminate signal sent to the **syslogd** daemon ends the daemon. The **syslogd** daemon logs the end-signal information and terminates immediately.

Each message is one line. A message can contain a priority code, marked by a digit enclosed in < > (angle braces) at the beginning of the line. Messages longer than 900 bytes may be truncated.

The **/usr/include/sys/syslog.h** include file defines the facility and priority codes used by the configuration file. Locally written applications use the definitions contained in the **syslog.h** file to log messages via the **syslogd** daemon.

Note: The maximum file size for the **syslogd** log file cannot exceed 2GB.

Flags

-a

Suppresses the reverse host name lookup for the messages coming from the remote host and logs the IP address of the remote host in the log files.

-d

Turns on debugging.

-e

Specifies enhanced rotation. All compressed and uncompressed files that are available in the log directory and that are created by the **syslogd** daemon are considered for rotation.

-f ConfigurationFile

Specifies an alternate configuration file.

-m MarkInterval

Specifies the number of minutes between the **mark** command messages. If you do not use this flag, the **mark** command sends a message with **LOG_INFO** priority sent every 20 minutes. This facility is not enabled by a **selector** field containing an asterisk (*), which selects all other facilities.

-M all

Specifies not to suppress duplicate messages in logfile. This flag is valid only if used with the **all** argument.

-s

Specifies to forward a "shortened" message to another system (if it is configured to do so) for all the forwarding syslog messages generated on the local system.

-r

Suppresses logging of messages received from remote hosts.

-R

Disables the facility to receive messages from the network using the internet domain socket.

-n

Suppresses the "Message forwarded from <log_host_name>: " string added to the beginning of the syslog message that is forwarded to a remote log host.

-N

Suppresses logging of priority and facility information for each log message.

-p

Specifies an alternate path name for the datagram socket.

-A AdditionalLog

Specifies additional logs that the syslogd daemon checks. By default, the syslogd daemon checks the `/dev/log` file for messages. If this flag is specified, it also checks the additional files for messages. The additional logs might be in the **chroot** path.

Configuration File

The configuration file informs the **syslogd** daemon where to send a system message, depending on the message's priority level and the facility that generated it.

If you do not use the **-f** flag, the **syslogd** daemon reads the default configuration file, the `/etc/syslog.conf` file.

The **syslogd** daemon ignores blank lines and lines beginning with a number sign (#).

Format

Lines in the configuration file for the **syslogd** daemon contain a **selector** field, an **action** field, and an optional **rotation** field, separated by one or more tabs or spaces.

The **selector** field names a facility and a priority level. Separate facility names with a , (comma). Separate the facility and priority-level portions of the **selector** field with a . (period). Separate multiple entries in the same selector field with a ; (semicolon). To select all facilities, use an * (asterisk).

The **action** field identifies a destination (file, host, or user) to receive the messages. If routed to a remote host, the remote system will handle the message as indicated in its own configuration file. To display messages on a user's terminal, the **destination** field must contain the name of a valid, logged-in system user.

The **rotation** field identifies how rotation is used. If the **action** field is a file, then rotation can be based on size or time, or both. One can also compress and/or archive the rotated files.

Facilities

Use the following system facility names in the **selector** field:

Facility	Description
kern	Kernel
user	User level
mail	Mail subsystem
daemon	System daemons
auth	Security or authorization
syslog	syslogd daemon
lpr	Line-printer subsystem
news	News subsystem
uucp	uucp subsystem
local0 through local7	Local use
*	All facilities

Priority Levels

Use the following message priority levels in the **selector** field. Messages of the specified priority level and all levels above it are sent as directed.

Priority	Description
emerg	Specifies emergency messages (LOG_EMERG). These messages are not distributed to all users. LOG_EMERG priority messages can be logged into a separate file for reviewing.
alert	Specifies important messages (LOG_ALERT), such as a serious hardware error. These messages are distributed to all users.
crit	Specifies critical messages not classified as errors (LOG_CRIT), such as improper login attempts. LOG_CRIT and higher-priority messages are sent to the system console.
err	Specifies messages that represent error conditions (LOG_ERR), such as an unsuccessful disk write.
warning	Specifies messages for abnormal, but recoverable, conditions (LOG_WARNING).
notice	Specifies important informational messages (LOG_NOTICE). Messages without a priority designation are mapped into this priority message.
info	Specifies informational messages (LOG_INFO). These messages can be discarded, but are useful in analyzing the system.
debug	Specifies debugging messages (LOG_DEBUG). These messages may be discarded.
none	Excludes the selected facility. This priority level is useful only if preceded by an entry with an * (asterisk) in the same selector field.

Destinations

Use the following message destinations in the **action** field.

File Name

Full path name of a file opened in append mode

@Host

Host name, preceded by the at sign (@)

User[, User][...]

User names

All users

centralizedlog LogSpaceName/LogStreamName

PowerHA® pureScale logstream

Note: You must have PowerHA pureScale appliance to use the *centralizedlog LogSpaceName/LogStreamName* message destination.

Rotation

Use the following rotation keywords in the **rotation** field.

rotate

This keyword must be specified after the **action** field.

size

This keyword specifies that rotation is based on size. It is followed by a number and either a **k** (kilobytes) or **m**(megabytes).

time

This keyword specifies that rotation is based on time. It is followed by a number and either a **h**(hour) or **d**(day) or **w**(week) or **m**(month) or **y**(year).

files

This keyword specifies the total number of rotated files. It is followed by a number. If not specified, then there are unlimited number of rotated files.

compress

This keyword specifies that the saved rotated files will be compressed.

archive

This keyword specifies that the saved rotated files will be copied to a directory. It is followed by the directory name.

Effect of command line flags on syslogd rotation:**The -e flag:**

This flag is used to enhance the **syslogd** rotation policy. When this flag is used, all the compressed and uncompressed files are considered during rotation.

If your log file rotation frequency is only determined by time, you can reset the timer by entering the following command:

```
refresh -s syslogd
```

The next rotation that is based on the time of the previous rotation does not occur when this command is run during the scheduled time interval.

Examples

1. To log all mail facility messages at the debug level or above to the file **/tmp/maileyslog**, enter the following command:

```
mail.debug /tmp/maileyslog
```

2. To send all system messages except those from the mail facility to a host named **rigil**, enter the following command:

```
*.debug;mail.none @rigil
```

3. To send messages at the **emerg** priority level from all facilities, and messages at the **crit** priority level and above from the mail and daemon facilities, to users nick and jam, enter the following command:

```
*.emerg;mail,daemon.crit nick, jam
```

4. To send all mail facility messages to all users' terminal screens, enter the following command:

```
mail.debug *
```

5. To log all facility messages at the debug level or above to the file **/tmp/syslog.out**, and have the file rotated when it gets larger than 500 kilobytes or if a week passes, limit the number of rotated files to 10, use compression and also use **/syslogfiles** as the archive directory, enter the following command:

```
*.debug /tmp/syslog.out rotate size 500k time 1w files 10 compress archive /syslogfiles
```

6. To set the rotation schedule for the **syslog.out** file to rotate only every five days, enter the following command:

```
*.debug /var/log/syslog.out rotate time 5d
```

You can reset the timer at any time before the next rotation by entering the following command:

```
refresh -s syslogd
```

After you reset the timer, the next rotation occurs after the scheduled interval of time that starts at the time when the refresh command is entered.

Files

/etc/syslog.conf

Controls the output of **syslogd**.

/etc/syslog.pid

Contains the process ID.

t

The following AIX commands begin with the letter *t*.

tab Command

Purpose

Changes spaces into tabs.

Syntax

tab [-e] [*File ...*]

Description

The **tab** command reads the file specified by the *File* parameter or standard input, and replaces spaces in the input with tab characters wherever the **tab** command can eliminate one or more spaces. If you specify a file with the *File* parameter, the **tab** command writes the resulting file back to the original file. If the input is standard input, the **tab** command writes to standard output. The **tab** command assumes that tab stops are set every eight columns, starting with column nine. The file name specified for the *File* parameter cannot exceed **PATH_MAX**-9 bytes in length.

Flag

Item	Description

-e	Replaces only those spaces at the beginning of a line up to the first non-space character.

Example

To replace space characters in the *File* file with tab characters, enter:

```
tab File
```

File

Item	Description
/usr/bin/tab	Contains the tab command.

tabs Command

Purpose

Sets tab stops on terminals.

Syntax

tabs [*TabSpec ...*] [+m [*Number*]] [-T *Terminal ...*]

Description

The **tabs** command specifies tab stops on terminals that support remotely settable hardware tab characters. Tab stops are set according to the *TabSpec* parameter, and previous settings are erased.

When you use the **tabs** command, always refer to the leftmost column number as 1, even if your workstation refers to it as 0.

If you do not specify the *TabSpec* parameter, the default value is **-8**.

The following preset formats can be specified for the *TabSpec* parameter:

Item	Description

-a	Sets the tabs to 1, 10, 16, 36, and 72 (IBM System/370 Assembler first format).

-a2	Sets the tabs to 1, 10, 16, 40, and 72 (IBM System/370 Assembler second format).

-c	Sets the tabs to 1, 8, 12, 16, 20, and 55 (COBOL normal format).

-c2	Sets the tabs to 1, 6, 10, 14, and 49 (COBOL compact format, columns 1-6 omitted). With this code, the first column position corresponds to card column 7. One space gets you to column 8, and a tab gets you to column 12. Files using this code should include a format specification of:

```
<:t-c2 m6 s66 d:>
```

-c3	Sets the tabs to 1, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58, 62, and 67 (COBOL compact format with more tabs than -c2). These tabs provide the recommended format for COBOL. Files using this code should include a format specification of:

```
<:t-c3 m6 s66 d:>
```

-f	Sets the tabs to 1, 7, 11, 15, 19, and 23 (FORTRAN).

-p	Sets the tabs to 1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, and 61 (PL/I).

-s	Sets the tabs to 1, 10, and 55 (SNOBOL).

-u	Sets the tabs to 1, 12, 20, and 44.

In addition to the preset formats, the *TabSpecs* parameter can include:

Item	Description
-Number	Sets regularly repeating tabs at every <i>Number</i> column. (The standard operating system tab setting is -8 . The -8 setting is required when using the nroff command with the -h flag.) Another special case is the -0 setting, which implies no tabs at all. If more than 20 tabs are set, you must run the tabs command twice to clear them.
<i>Number1, Number2,...</i>	Sets tabs at the specified column numbers (a comma-separated list in ascending order). You can specify up to 40 numbers. If any number except the first has a plus-sign prefix, the prefixed number is added to the previous number for the next setting. Thus, the tab list specified by 1,10,20,30 provides the same tab settings as the tab list specified by 1,10,+10,+10 .
-Filep	Reads the first line of the <i>Filep</i> file for a format specification. If the tabs command finds a format specification, the tabs command sets tabs as specified. If the tabs command does not find a format specification, it sets tabs to the system default (-8).

It is sometimes convenient to maintain text files with nonstandard tab stop settings (tab stops that are not set at every eighth column). Such files must be converted to a standard format. This is often done by replacing all tab characters with the appropriate number of space characters, before they can be processed by any commands. A format specification occurring in the first line of a text file specifies how tab characters are to be expanded in the remainder of the file.

A format specification consists of a sequence of parameters separated by blanks and surrounded by **<:** and **:>**. Each parameter consists of a letter key, possibly followed immediately by a value. The following parameters are recognized:

Item	Description						
<i>ttabs</i>	Specifies the tab stop settings for a file. The value of <i>tabs</i> must be one of the following: • A list of column numbers separated by commas, indicating tab stops set at the specified columns. • A - (dash) followed immediately by an integer <i>n</i>, indicating tab stops set at intervals of <i>n</i> columns, that is, at 1+<i>n</i>, 1+2*<i>n</i>, and so on. • A - (dash) followed by the name of a preset tab stop specification. Up to 40 numbers are allowed in a comma-separated list of tab stop settings. If any number (except the first one) is preceded by a plus sign, it is taken as an increment to be added to the previous value. Therefore, the formats t1, 10, 20, 30 and t1, 10, +10, +10 are considered identical. Standard tab stops are specified by t-8, or, equivalently, t1, 9, 17, 25. This is the tab stop setting that most system utilities assume, and is the most likely setting to find at a terminal. The specification t-0 specifies no tab stops at all. The preset tab stop specifications that are recognized are as follow: a 1, 10, 16, 36, 72 Assembler, IBM System/370, first format a2 1, 10, 16, 40, 72 Assembler, IBM System/370, second format c 1, 8, 12, 16, 20, 55 COBOL, normal format c2 1, 6, 10, 14, 49 COBOL compact format (columns 1-6 omitted). Using this code, the first typed character corresponds to card column 7; one space gets you to column 8; and a tab gets you to column 12. Files using this tab stop setup should include a format specification as follows: <pre><:t-c2 m6 s66 d:></pre> c3 1, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58, 62, 67 COBOL compact format (columns 1-6 omitted) with more tab stops than c2. This is the recommended format for COBOL. The appropriate format specification is: <pre><:t-c3 m6 s66 d:></pre> f 1, 7, 11, 15, 19, 23 FORTRAN p 1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61 PL/I s 1, 10, 55 SNOBOL u 1, 12, 20, 44 UNIVAC 1100 Assembler <tr> <td><i>ssize</i></td><td>Specifies a maximum line size. The value of <i>size</i> must be an integer. Size checking is performed after tab characters have been expanded, but before the margin is adjusted.</td></tr> <tr> <td><i>mmargin</i></td><td>Specifies the number of space characters to be added to the beginning of each line. The value of <i>margin</i> must be an integer.</td></tr> <tr> <td><i>d</i></td><td>Indicates that the line containing the format specification is to be deleted from the converted file. The <i>d</i> parameter takes no value.</td></tr>	<i>ssize</i>	Specifies a maximum line size. The value of <i>size</i> must be an integer. Size checking is performed after tab characters have been expanded, but before the margin is adjusted.	<i>mmargin</i>	Specifies the number of space characters to be added to the beginning of each line. The value of <i>margin</i> must be an integer.	<i>d</i>	Indicates that the line containing the format specification is to be deleted from the converted file. The <i>d</i> parameter takes no value.
<i>ssize</i>	Specifies a maximum line size. The value of <i>size</i> must be an integer. Size checking is performed after tab characters have been expanded, but before the margin is adjusted.						
<i>mmargin</i>	Specifies the number of space characters to be added to the beginning of each line. The value of <i>margin</i> must be an integer.						
<i>d</i>	Indicates that the line containing the format specification is to be deleted from the converted file. The <i>d</i> parameter takes no value.						

Item	Description
<i>e</i>	Indicates that the current format is valid only until another format specification is encountered in the file. The <i>e</i> parameter takes no value.

Default values, which are assumed for parameters not supplied, are **t-8** and **m0**. If the *s* parameter is not specified, no size checking is performed. If the first line of a file does not contain a format specification, the above defaults are assumed for the entire file. The following is an example of a line containing a format specification:

```
<:t5,10,15 s72:>
```

If a format specification can be disguised as a comment, it is not necessary to code the *d* parameter.

Flags

Item	Description
-T <i>Terminal</i>	Identifies the terminal so the tabs command can set tabs and margins correctly. The <i>Terminal</i> variable is one of the terminals specified in the greek command. Supported values for the <i>Terminal</i> variable include: ANSI Any ANSI terminal, such as a VT100 terminal. hp Hewlett-Packard hardcopy terminals. 2621 Hewlett-Packard 2621. 2640 Hewlett-Packard 2640. 2645 Hewlett-Packard 2645. Additional hardcopy terminals supported by the tabs command include: • 1620 • 1620-12 • 1620-12-8 • 1700 • 1700-12 • 1700-12-8 • 300 • 300-12 • 300s • 300s-12 • 40-2 • 4000a • 4000a-12 • 43 • 450 • 450-12 • 450-12-8 • tn1200 • tn300 • oki If you do not provide the -T flag, the value of the environment variable TERM is used. If the -T flag is provided with no value or if -T and TERM have invalid values, the error message <code>unknown terminal</code> is displayed and the command terminates.
+m <i>Number</i>	Moves all tabs to the right the number of columns specified by the <i>Number</i> variable. This flag also sets the left margin to the column specified by the <i>Number</i> variable. If m is specified without a value, the default value for the <i>Number</i> variable is 10. The leftmost margin on most workstations is defined by +m0. The first column for tabs is defined as column 0 not column 1. Note: If the same flag occurs more than once, only the last flag takes effect.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Examples

1. To set tabs every four spaces, enter:

```
tabs -4
```

2. To set tabs every ten spaces on a VT100 terminal, enter:

```
tabs -10 -TANSI
```

File

Item	Description
/usr/bin/tabs	Contains the tabs command.

tail Command

Purpose

Displays the last few lines of a file.

Syntax

Standard Syntax

```
tail [ -f ] [ -c Number | -n Number | -m Number | -b Number | -k Number ] [ File ]
```

To Display Lines in Reverse Order

```
tail [ -r ] [ -n Number ] [ File ]
```

Description

The **tail** command writes the file specified by the *File* parameter to standard output beginning at a specified point. If no file is specified, standard input is used. The *Number* variable specifies how many units to write to standard output. The value for the *Number* variable can be a positive or negative integer. If the value is preceded by + (plus sign), the file is written to standard output starting at the specified number of units from the beginning of the file. If the value is preceded by - (minus sign), the file is written to standard output starting at the specified number of units from the end of the file. If the value is not preceded by + (plus sign) or - (minus sign), the file is read starting at the specified number of units from the end of the file.

The type of unit used by the *Number* variable to determine the starting point for the count is determined by the **-b**, **-c**, **-k**, **-m**, or **-n** flag. If one of these flags is not specified, the **tail** command reads the last ten lines of the specified file and writes them to standard output. This is the same as entering **-n 10** at the command line.

The **-m** flag provides consistent results in both single- and double-byte character environments. The **-c** flag should be used with caution when the input is a text file containing multibyte characters, because output can be produced that does not start on a character boundary.

Flags

Item	Description
-b <i>Number</i>	Reads the specified file beginning at the 512-byte block location indicated by the <i>Number</i> variable.
-c <i>Number</i>	Reads the specified file beginning at the byte location indicated by the <i>Number</i> variable.
-f	If the input file is a regular file or if the <i>File</i> parameter specifies a FIFO (first-in-first-out), the tail command does not terminate after the last specified unit of the input file has been copied, but continues to read and copy additional units from the input file as they become available. If no <i>File</i> parameter is specified and standard input is a pipe, the -f flag is ignored. The tail -f command can be used to monitor the growth of a file being written by another process.
-k <i>Number</i>	Reads the specified file beginning at the 1KB block location indicated by the <i>Number</i> variable.
-m <i>Number</i>	Reads the specified file beginning at the multibyte character location indicated by the <i>Number</i> variable. Using this flag provides consistent results in both single- and double-byte character-code-set environments.
-n <i>Number</i>	Reads the specified file from the first or last line location as indicated by the sign (+ or - or none) of the <i>Number</i> variable and offset by the number of lines <i>Number</i> .
-r	Displays the output from the end of the file in reverse order. The default for the -r flag prints the entire file in reverse order. If the file is larger than 20,480 bytes, the -r flag displays only the last 20,480 bytes.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Examples

1. To display the last 10 lines of the notes file, enter:

```
tail notes
```

2. To specify the number of lines to start reading from the end of the notes file, enter:

```
tail -n 20 notes
```

3. To display the notes file a page at a time, beginning with the 200th byte, enter:

```
tail -c +200 notes | pg
```

4. To follow the growth of a file, enter:

```
tail -f accounts
```

This displays the last 10 lines of the accounts file. The **tail** command continues to display lines as they are added to the accounts file. The display continues until you press the Ctrl-C key sequence to stop it.

File

Item	Description
<code>/usr/bin/tail</code>	Contains the tail command.

talk Command

Purpose

Converse with another user.

Syntax

talk {*User* | *User@Host* | *Host!User* | *Host.User* | *Host:User*} [*Tty*] [*Pty*]

Description

The `/usr/bin/talk` command allows two users on the same host or on different hosts to have an interactive conversation. The **talk** command opens both a send window and a receive window on each user's display. Each user is then able to type into the send window while the **talk** command displays what the other user is typing.

To initiate a conversation, a local user executes the **talk** command and specifies a remote user's login ID. The remote user's login ID can contain NLS characters. If the remote user is on a remote host, the name of the host must also be specified in one of the following ways:

```
User@Host
Host!User
Host.User
Host:User
```

When using full domain names, the only valid form for specifying the user and host is *User@Host*. For example, `michael@host17.dev.ibm.com` initiates a conversation with user `michael` at host `host17` in the `dev.ibm.com` domain.

When the local user initiates the conversation, a message is sent to the remote user, inviting a conversation. If the local user also specifies `TTY`, the invitation message is sent only to the specified terminal. Otherwise, the invitation is sent to the remote user's login terminal. This usually is the console, but it may be another terminal. Once this invitation is received, the **talk** command displays two windows on the local user's terminal and displays progress messages until the remote user responds to the invitation.

Note: If the remote user is running AIXwindows and has no other terminals open, the **talk** command cannot send an invitation.

To have the conversation, the remote user also has to execute the **talk** command from any terminal and specify the local user's account name and host name, if appropriate. When the remote user accepts the invitation, the **talk** command displays two windows on each user's terminal. One window displays what is typed by the local user; the other window displays what is typed by the remote user. To end the conversation, either user can press the Interrupt (Ctrl-C) key sequence and the connection is closed. The Interrupt key sequence can be displayed and modified using the **stty** command.

If the users involved in the conversation are using National Language Support (NLS) capabilities, their terminals must support the printing of NLS characters. The same is true for conversations using Kanji capabilities; the terminals being used must support the printing of Kanji characters.

The **talk** command requires a valid address to which to bind. The host name of the remote machine must be bound to a working network interface, which is usable by other network commands, such as the **ping** command. If a machine has no network interface, that is a standalone machine, it must bind its host name to the loopback address (127.0.0.1) in order for the **talk** command to work. For example, two

users named `local` and `remote` on a standalone machine could initiate a conversation, using the **talk** command, by entering:

```
talk remote@loopback
```

To which user `remote` responds:

```
talk local@loopback
```

To disallow **talk** command invitations, the remote user can issue the **mesg** command.

Note: The **talk** command uses the Talk 4.3 protocol.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To talk to a user logged in on a remote host, enter:

```
talk dale@host2
```

In this example, the local user wants to talk with user `dale` who is logged in on `host2`.

2. To talk to a user only if that user is logged in on the console of a remote host, enter:

```
talk dale@host2 console
```

User `dale` receives this message only if logged in on the console at `host2`.

talkd Daemon

Purpose

Provides the server function for the **talk** command.

Syntax

/usr/sbin/talkd [**-s**]

Description

Note: The **talkd** daemon is normally started by the **inetd** daemon. It can also be controlled from the command line, using SRC commands.

The **/usr/sbin/talkd** daemon is the server that notifies a user (the recipient) that another user (the caller) wants to initiate a conversation. The daemon sets up the conversation if the recipient accepts the invitation. The caller initiates the conversation by executing the **talk** command specifying the recipient. The recipient accepts the invitation by executing the **talk** command specifying the caller.

The **talkd** daemon listens at the socket defined in the **/etc/services** file. When the **talkd** daemon receives a LOOK_UP request from a local or remote **talk** process, the **talkd** daemon scans its internal invitation table for an entry that pairs the client process (the local or remote **talk** process) with a caller.

If no entry exists in the invitation table, the **talkd** daemon assumes that the client process is the caller. The **talkd** daemon then receives the client process' ANNOUNCE request. The **talkd** daemon broadcasts

an invitation on the remote computer where the recipient first logged in (unless the caller specifies a particular tty device). This terminal usually is the console, but it may be another terminal.

Otherwise, the invitation is sent to the terminal that the second user first logged in to. This usually is the console, but it may be another terminal.

If an entry does exist in the **talkd** daemon's internal invitation table, the **talkd** daemon assumes that the client is the recipient. The **talkd** daemon returns the appropriate rendezvous address to the **talk** process for the recipient. The recipient process then establishes a stream connection with the caller process.

Note: The **talkd** daemon uses the Talk 4.3 protocol. The subserver name for the AIX protocol is **ntalk**.

Changes to the **talkd** daemon can be made using the System Management Interface Tool (SMIT) or System Resource Controller (SRC), by editing the **/etc/inetd.conf** or **/etc/services** file. Entering **talkd** at the command line is not recommended. The **talkd** daemon is started by default when it is uncommented in the **/etc/inetd.conf** file.

The **inetd** daemon gets its information from the **/etc/inetd.conf** file and the **/etc/services** file.

After changing the **/etc/inetd.conf** or **/etc/services** file, run the **refresh -s inetd** or **kill -1 InetdPID** command to inform the **inetd** daemon of the changes to its configuration file.

Debugging messages are sent to the **syslogd** daemon.

Note: The **talkd** daemon should be controlled using the System Management Interface Tool (SMIT) or by changing the **/etc/inetd.conf** file.

Manipulating the talkd Daemon with the System Resource Controller

The **talkd** daemon is a subserver of the **inetd** daemon, which is a subsystem of the System Resource Controller (SRC). The **talkd** daemon is a member of the **tcpip** SRC subsystem group. This daemon is enabled by default in the **/etc/inetd.conf** file and can be manipulated by the following SRC commands:

Item	Description
<u>startsrc</u>	Starts a subsystem, group of subsystems, or a subserver.
<u>stopsrc</u>	Stops a subsystem, group of subsystems, or a subserver.
<u>lssrc</u>	Gets the status of a subsystem, group of subsystems, or a subserver.

Flags

Item	Description

-s	Turns on socket-level debugging.

Examples

1. To start the **talkd** daemon, enter the following:

```
startsrc -t ntalk
```

This command starts the **talkd** subserver.

2. To stop the **talkd** daemon normally, enter the following:

```
stopsrc -t ntalk
```

This command allows all pending connections to start and existing connections to complete but prevents new connections from starting.

3. To force stop the **talkd** daemon and all **talkd** connections, enter the following:

```
stopsrc -f -t ntalk
```

This command terminates all pending connections and existing connections immediately.

4. To display a short status report about the **talkd** daemon, enter the following:

```
lssrc -t ntalk
```

This command returns the daemon's name, process ID, and state (active or inactive).

Files

Item	Description

<code>/etc/utmp</code>	Contains data about users currently logged in.

tapechk Command

Purpose

Performs consistency checking on the streaming tape device.

Syntax

```
tapechk [ -? ] Number1 Number2
```

Description

The **tapechk** command performs rudimentary consistency checking on an attached streaming tape device. Some hardware malfunctions of a streaming tape drive can be detected by simply reading a tape. The **tapechk** command provides a way to perform tape reads at the file level.

Because the streaming tape drive cannot backspace over physical data blocks or files, the **tapechk** command rewinds the tape to its starting position prior to each check. This command either checks data for the next number of files specified by the *Number1* parameter or skips the next number of files specified by the *Number2* parameter. If you do not specify any parameters, the **tapechk** command rewinds the tape and checks only the first physical block.

The **tapechk** command uses the device in the **TAPE** environment variable if it is defined. Otherwise, the default tape device is `/dev/rmt0`.

Note: The **backup** command allows you to archive files selectively or as an entire file system. It writes data as a continuous stream terminated by a file mark, regardless of the number of files specified. The **tapechk** command perceives each stream of data as a single file, which is important when you specify numeric parameters.

Although you can use the **tapechk** command on any streaming tape cartridge, it is primarily designed for checking tapes written by the **backup** command.

Flag

Item	Description

<code>-?</code>	Explains the format of the tapechk command.

Note: If you specify the `-?` flag, it must be specified before the *Number1* and *Number2* parameters.

Exit Status

This command returns the following exit values:

Item	Description

0	Successful completion.
>0	An error occurred.

Example

To check the first three files on a streaming tape device, enter:

```
tapechk 3
```

File

Item	Description
<code>/usr/sbin/tapechk</code>	Contains the tapechk command.

tar Command

Purpose

Manipulates archives.

Syntax

X/Open Standards:

```
tar {-c|-r|-t|-u|-x} [-B] [-d] [-E] [-F] [--format=pax] [-h] [-i] [-l] [-m] [-o] [-p] [-s]
[-U] [-v] [-w]
    [-Number] [-f Archive] [-b Blocks]
    [-S [Feet] [Feet @Density] [Blocksb]] [-L InputList] [-X ExcludeList]
    [-N Blocks] [-R] [-D] [-C Directory] [-Z] File | Directory ...
```

Berkeley Standards:

```
tar {c|r|t|u|x}[bBdDEfFhiLlXmNopRsSUvwZ[0-9]]
    [ Blocks ] [ Archive ] [ InputList ] [ ExcludeFile ]
    [ [ Feet ] | [ Feet@Density ] ] [ Blocksb ] ] Directory | File ...
```

Description

Note:

1. The **ustar** header format allows unlimited ($2^{64}-1$) file sizes.
2. The **tar** command does not preserve the sparse nature of any file that is sparsely allocated. Any file that was originally sparse before the restoration will have all space allocated within the filesystem for the size of the file.

The **tar** command manipulates archives by writing files to, or retrieving files from an archive storage medium. The files used by the **tar** command are represented by the *File* parameter. If the *File* parameter refers to a directory, then that directory and recursively all files and directories within it are referenced as well.

The **tar** command looks for archives on the default device (usually tape), unless you specify another device with the **-f** *Archive* flag. When specifying path names that are greater than 100 characters for the United States Tape Archiver (USTAR) format, remember that the path name is composed of a prefix buffer, a / (slash), and a name buffer.

> Starting from AIX 7.3.1, the **tar** command can archive files in the portable archive exchange (PAX) format and can extract files from the tar file that is in the PAX format. You must specify the

--format=pax flag to generate a tar file in the PAX format. |<

The **tar** command supports the length of **path+filename** only till the system defined **PATH_MAX** limit. Any length of **path+filename** input greater than **PATH_MAX** limit is not archived

When writing to an archive, the **tar** command uses a temporary file (the **/tmp/tar*** file) and maintains in memory a table of files with several links. You receive an error message if the **tar** command cannot create the temporary file, or if there is not enough memory available to hold the link tables.

Two groups of flags exist for the **tar** command: the required flags and the optional flags. The required flags control the actions of the **tar** command and include the **-c**, **-r**, **-t**, **-u**, and **-x** flags. At least one required flag must be selected for the **tar** command to function. Having selected a required flag, you can select an optional flag but none are necessary to control the **tar** command.

Note:

1. When the storage device is an ordinary file or a block special file, the **-u** and **-r** flags backspace. However, raw magnetic tape devices do not support backspacing. So when the storage device is a raw magnetic tape, the **-u** and **-r** flags rewind the tape, open it, and then read it again.
2. Records are one block long on block magnetic tape, but they are typically less than half as dense on raw magnetic tape. As a result, although a blocked raw tape must be read twice, the total amount of tape motion is less than when reading one-block records from a block magnetic tape once.
3. The structure of a streaming tape device does not support the addition of information at the end of a tape. Consequently when the storage device is a streaming tape, the **-u** and **-r** flags are not valid options. An attempt to use these flags results in the following error message:

```
tar: Update and Replace options not valid for a
streaming tape drive.
```

4. No recovery exists from tape errors.
5. The performance of the **tar** command to the IBM9348 Magnetic Tape Unit Model 12 can be improved by changing the default block size. To change the block size, enter the following at the command line:

```
chdev -l <device_name> -a block_size=32k
```

For more information on using tape devices see the **rmt** special file.

Flags

Flags for the **tar** command are in two groups, the required and the optional. You must supply at least one required flag to control the **tar** command.

Table 25. Required Flags

Required Flags	Description
-c	Creates a new archive and writes the files specified by one or more <i>File</i> parameters to the beginning of the archive.
-r	Writes the files specified by one or more <i>File</i> parameters to the end of the archive. This flag is not valid for any tape devices because such devices do not support the addition of information at the end of a tape.
-t	Lists the files in the order in which they appear in the archive. Files can be listed more than once.

Table 25. Required Flags (continued)

Required Flags	Description
-u	Adds the files specified by one or more <i>File</i> parameters to the end of the archive only if the files are not in the archive already, or if they have been modified since being written to the archive. The -u flag is not valid for any tape devices because such devices do not support the addition of information at the end of a tape.
-U	Allows archival and extraction of Extended Attributes. The Extended Attributes include Access control list (ACL) also.
-x	Extracts the files specified by one or more <i>File</i> parameters from the archive. If the <i>File</i> parameter refers to a directory, the tar command recursively extracts that directory from the archive. If you do not specify the <i>File</i> parameter, the tar command extracts all of the files from the archive. When an archive contains multiple copies of the same file, the last copy extracted overwrites all previously extracted copies. If the file being extracted does not already exist on the system, the file is created. If you have the proper permissions, the tar command restores all files and directories with the same owner and group IDs as they have on the tape. If you do not have the proper permissions, the files and directories are restored with your owner and group IDs. It is not possible to ask for any occurrence of a file other than the last.

Table 26. Optional Flags

Optional Flags	Description
-B	Forces input and output blocking to 20 blocks per record. With this option, the tar command can work across communications channels where blocking may not be maintained.
-b Blocks	Specifies the number of 512 bytes blocks per record. Both the default and the maximum is 20, which is appropriate for tape records. Due to the size of interrecord gaps, tapes written with large blocking factors can hold much more data than tapes with only one block per record. The block size is determined automatically when tapes are read (the -x or -t function flags). When archives are updated with the -u and -r functions, the existing record size is used. The tar command writes archives using the specified value of the <i>Blocks</i> parameter only when creating new archives with the -c flag. For output to ordinary files with the -f flag, you can save disk space by using a blocking factor that matches the size of disk blocks (for example, the -b4 flag for 2048-byte disk blocks).

Table 26. Optional Flags (continued)

Optional Flags	Description
-C Directory -D -d -E -F	Causes the tar command to perform a chdir subroutine to the directory specified by the <i>Directory</i> variable. Using the -C flag allows multiple directories that are not related by a close common parent to be archived, using short relative path names. For example, to archive files from the /usr/include and /etc directories, you might use the following command: <pre>tar c -C /usr/include File1 File2 -C /etc File3 File4</pre> You can use multiple -C options when you extract files from the archive. When you use multiple -C options, each instance of the -C Directory is relative to the one that is listed before it in the command. For example, the second -C Directory is relative to the first -C Directory. If an archive contains a file with an absolute path name, for example /home/dir1/filename, the file is extracted into the directory that is specified by the -C Directory by removing the leading slash (/) from the filepath or filename. The -C Directory flag must appear after all other flags and can appear in the list of file names given. Suppress recursive processing when directories are specified. Makes separate entries for block files, special character files, and first-in-first-out (FIFO) piped processes. Normally, the tar command will not archive these special files. When writing to an archive with the -d flag, the tar command makes it possible to restore empty directories, special files, and first-in-first-out (FIFO) piped processes with the -x flag. Restriction: Although anyone can archive special files, only a user with root user authority can extract them from an archive (FIFO can also be extracted by non-root users). Avoids truncation of the long user and group names during addition of files to new or existing archive. Checks the file type before archiving. Source Code Control Systems (SCCS), Revision Control Systems (RCS), files named core, errs, a.out, and files ending in .o (dot o) are not archived.

Table 26. Optional Flags (continued)

Optional Flags	Description
-f <i>Archive</i>	Uses the <i>Archive</i> variable as the archive to be read or written. When this flag is not specified, the tar command uses a system-dependent default file name of the form /dev/rmt0 . If the <i>Archive</i> variable specified is - (minus sign), the tar command writes to standard output or reads from standard input. If you write to standard output, the -c flag must be used.
> > --format=pax	Creates an archive file in the PAX/POSIX format. If you do not specify this flag, by default, the tar file is created in the USTAR/GNU format. < <
-h	Forces the tar command to follow symbolic links as if they were normal files or directories. Normally, the tar command does not follow symbolic links.
-i	Ignores header checksum errors. The tar command writes a file header containing a checksum for each file in the archive. When this flag is not specified, the system verifies the contents of the header blocks by recomputing the checksum and stops with a directory checksum error when a mismatch occurs. When this flag is specified, the tar command logs the error and then scans forward until it finds a valid header block. This permits restoring files from later volumes of a multi-volume archive without reading earlier volumes.
-L <i>InputList</i>	The <i>Inputlist</i> argument to the -L option should always be the name of the file that lists the files and directories that need to be archived or extracted.
-l	Writes an error message to standard output for each file with a link count greater than 1 whose corresponding links were not also archived. For example, if file1 and file2 are hard-linked together and only file1 is placed on the archive, then the -l flag will issue an error message. Error messages are not displayed if the -l flag is not specified.
-m	Uses the time of extraction as the modification time. The default is to preserve the modification time of the files.
-N <i>Blocks</i>	Allows the tar command to use very large clusters of blocks when it deals with streaming tape archives. Note however, that on input, the tar command cannot automatically determine the block size of tapes with very long block sizes created with this flag. In the absence of a -N <i>Blocks</i> flag, the largest block size that the tar command can automatically determine is 20 blocks.

Table 26. Optional Flags (continued)

Optional Flags	Description
-o -p -R -s -S Blocks b, -S Feet, -S Feet@Density -U	Provides backwards compatibility with older versions (non-AIX) of the tar command. When this flag is used for reading, it causes the extracted file to take on the User and Group ID (UID and GID) of the user running the program, rather than those on the archive. This is the default behavior for the ordinary user. Restores fields to their original modes, ignoring the present umask. The setuid, setgid, and tacky bit permissions are also restored to the user with root user authority. This flag restores files and directories to their original mode. Use recursion when directories are specified. Ignored when used with the -D option. Tries to create a symbolic link If the tar command is unsuccessful in its attempt to link (regular link) two files with the -s flag. Specifies the number of 512 KB blocks per volume (first format), independent of the tape blocking factor. You can also specify the size of the tape in feet by using the second form, in which case the tar command assumes a default <i>Density</i> variable. The third form allows you to specify both tape length and density. Feet are assumed to be 11 inches long to be conservative. This flag lets you deal more easily with multivolume tape archives, where the tar command must be able to determine how many blocks fit on each volume. Note: 1. Tape drives vary in density capabilities. The <i>Density</i> variable calculates the amount of data a system can fit on a tape. 2. When using 1/4-inch tape devices, be sure to take into account the number of tracks on the tape device when specifying the value for the <i>Feet</i> variable. For example, a 4-track, 1/4-inch tape drive with a 600-foot tape and a density of 8000 bpi can be specified using the -S Feet@Density flag as follows: <pre data-bbox="899 1604 1065 1638">-S 2400@8000</pre> where 600 feet multiplied by 4 tracks equals 2400 feet. Archives or restores named extended attributes and ACLs. When listing, this option will display the names of any named extended attributes and the type of any ACLs associated with each file that are part of the archive image.

Table 26. Optional Flags (continued)

Optional Flags	Description
-v	Lists the name of each file as it is processed. With the -t flag, -v gives more information about the tape entries, including file sizes, times of last modification, User Number (UID), Group Number (GID), and permissions.
-w	Displays the action to be taken, followed by the file name, and then waits for user confirmation. If the response is affirmative, the action is performed. If the response is not affirmative, the file is ignored.
-Number	Uses the /dev/rmtNumber file instead of the default. For example, the -2 flag is the same as the -f/dev/rmt2 file.
-X ExcludeList	Excludes the file names or directories given in the <i>ExcludeList</i> from the tar archive being created, extracted or listed. The <i>ExcludeList</i> shall contain only one filename or directory per line which are to be excluded from the tar archive being created, extracted from or listed. The -X option can be specified multiple times and it takes precedence over all other options.
-Z	Archives the Encrypted File System (EFS) information of encrypted files or directories. The EFS information is extracted by default. When you specify the -t and -v flags along with the -Z flag, an e indicator is displayed after the file mode for encrypted files and directories that were archived with the -Z flag, and a hyphen (-) is displayed after the file mode for other files. Restriction: Archives created with the -Z flag can be restored only on AIX 6.1 or later releases.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To write the `file1` and `file2` files to a new archive on the default tape drive, enter:

```
tar -c file1 file2
```

2. To extract all files in the /tmp directory from the archive file on the /dev/rmt2 tape device and use the time of extraction as the modification time, enter:

```
tar -xm -f/dev/rmt2 /tmp
```

3. To create a new archive file that contains the file1 file and pass the archive file to the **dd** command to be written to the /dev/rmt1 device, enter:

```
tar -cvf - file1 | dd of=/dev/rmt1 conv=sync
```

4. To display the names of the files in the out.tar disk archive file on the current directory, enter:

```
tar -vtf out.tar
```

5. To expand the compressed **tar** archive file, fil.tar.z, pass the file to the **tar** command, and extract all files from the expanded **tar** archive file, enter:

```
zcat fil.tar.Z | tar -xvf -
```

6. To archive the contents of /usr/include and /usr/bin files using short relative path names, enter:

```
cd /usr  
tar -cvf/dev/rmt0 -C./include . -C ../bin .
```

Requirement: When specifying multiple instances of the **-C** flag with relative path names, the user must take the previous **-C** flag request into account.

7. To archive to an 8-mm device when using the **-S** flag, enter:

```
tar -cvf /dev/rmt0 -S 4800000b /usr
```

Restriction: When archiving to an 8-mm device, avoid using the **-S Feet** and **-S Feet@Density** flags, because the 8-mm device does not use the concept of density when writing to a tape.

8. To archive a list of all C files that is listed in the file through the *InputList* argument of the **-L** option, enter:

```
tar -cvf fl.tar -L fl_list
```

Where *fl_list* is a file consisting a list of all .c files in it. This can be obtained as follows:

```
ls *.c > fl_list
```

9. To archive a list of all C files by setting a variable using the **-L** option, enter:

```
ls *.c > fl_list  
fl=fl_list  
tar -cvf var.tar -L $fl
```

10. To avoid the truncation of long user or group names during creation of the archive, enter:

```
tar -cvEf file.tar file
```

11. To create a new archive file that contains the file1 file with ACL and EA, enter:

```
tar -cvUf /tmp/tar.ar file1
```

12. **>** To create a new PAX format archive file that contains the file1 file, enter the following command:

```
tar -c --format=pax -f result.tar file1
```

<

Berkeley Options

The following are examples of the Berkeley options using the **tar** command:

Tip: With Berkeley options the arguments to the flags should be given in exact order in which the flags are given below. For example:

```
tar cvfbL test.tar 20 infile
```

where `test.tar` is archive tar file, `20` is number of blocks, and `infile` is *Inputlist* for the archive.

1. To archive all directories and complete filenames listed in input list file `infile` into `ar.tar`, enter :

```
tar cvfL ar.tar infile
```

Where `infile` contains the pathnames of files that are to be archived.

2. To archive files within directories listed in the input list file `infile` into `ar.tar`, enter:

```
tar cvRfL ar.tar infile
```

3. To extract directories and complete files specified in the input list file `infile` from an archive named `ar.tar`, enter:

```
tar xvfL ar.tar infile
```

4. To extract files from within directories and complete files specified in the input list file `infile` from an archive named `ar.tar`, enter:

```
tar xvRfL ar.tar infile
```

Files

Item	Description
<code>/dev/rmt0</code>	Specifies the default tape device.
<code>/bin/tar</code>	Specifies the symbolic link to the tar command.
<code>/usr/bin/tar</code>	Contains the tar command.
<code>/tmp/tar*</code>	Specifies a temporary file.

Tip: In AIX 3.2, the entire `/bin` directory is a symbolic link to `/usr/bin`.

tbl Command

Purpose

Formats tables for the **nroff** and **troff** commands.

Syntax

```
tbl [ -TX ] [ - ] [ File... | - ]
```

Description

The **tbl** command is a preprocessor that formats tables for the **nroff** and **troff** commands. It reads one or more files. If no *File* parameter or `-` (minus sign) is specified as the last parameter, the command reads standard input by default. It copies the input unchanged to standard output, except for text between lines containing **.TS** and **.TE**. The **tbl** command reformats such text, which describes tables, without altering the **.TS** and **.TE** lines.

Depending on the target output device, the output formatted by the **nroff** command may need to be post-processed by the **col** command to produce correct output.

Note: To minimize the volume of data passed through pipelines, enter the **tbl** command first when using it with the **eqn** or **neqn** command.

Input Format

The **tbl** command processes text that is displayed within the following format:

```
[ {.DS .DF} ]
.TS
Options ;
Format .
Data
.TE
[.DE]
```

To include short tables in an **mm** macro document, enclose them within the **.DS** (or **.DF**) and **.DE** macro pair.

Options

Following are the available global options for the input format:

Option	Purpose
center or CENTER	Centers the line.
expand or EXPAND	Expands to line length.
box or BOX	Encloses in a box.
allbox or ALLBOX	Boxes all entries.
doublebox or DOUBLEBOX	Encloses in two boxes.
tab(Character) or TAB(Character)	Changes the tab character to the <i>Character</i> value.
linesize(Number) or LINESIZE(Number)	Makes all lines the thickness of the point size specified by the <i>Number</i> value.
delim(XY) or DELIM(XY)	Recognizes the <i>X</i> and <i>Y</i> variables as eqn command delimiters.
;	Denotes end of options.

Format

The *Format* variable in the Input Format describes the format of text. Each format line (the last of which must end with a period) describes all remaining lines of the table. A single-key letter describes each column of each line of the table. Follow this key letter with specifiers that determine the font and point size of the corresponding item, indicate where vertical bars are to be displayed between columns, and determine such things as column width and intercolumn spacing. The following are the available key letters:

Item	Description
l or L	Left-adjusts column.
r or R	Right-adjusts column.
c or C	Centers column.

Item	Description
n or N	Numerically aligns column. Note: Numerically aligned data, n or N format specification, are based upon the locale that is specific for <i>RADIXCHAR</i> , which is assumed to be a single character. The alignment can also be determined using the \& (backslash, ampersand) character sequence independent of the presence of any <i>RADIXCHAR</i> characters. If more than one <i>RADIXCHAR</i> character is displayed in a numerically aligned field, the last one is used for alignment. If no <i>RADIXCHAR</i> characters are displayed in a particular column, the alignment is based on the last ASCII arabic numeral. If there is no ASCII numeral and no <i>RADIXCHAR</i> character in a column, the data is centered.
a or A	Left-adjusts subcolumn.
s or S	Spans item horizontally.
t or T	Pushes vertical spans to top.
v or V	Adjusts vertical line spacing.
^	Spans item vertically.
u or U	Moves item half-line up.
z or Z	Indicates zero-width item.
-	Indicates horizontal line.
=	Indicates double horizontal line.
 	Indicates vertical line.
 	Indicates double vertical line.
b or B	Indicates boldface item.
i or I	Indicates italic item.
f <i>Character</i> or F <i>Character</i>	Changes to the font specified by the <i>Character</i> variable.
p <i>Number</i> or P <i>Number</i>	Changes to the size specified by the <i>Number</i> variable.
w (<i>Number</i>) or W (<i>Number</i>)	Sets minimum column width equal to the <i>Number</i> variable value.
<i>NumberNumber</i>	Spaces between columns.
e or E	Makes equal-width columns.
.	Ends format.

Data

Handling data within the input format, especially for tables, uses the following line commands:

Item	Description
T {... T }	Indicates text block, as follows: <i>Data</i> <TAB> T { <i>Text Block</i> T <TAB> <i>Data</i>

Item	Description
<code>\</code>	Writes short horizontal line.
<code>\RX</code>	Repeats the <i>X</i> parameter value across a column.
<code>\^</code>	Indicates that the item listed previously spans downward into this row.
<code>.T&</code>	Starts new format.
<code>.TS H</code> , <code>.TH</code> , and <code>.TE</code>	Allows multi-page tables with column headings repeated on each page. (This is a feature of the mm macros.)

Parameters

Item Description

File Specifies the files that the **tbl** command will be processing.

Flags

Item	Description
<code>-TX</code>	Uses only full vertical line motions, making the output suitable for line printers and other devices that do not have partial vertical line motions.
<code>--</code>	(double dash) Indicates the end of flags.
<code>-</code>	Forces input to be read from standard input.

Examples

The following example shows coded input, and associated table output of the **tbl** command. The @ (at sign) is used in input to represent an input tab character.

Input

```
.TS
center box ;
cB s s
cI | cI s
^ | c c
l | n n .
Household Population

Town@Households
@Number@Size
=
Bedminster@789@3.26
Bernards Twp.@3087@3.74
Bernardsville@2018@3.30
Bound Brook@3425@3.04
Bridgewater@7897@3.81
Far Hills@240@3.19
.TE
```

tc Command

Purpose

Interprets text into the **troff** command output for the Tektronix 4015 system.

Syntax

```
tc [ -t ] [ -e ] [ -a Number ] [ -o List | -s Number ] [ -- ] [ File | - ]
```

Description

The **tc** command interprets input as output from the **troff** command. The **tc** command reads one or more English-language files. If no file is specified or the - (minus sign) flag is specified as the last parameter, standard input is read by default. The standard output of the **tc** command is intended for a Tektronix 4015 (a 4014 terminal with ASCII and APL character sets). The various typesetter sizes are mapped into the 4014's four sizes. The entire **troff** command character set is drawn using the 4014 character generator, with overstruck combinations where necessary.

At the end of each page, the **tc** command waits for a new-line character from the keyboard before continuing to the next page. While it waits, the following commands are recognized:

Item	Description
!Command	Sends the value of the <i>Command</i> variable to the shell.
-e	Does not erase before each page.
-Number	Skips backward the specified number of pages.
-aNumber	Sets the aspect ratio to the value of the <i>Number</i> variable.
?	Prints a list of available options.

Note: The **tc** command does not distinguish among fonts.

Parameters

Item	Description
<i>File</i>	Specifies the English-language text files to be interpreted as output from the troff command.

Flags

Item	Description
-a Number	Sets the aspect ratio to the specified number. The default is 1.5.
-e	Does not erase before each page.
-o List	Prints only the pages enumerated in the <i>List</i> variable. The list consists of pages and page ranges (for example, 5-17) separated by commas. The range <i>Number-</i> goes from the <i>Number</i> variable value to end; the range <i>-Number</i> goes from the beginning to and including the page specified by the <i>Number</i> variable.
-s Number	Skips the first specified number of pages.
-t	Does not wait between pages when directing output into a file.
-	Reads from standard input.
--	(double dash) Indicates the end of flags.

Example

To use the **tc** command in a pipeline with the **troff** command, enter:

```
troff [Flag...] [File...] | tc
```

tcbck Command

Purpose

Audits the security state of the system.

Syntax

Check Mode

tcbck { **-n** | **-p** | **-t** | **-y** } [**-i**] [**-o**] { **ALL** | **tree** | { *Name ... Class ...* } }

Update Mode

tcbck -a -f *File* | *PathName Attribute = Value ...*

OR

tcbck -d -f *File* | { *PathName ...* | *Class ...* }

OR

tcbck -l */dev/filename /dev/filename*

Exit Status

This command returns the following exit values:

0

User definition files are appropriate.

>0

An error occurred or there is an error in one or more user definition files.

The following error codes are returned:

EINVAL (22)

Invalid command line arguments

ENOENT (2)

One or more user definition files do not exist

ENTRUST (114)

Errors in user definitions in the database files

Description

The **tcbck** command audits the security state of the system by checking the installation of the files defined in the **/etc/security/sysck.cfg** file (the sysck database). Each file definition in the **/etc/security/sysck.cfg** file can include one or more attributes that describe proper installation. When invoked with no flags and with no parameters, the **tcbck** command prints a synopsis of its syntax.

The tcbck database usually defines all the files and programs that are part of the trusted computing base, but the root user or a member of the security group can choose to define only those files considered to be security-relevant.

Note: This command writes its messages to **stderr**.

Flags

Item	Description
-a	Adds or updates file definitions in the sysck database.
-d	Deletes file definitions from the sysck database.

Item	Description
-f File	Specifies that file definitions be read from <i>File</i> .
-i	Excludes filesystems under directories listed in the treeck_nodir attribute when the tree option is specified.
-l	(Lowercase L) Adds entries to the sysck.cfg file for /dev/ files that the administrator would like registered with the Trusted Computing Base.
-n	Specifies the checking mode and indicates that errors are to be reported, but not fixed.
-o	Writes output to syslog.
-p	Specifies the checking mode and indicates that errors are to be fixed, but not reported.
-t	Specifies the checking mode and indicates that errors are to be reported with a prompt asking whether the error should be fixed.
-y	Specifies the checking mode and indicates that errors are to be fixed and reported.

Modes of Operation

The **tcbck** command has two modes of operation: check mode and update mode. A description of each mode follows.

Check Mode

In check mode, the **tcbck** command checks file definitions against the installed files. You can check all the file definitions in the sysck database (the **/etc/security/sysck.cfg** file) by specifying the **ALL** value, or all the files in the file system tree by specifying the **tree** value. If you prefer to check specific files, you can use the *Name* parameter to give the path names of individual files or the *Class* parameter to group several files into a logical group that is defined by a class name, such as *audit*. You must select one of the following: the **ALL** or **tree** values, or one or more files identified by the *Class* or *Name* parameter.

If the **tree** value is the selection criterion, all the files in the file system tree are checked to ensure that all the relevant files are defined in the sysck database. Files defined in the tcbck database are checked against their definitions. Files not in the tcbck database must *not*:

- Have the **trusted computing base** attribute set.
- Be **setuid** or **setgid** to an administrative ID.
- Be linked to a file in the tcbck database.
- Be a device special file.

If the **tcbck** command is running in check mode with both the **tree** value and the **-t** flag and an error occurs, the command provides an error message and prompts you for a decision on how or whether the error should be corrected. If you decide not to delete the file or turn off illegal permissions, you are prompted for a decision on updating the database. If you request an update, the system supplies missing information, such as the name of the file, the link, or the unregistered device name.

A flag (**-n**, **-p**, **-t**, **-y**) also must be included to specify check mode and identify the method of error handling. If there is a duplicate stanza in the **/etc/security/sysck.cfg** file, an error is reported, but not fixed.

Updating the Vital Product Database (VPD) involves defining the **type**, **checksum**, and **size** attributes of each file to the VPD manager. This information is used to verify a correct installation. If these attributes are not defined in **-f File**, they are computed when the program is installed or updated. The **checksum** attribute is computed with a method specifically defined for the VPD manager. Refer to [“Fixing Errors” on page 703](#) for more information on file attributes.

The only file definitions modified during an update are the new definitions that indicate a file is part of the trusted computing base (TCB). The *File* parameter is the stanza file that contains the file definitions in **tcbck** format, and is defined in the **/etc/security/sysck.cfg** file. When the update is complete, the files are checked against their file definitions in the stanza file and errors are fixed and reported.

Programs that require **setuid** or **setgid** privilege must be in the tcbck database, or these privileges will be cleared when the **tcbck** command runs in Check mode.

Update Mode

In update mode, the **tcbck** command adds (**-a**), deletes (**-d**), or modifies file definitions in the **/etc/security/sysck.cfg** file for the file specified by the *File* parameter, the *PathName* parameter, or the *Class* parameter. The *Class* parameter permits you to group several files into a logical group that is defined by a class name, such as audit. The **tcbck** command also deletes the specified stanzas from the **/etc/security/sysck.cfg** file.

In update mode, the **tcbck** command (**-l**) adds or modifies **/dev/** entry definitions in the **/etc/security/sysck.cfg** file for the specified **/dev** entry. This flag should be run by the administrator to add newly created devices that are trusted to the **sysck.cfg** file. If new devices are not added to the **sysck.cfg** file, the tree option produces warnings of unregistered devices.

The **-l** flag creates a stanza for each **/dev/** entry listed on the command line. The information for the stanza is taken from the current status of the **/dev** entry. The stanza includes:

Device name	/dev/ entry name
File type	Either FILE , DIRECTORY , FIFO , SYMLINK , BLK_DEV , CHAR_DEV , or MPX_DEV
Owner ID	Owner name
Group ID	Group name
Permissions	Read/write/execute permissions for owner, group and other. SUID , SGID , SVTX and TCB attribute bits
Target	If the file is a symbolic link, the target file will be listed.

File definitions to be added or modified with the **-a** flag can be specified on the command line or in a file as *Attribute=Value* statements. The following attributes can be used:

Item	Description
acl	The access control list for the file. If the value is blank , the acl attribute is removed. If no value is specified, the command computes a value, according to the format described in Access Control Lists.
class	The logical group of the file. A value must be specified, because it cannot be computed. If the value is blank , the class attribute is removed from the specified file stanza. The value is <i>ClassName</i> [<i>ClassName</i>].
checksum	The checksum of the file. If the value is blank , the checksum attribute is removed. If no value is specified, the command computes a value, according to the format given in the sum command. The value is the output of the sum -r command, including spaces.
group	The file group. If the value is blank , the group attribute is removed. If no value is specified, the command computes a value, which can be a group ID or a group name.
links	The hard links to this file. If the value is blank , the links attribute is removed. A value must be specified, because it cannot be computed. The value must be an absolute path name, expressed as <i>Path</i> [, <i>Path</i> ...].
mode	The File mode. If the value is blank , the mode attribute is removed. If no value is specified, the command computes a value, which can be an octal number or string (<i>rwX</i>), and have the tcb , SUID , SGID , and SVTX attributes.
owner	The file owner. If the value is blank , the owner attribute is removed. If no value is specified, the command computes a value, which can be a user ID or a user name.
program	The associated checking program for the file. If the value is blank , the program attribute is removed. A value must be specified, because it cannot be computed. The value must be an absolute path name. If flags are specified, the value should be expressed as <i>Path</i> , <i>Flag</i> .

Item	Description
symlinks	The symbolic links to the file. If the value is blank , the symlinks attribute is removed. A value must be specified, because it cannot be computed. The value must be an absolute path name, expressed as <i>Path</i> [, <i>Path</i> ..].
size	The size of the file in bytes. If the value is blank , the size attribute is removed. If no value is specified, the command computes a value. The value is a decimal number.
source	The source for the file. If the value is blank , the source attribute is removed. If no value is specified, an empty file of the appropriate type is created. The value must be an absolute path name.
type	The type of file. This value cannot be blank . If no value is specified, the command computes a value, which can be the FILE , DIRECTORY , FIFO , BLK_DEV , CHAR_DEV , or MPX_DEV keywords.

You can add, delete, or modify the attributes of the **tcck** command by creating or modifying a **sysck** stanza in the **/etc/security/sysck.cfg** file. The following attributes can be used:

Item	Description
checksum	An alternate checksum command to compute the checksum value of files. The system appends the name of each file to the command. If the value is blank , this alternate checksum attribute is removed. The value is the command string to be run on each file. The default string is /usr/bin/sum -r < .
setgids	An additional list of administrative groups to be checked for setgid programs that are not valid (groups with ID numbers greater than 200). If the value is blank , the setgids attribute is removed. The value is a comma separated list of group names.
setuids	An additional list of administrative users to be checked for setuid programs that are not valid (users with ID numbers greater than 200). If the value is blank , the setuids attribute is removed. The value is a comma separated list of user names.
treeck_nodir	A list of directories to be excluded from verification by the tcck command. If the value is blank, the treeck_nodir attribute is removed. The value is a comma separated list of directories. File systems that exist under directories contained in this attribute are <i>not</i> excluded. Use the -i flag to exclude these file systems. Use this option only when the tree option is specified.
treeck_novfs	A list of file systems to be excluded from verification by the tcck command during a check of an installed file system tree. If the value is blank , the treeck_novfs attribute is removed. The value is a comma separated list of file systems. Use this option only when the tree option is specified.

Refer to the **/etc/security/sysck.cfg** file for more information about these attributes and [“Examples”](#) on page 704 for information about a typical stanza.

If *Attributes* are included without values, the command tries to compute the value from the file to be changed. The **type** attribute is mandatory, but the others do not need to be specified.

Fixing Errors

To fix errors, the **tcck** command usually resets the attribute to the defined value. For the following attributes, the command modifies its actions as described:

Item	Description
checksum	Disables the file by clearing its access control list, but does not stop any further checks.
links	Creates any missing hard links. If a link exists to another file, the link is deleted.

Item	Description
program	Invokes the program, which must exist and have an absolute path name. A message is printed if an error occurs, but no additional action is taken.
size	Disables the file by clearing its access control list, but does not stop any further checks.
source	Copies the source file to the file identified by the <i>File</i> parameter. If the source is null, any existing file is deleted and a file of the correct type is created.
symlinks	Creates any missing symbolic links. If a link exists to another file, the link is deleted.
type	Disables the file by clearing its access control list, and stops any further checks.

If you used the **-t** flag with the **tcback** command, you are prompted for a decision on fixing errors. If you answer yes, errors are fixed. If you give any other response, errors are not fixed.

Security

Access Control: This command grants execute (x) access only to the root user and members of the security group. The command should be setuid to the root user and have the **trusted computing base** attribute.

Files Accessed:

Mode	File
r	/etc/passwd
r	/etc/group
r	/etc/security/user
rw	/etc/security/sysck.cfg
x	/usr/bin/aclget
x	/usr/bin/aclput
x	/usr/bin/sum

Auditing Events:

Event	Information
TCBCK_Check	file, error, status
TCBCK_Update	file, function

Examples

1. To add the **/bin/boo** file with **acl**, **checksum**, **class**, **group**, **owner**, and **program** attributes to the **tcback** database, type:

```
tcback -a /bin/boo acl checksum class=audit group owner\
program=/bin/boock
```

The resulting stanza will contain the attributes given previously, with computed values inserted for those attributes you do not define. The database will contain a stanza like the following:

```
/bin/boo:
acl =
checksum = 48235
class = audit
group = system
owner = root
program = /bin/boock
type = FILE
```


The attribute values are added to the installation definition but are not checked for correctness. The **program** attribute value comes from the command line, the **checksum** attribute value is computed with the **checksum** program, and all the others, except **acl**, are computed from the file i-node.

2. To indicate that the size of a file should be checked but not added to the database, because it can expand during installation, use the **VOLATILE** keyword, as in the following example for the **/etc/passwd** file:

```
/etc/passwd:
  type = FILE
  owner = root
  group = system
  size = 1234,VOLATILE
```

3. To delete the **/bin/boo** file definition from the **tcbck** database, type:

```
tcbck -d /bin/boo
```

4. To delete all definitions with a class of **audit** from the **tcbck** database, type:

```
tcbck -d audit
```

5. To check all the files in the **tcbck** database, and fix and report all errors, type:

```
tcbck -y ALL
```

6. To exclude the **/calvin** and the **/hobbes** file systems from verification during a security audit of an installed file system tree, type:

```
tcbck -a sysck treeck_novfs=/calvin,/hobbes
```

7. To exclude a directory from verification during a security audit, type:

```
tcbck -a sysck treeck_nodir=/home/john
```

8. To add **jfh** and **jsl** as administrative users and **developers** as an administrative group to be verified during a security audit of an installed file, type:

```
tcbck -a sysck setuids=jfh,jsl setgids=developers
```

9. To create/modify **sysck.cfg** stanza entries for the newly created **/dev** entries **foo** and **bar**, type:

```
tcbck -l /dev/foo /dev/bar
```

Note: By adding these entries you are registering them as part of the Trusted computing base.



Attention: Although the special characters "\$" and "?" are allowed in this routine, using them in filenames may result in potential problems such as ambiguous files.

Files

Item	Description
/usr/bin/tcbck	Specifies the path to the tcbck command.
/etc/security/sysck.cfg	Specifies the path to the system configuration database.

tcopy Command

Purpose

Copies a magnetic tape.

Syntax

tcopy *Source* [*Destination*]

Description

The **tcopy** command copies magnetic tapes. Source and target file names are specified by the *Source* and *Destination* parameters. The **tcopy** command assumes that there are two tape marks at the end of the tape, and it ends when it finds the double file marks. With only a source tape specified, the **tcopy** command prints information about the size of records and tape files

Examples

To copy from one streaming tape to a 9-track tape, enter:

```
tcopy /dev/rmt0 /dev/rmt8
```

Files

Item	Description
/usr/bin/tcopy	Contains the tcopy command.

tcpdump Command

Purpose

Dumps traffic on a network

Syntax

tcpdump [**-a**] [**-A**] [**-B** *buffer_size*] [**-d**] [**-D**] [**-e**] [**-f**] [**-l**] [**-K**] [**-L**] [**-M** *secret*] [**-r** *file*] [**-n**] [**-N**] [**-O**] [**-p**] [**-q**] [**-Q**] [**-V**] [**-R**] [**-S**] [**-t**] [**-T**] [**-u**] [**-U**] [**-v**] [**-x**] [**-X**] [**-c** *count*] [**-C** *file_size*] [**-F** *file*] [**-G** *rotate_seconds*] [**-i** *interface*] [**-s** *snaphlen*] [**-w** *file*] [**-E** *addr*] [**-y** *datalinktype*] [**-z** *command*] [**-Z** *user*] [*expression*]

Description

The **tcpdump** command prints the headers of packets on a network interface that match the boolean expression. You can run the command with the **-w** flag to save the packet data in a file for further analysis. You can also run the command with the **-r** flag to read data from a saved packet file instead reading the packets from a network interface. In all cases, only packets that match expression is processed by the **tcpdump** command.

If it is not run with the **-c** flag, **tcpdump** continues capturing packets until it is interrupted by a SIGINT signal (typically control-C) or a SIGTERM signal (typically the **kill** (1) command). If **tcpdump** is run with the **-c** flag, it captures the packets until it is interrupted by a SIGINT or SIGTERM signal or the specified number of packets have been processed.

The **tcpdump** command returns the following counts after capturing all the packets:

packets "received by filter"

Counts all packets regardless of whether they were matched by the filter expression.

packets "dropped by kernel"

The number of packets that were dropped, due to a lack of buffer space.

Allowable Primitives

dst host host

True if the IPv4/v6 destination field of the packet is host, which may be either an address or a name.

src host host

True if the IPv4/v6 source field of the packet is host.

host host

True if either the IPv4/v6 source or destination of the packet is host. Any of the above host expressions can be prepended with the keywords, ip, arp, rarp, or ip6 as in: ip host host which is equivalent to:

```
ether proto \ip and host host
```

If host is a name with multiple IP addresses, each address is checked for a match.

ether dst ehost

True if the ethernet destination address is ehost. Ehost may be either a name from /etc/ethers or a number (see ethers(3N) for numeric format).

ether src ehost

True if the ethernet source address is ehost.

ether host ehost

True if either the ethernet source or destination address is ehost.

gateway host

True if the packet used host as a gateway. For example, the ethernet source or destination address was host but neither the IP source nor the IP destination was host. Host must be a name and must be found both by the machine's host-name-to-IP-address resolution mechanisms (host name file, DNS, NIS, etc.) and by the machine's host-name-to-Ethernet-address resolution mechanism (/etc/ethers, and so on). An equivalent expression is ether host ehost and not host host which can be used with either names or numbers for host /ehost. This syntax does not work in IPv6-enabled configuration at this moment.

dst net net

True if the IPv4/v6 destination address of the packet has a network number of net.

src net net

True if the IPv4/v6 source address of the packet has a network number of net.

net net

True if either the IPv4/v6 source or destination address of the packet has a network number of net.

net net mask netmask

True if the IP address matches net with the specific netmask. This might be qualified with src or dst. This syntax is not valid for IPv6 net.

net net/len

True if the IPv4/v6 address matches net with a netmask len bits wide. May be qualified with src or dst.

dst port port

True if the packet is ip/tcp, ip/udp, ip6/tcp or ip6/udp and has a destination port value of port. The port can be a number or a name used in /etc/services (see tcp(4P) and udp(4P)). If a name is used, both the port number and protocol are checked. If a number or ambiguous name is used, only the port number is checked (For example, dst port 513 prints both tcp/login traffic and udp/who traffic, and port domain prints both tcp/domain and udp/domain traffic).

src port port

True if the packet has a source port value of port.

port port

True if either the source or destination port of the packet is port. Any of the above port expressions can be prepended with the keywords, tcp or udp, as in: tcp src port port which matches only tcp packets whose source port is port.

less length

True if the packet has a length less than or equal to length. This is equivalent to len <= length.

greater length

True if the packet has a length greater than or equal to length. This is equivalent to: `len >= length`.

ip proto protocol

True if the packet is an IP packet of protocol type protocol. Protocol can be a number or one of the names `icmp`, `icmp6`, `igmp`, `igrp`, `pim`, `ah`, `esp`, `rrrp`, `udp`, or `tcp`. Note that the identifiers `tcp`, `udp`, and `icmp` are also keywords and must be escaped via backslash (`\`), which is `\\` in the C-shell. Note that this primitive does not chase the protocol header chain.

ip6 proto protocol

True if the packet is an IPv6 packet of protocol type protocol. Note that this primitive does not chase the protocol header chain.

ip6 protochain protocol

True if the packet is IPv6 packet, and contains protocol header with type protocol in its protocol header chain. For example, `ip6 protochain 6` matches any IPv6 packet with TCP protocol header in the protocol header chain. The packet may contain, for example, authentication header, routing header, or hop-by-hop option header, between IPv6 header and TCP header. The Berkeley Packet Filter (BPF) code emitted by this primitive is complex and cannot be optimized by BPF optimizer code in `tcpdump`, so this can be somewhat slow.

ip protochain protocol

Equivalent to `ip6 protochain protocol`. But, this is used for `Ipv4`.

ether broadcast

True if the packet is an ethernet broadcast packet. The `ether` keyword is optional.

ip broadcast

True if the packet is an IPv4 broadcast packet. It checks for both the all-zeroes and all-ones broadcast conventions, and looks up the subnet mask on the interface on which the capture is being done.

If the subnet mask of the interface on which the capture is being done is not available, for example, because the interface on which capture is being done has no netmask this check does not work correctly.

ether multicast

True if the packet is an ethernet multicast packet. The `ether` keyword is optional. This is shorthand for `ether[0] & 1 != 0`.

ip multicast

True if the packet is an IP multicast packet.

ip6 multicast

True if the packet is an IPv6 multicast packet.

ether proto protocol

True if the packet is of ether type protocol. Protocol can be a number or one of the names `ip`, `ip6`, `arp`, `rarp`, `atalk`, `aarp`, `decnet`, `sca`, `lat`, `mopdl`, `moprc`, `iso`, `stp`, `ipx`, or `netbeui`. Note that these identifiers are also keywords and must be escaped via backslash (`\`).

[In the case of FDDI (e.g., ``fddi protocol arp'`), Token Ring (e.g., ``tr protocol arp'`), and IEEE 802.11 wireless LANs (e.g., ``wlan protocol arp'`), for most of those protocols, the protocol identification comes from the 802.2 Logical Link Control (LLC) header, which is usually layered on top of the FDDI, Token Ring, or 802.11 header. When filtering for most protocol identifiers on FDDI, Token Ring, or 802.11, `tcpdump` checks only the protocol ID field of an LLC header in so-called SNAP format with an Organizational Unit Identifier (OUI) of `0x000000`, for encapsulated Ethernet; it doesn't check whether the packet is in SNAP format with an OUI of `0x000000`. The exceptions are:

iso

`tcpdump` checks the DSAP (Destination Service Access Point) and SSAP (Source Service Access Point) fields of the LLC header.

stp and netbeui

`tcpdump` checks the DSAP of the LLC header.

atalk

`tcpdump` checks for a SNAP-format packet with an OUI of `0x080007` and the AppleTalk etype.

In the case of Ethernet, tcpdump checks the Ethernet type field for most of those protocols. The exceptions are:

iso, sap, and netbeui

tcpdump checks for an 802.3 frame and then checks the LLC header as it does for FDDI, Token Ring, and 802.11.

atalk

tcpdump checks both for the AppleTalk etype in an Ethernet frame and for a SNAP-format packet as it does for FDDI, Token Ring, and 802.11.

aarp

tcpdump checks for the AppleTalk ARP etype in either an Ethernet frame or an 802.2 SNAP frame with an OUI of 0x000000;

ipx

tcpdump checks for the IPX etype in an Ethernet frame, the IPX DSAP in the LLC header, the 802.3-with-no-LLC-header encapsulation of IPX, and the IPX etype in a SNAP frame.

decnet src host

True if the DECNET source address is host, which may be an address of the form 10.123, or a DECNET host name. [DECNET host name support is only available on Ultrix systems that are configured to run DECNET.]

decnet dst host

True if the DECNET destination address is host.

decnet host host

True if either the DECNET source or destination address is host.

ifname interface

True if the packet was logged as coming from the specified interface.

on interface

Synonymous with the ifname modifier.

rnr num

True if the packet was logged as matching the specified PF rule number (applies only to packets logged by OpenBSD's pf(4)).

rulenum num

Synonymous with the rnr modifier.

reason code

True if the packet was logged with the specified PF reason code. The known codes are: match, bad-offset, fragment, short, normalize, and memory (applies only to packets logged by OpenBSD's pf(4)).

action act

True if PF took the specified action when the packet was logged. Known actions are: pass and block (applies only to packets logged by OpenBSD's pf(4))

netbeui

ip, ip6, arp, rarp, atalk, aarp, decnet, iso, stp, ipx.

Abbreviations for:

```
ether proto p
```

where *p* is one of the above protocols.

lat, moprc, mopdl

Abbreviations for:

```
ether proto p
```

where *p* is one of the above protocols. Note that tcpdump does not currently know how to parse these protocols.

vlan [vlan_id]

True if the packet is an IEEE 802.1Q VLAN packet. If *vlan_id* is specified, only the packets that have the specified *vlan_id* are true. Note that the first *vlan* keyword encountered in expression changes the decoding offsets for the remainder of expression on the assumption that the packet is a VLAN packet.

tcp, udp, icmp

Abbreviations for:

```
ip proto p or ip6 proto p
```

where *p* is one of the above protocols.

iso proto protocol

True if the packet is an OSI packet of protocol type protocol. Protocol can be a number or one of the names clnp, esis, or isis.

clnp, esis, isis

Abbreviations for:

- iso proto p

where *p* is one of the above protocols.

l1, l2, iih, lsp, snp, csnp, psnp

Abbreviations for IS-IS PDU types.

vpi n

True if the packet is an ATM packet, for SunATM on Solaris, with a virtual path identifier of *n*.

vci n

True if the packet is an ATM packet, for SunATM on Solaris, with a virtual channel identifier of *n*.

lane

True if the packet is an ATM packet, for SunATM on Solaris, and is an ATM LANE packet. Note that the first *lane* keyword encountered in expression changes the tests done in the remainder of expression on the assumption that the packet is either a LANE emulated Ethernet packet or a LANE LE Control packet. If *lane* isn't specified, the tests are done under the assumption that the packet is an LLC-encapsulated packet.

llc

True if the packet is an ATM packet, for SunATM on Solaris, and is an LLC-encapsulated packet.

oamf4s

True if the packet is an ATM packet, for SunATM on Solaris, and is a segment OAM F4 flow cell (VPI=0 & VCI=3).

oamf4e

True if the packet is an ATM packet, for SunATM on Solaris, and is an end-to-end OAM F4 flow cell (VPI=0 & VCI=4).

oamf4

True if the packet is an ATM packet, for SunATM on Solaris, and is a segment or end-to-end OAM F4 flow cell (VPI=0 & (VCI=3 | VCI=4)).

oam

True if the packet is an ATM packet, for SunATM on Solaris, and is a segment or end-to-end OAM F4 flow cell (VPI=0 & (VCI=3 | VCI=4)).

metac

True if the packet is an ATM packet, for SunATM on Solaris, and is on a meta signaling circuit (VPI=0 & VCI=1).

bcc

True if the packet is an ATM packet, for SunATM on Solaris, and is on a broadcast signaling circuit (VPI=0 & VCI=2).

sc

True if the packet is an ATM packet, for SunATM on Solaris, and is on a signaling circuit (VPI=0 & VCI=5).

ilmic

True if the packet is an ATM packet, for SunATM on Solaris, and is on an ILMI circuit (VPI=0 & VCI=16).

connectmsg

True if the packet is an ATM packet, for SunATM on Solaris, and is on a signaling circuit and is a Q.2931 Setup, Call Proceeding, Connect, Connect Ack, Release, or Release Done message.

metaconnect

True if the packet is an ATM packet, for SunATM on Solaris, and is on a meta signaling circuit and is a Q.2931 Setup, Call Proceeding, Connect, Release, or Release Done message.

expr relop expr

True if the relation holds, where relop is one of >, <, >=, <=, =, !=, and expr is an arithmetic expression composed of integer constants (expressed in standard C syntax), the normal binary operators [+ , - , * , / , & , |], a length operator, and special packet data accessors. To access data inside the packet, use the following syntax:

```
proto [ expr : size ]
```

Proto is one of ether, fddi, tr, wlan, ppp, slip, link, ip, arp, rarp, tcp, udp, icmp or ip6, and indicates the protocol layer for the index operation. (ether, fddi, wlan, tr, ppp, slip and link all refer to the link layer.) Note that tcp, udp and other upper-layer protocol types only apply to IPv4, not IPv6 (this will be fixed in the future). The byte offset, relative to the indicated protocol layer, is given by expr. Size is optional and indicates the number of bytes in the field of interest; it can be either one, two, or four, and defaults to one. The length operator, indicated by the keyword len, gives the length of the packet.

For example, ether[0] & 1 != 0 catches all multicast traffic. The expression ip[0] & 0xf != 5 catches all IP packets with options. The expression ip[6:2] & 0x1fff = 0 catches only unfragmented datagrams and frag zero of fragmented datagrams. This check is implicitly applied to the tcp and udp index operations. For instance, tcp[0] always means the first byte of the TCP header, and never means the first byte of an intervening fragment.

Some offsets and field values may be expressed as names rather than as numeric values. The following protocol header field offsets are available: icmp type (ICMP type field), icmp code (ICMP code field), and tcp flags (TCP flags field).

The following ICMP type field values are available: icmp-echoreply, icmp-unreach, icmp-sourcequench, icmp-redirect, icmp-echo, icmp-routeradvert, icmp-routersolicit, icmp-timxceed, icmp-paramprob, icmp-tstamp, icmp-tstampreply, icmp-ireq, icmp-ireqreply, icmp-maskreq, icmp-maskreply.

The following TCP flags field values are available: tcp-fin, tcp-syn, tcp-rst, tcp-push, tcp-ack, tcp-urg.

Combining Primitives

A parenthesized group of primitives and operators (parentheses are special to the Shell and must be escaped).

```
Negation ('!' or 'not').
```

```
Concatenation ('&&' or 'and').
```

```
Alternation ('||' or 'or').
```

Negation has highest precedence. Alternation and concatenation have equal precedence and associate left to right. Note that explicit and tokens, not juxtaposition, are now required for concatenation.

If an identifier is given without a keyword, the most recent keyword is assumed. For example, not host vs and ace is short for not host vs and host ace which should not be confused with not (host vs or ace)

Expression arguments can be passed to `tcpdump` as either a single argument or as multiple arguments, whichever is more convenient. Generally, if the expression contains Shell metacharacters, it is easier to pass it as a single, quoted argument. Multiple arguments are concatenated with spaces before being parsed.

Flags

Item	Description
-a	Attempts to convert network and broadcast addresses to names.
-A	Prints each packet (minus its link level header) in ASCII. Handy for capturing web pages.
-B <i>buffer_size</i>	Indicates the buffer size in kilobytes. Smaller values are accepted. If the buffer size is smaller than the minimum value that is set by the BPF, the actual buffer size is ignored and the value that is set by the Berkeley Packet Filter (BPF) is used. If the -B option is not specified, the buffer size defaults to 32,768.
-c <i>Count</i>	Exits after receiving <i>Count</i> packets.
-C <i>file_size</i>	Before writing a raw packet to a <i>savefile</i> , check whether the file is currently larger than <i>file_size</i> and, if so, close the current <i>savefile</i> and open a new one. Save files after the first <i>savefile</i> has the name specified with the -w flag, with a number after it, starting at 2 and continuing upward. The units of <i>file_size</i> are millions of bytes (1,000,000 bytes, not 1,048,576 bytes).
-d	Dumps the compiled packet-matching code to standard output, then stops.
-D	Prints the list of the network interfaces available on the system and on which tcpdump can capture packets. For each network interface, a number and an interface name (possibly followed by a text description of the interface) is printed. The interface name or the number can be supplied to the -i flag to specify an interface on which to capture.
-dd	Dumps packet-matching code as a C program fragment.
-ddd	Dumps packet-matching code as decimal numbers (preceded with a count).
-e	Prints the link-level header on each dump line.

Item	Description
-E <i>addr</i>	Use <code>spi@ipaddr algo:secret</code> for decrypting IPsec ESP packets that are addressed to <i>addr</i> and contain Security Parameter Index value <code>spi</code>. This combination may be repeated with comma or newline separation. Note: Setting the secret for IPv4 ESP packets is now supported. Algorithms may be <code>des-cbc</code>, <code>3des-cbc</code>, <code>blowfish-cbc</code>, <code>rc3-cbc</code>, <code>cast128-cbc</code>, or <code>none</code>. The default is <code>des-cbc</code>. The ability to decrypt packets is only present if libcrypto is installed and is in <code>LIBPATH</code>. <code>secret</code> is the ASCII text for ESP secret key. If preceded by <code>0x</code>, then a hex value is read. The option assumes RFC2406 ESP, not RFC1827 ESP. The option is for debugging purposes only and the use of this option with a true secret key is discouraged. By presenting the IPsec secret key onto command line you make it visible to others, via ps(1) and other occasions. In addition to the above syntax, the tcpdump command might use the syntax file name to read the specified file. The file is opened upon receiving the first ESP packet, so any special permissions that tcpdump may have been given, should already have been given up.
-f	Prints foreign IPv4 addresses numerically rather than symbolically. The test for foreign IPv4 addresses is done by using the IPv4 address and netmask of the interface on which capture is being performed. This option does not work correctly if that address or netmask is not available.
-F <i>file</i>	Use <i>file</i> as input for the filter expression. An additional expression given on the command line is ignored.
-G <i>rotate_seconds</i>	Rotates the dump file that is specified with the -w option every <i>rotate_seconds</i> seconds. If used in conjunction with the -C option, file names take the form of <i>file <count></i> , if the value specified in the <i>size</i> variable is reached first. Otherwise, the tcpdump command rotates the file when the value specified in the <i>rotate_seconds</i> variable is elapsed.

Item	Description
-i <i>interface</i>	Listens on <i>interface</i>. If unspecified, tcpdump searches the system <i>interface</i> list for the lowest numbered, configured up <i>interface</i> (excluding loopback). Ties are broken by choosing the earliest match. An <i>interface</i> number as printed by -D flag can be used as the <i>interface</i> argument.
-K	Skips verification of TCP checksum on interfaces that perform TCP checksum calculation in hardware. If this flag is not used, all outgoing TCP checksums are flagged as bad.
-l	Makes <i>stdout</i> line buffered. Useful if you want to see the data while capturing it. For example: <pre>tcpdump -l tee dat or tcpdump -l > dat & tail -f dat</pre>
-L	Lists the known data link types for the interface and exits.
-m <i>module</i>	Loads SMI MIB module definitions from the <i>module</i> file. This option can be used several times to load several MIB modules into tcpdump .
-M	Uses secret as a shared secret for validating the digests that are found in TCP segments by using the TCP-MD5 option (Request for Comment (RFC) 2385).
-n	Blocks converting the host addresses, and the port numbers to names.
-N	Omits printing domain name qualification of host names. For example, tcpdump prints nic instead of nic.ddn.mil.
-O	Keeps tcpdump from running the packet-matching code optimizer. This is useful only if you suspect a bug in the optimizer.
-p	Stops putting the interface into promiscuous mode. Note that the interface might be in promiscuous mode for some other reason; hence, -p cannot be used as an abbreviation for ether host {local-hw-addr} or ether broadcast.
-q	Quick output. Prints less protocol information so output lines are shorter.
-Q	Enables filtered system tracing for the recorded packets. You must run the AIX trace daemon to record the selected system events that are related to the network communication subsystem.
-r <i>file</i>	Read packets from <i>file</i> (which was created with the -w option). Standard input is used if <i>file</i> is "-".

Item	Description
-R	Assumes ESP/AH packets are based on old specification. (RFC1825 to RFC1829). If specified, tcpdump does not print replay prevention field. Since there is no protocol version field in ESP/AH specification, tcpdump cannot deduce the version of ESP/AH protocol.
-S	Prints absolute rather than relative TCP sequence numbers.
-s <i>snaplen</i>	Snarf <i>snaplen</i> bytes of data from each packet rather than the default of 68. 68 bytes is adequate for IP, ICMP, TCP and UDP but may truncate protocol information from name server and NFS packets (see below). Packets truncated because of a limited snapshot are indicated in the output with [<i>proto</i>], where <i>proto</i> is the name of the protocol level at which the truncation has occurred. Note that taking larger snapshots increases the amount of time it takes to process packets and effectively decreases the amount of packet buffering. This can cause packets to be lost. You should limit <i>snaplen</i> to the smallest number that captures the protocol information you are interested in. Setting <i>snaplen</i> to 0 means use the required length to catch whole packets.
-T	Forces packets selected by <i>expression</i> to be interpreted the specified type. Currently known types are cnfp (Cisco NetFlow protocol), rpc (Remote Procedure Call), rtp (Real-Time Applications protocol), rtcp (Real-Time Applications control protocol), snmp (Simple Network Management Protocol), tftp (Trivial File Transfer Protocol), vat (Visual Audio Tool), and wb (distributed White Board).
-t	Omits the printing of a timestamp on each dump line.
-tt	Prints an unformatted timestamp on each dump line.
-ttt	Prints a delta (in microseconds) between current and previous line on each dump line.
-tttt	Prints a timestamp in default format proceeded by date on each dump line.
-ttttt	Prints a delta (in microseconds) between the current and the first line on each dump line.
-u	Prints undecoded NFS handles.
-U	Make output saved via the -w option, for example, "packet- buffered." As each packet is saved, it is written to the output file, rather than being written only when the output buffer fills.

Item	Description
-v	Specifies slightly more verbose output. For example, the time to live, identification, total length and options in an IP packet are printed. Also enables additional packet integrity checks such as verifying the IP and ICMP header checksum.
-vv	Even more verbose output than -v . For example, additional fields are printed from NFS and reply packets are fully decoded.
-vvv	Even more verbose output than -vv . For example, telnet SB ... SE options are printed in full. With -X Telnet options are printed in hex as well.
-V	Sets the socket debug flag (the SO_DEBUG socket option) and the trace level on sockets. This flag must be used along with the -Q flag.
-w file	Writes the raw packets to <i>file</i> rather than parsing and printing them out. They can later be printed with the -r flag. Standard output is used if <i>File</i> is "-".
-x	Prints each packet (minus its link level header) in hexadecimal. The smaller of the entire packet or snaplen bytes is printed. Note that this is the entire link-layer packet, so for link layers that pad (e.g. Ethernet), the padding bytes is also printed when the higher layer packet is shorter than the required padding.
-xx	Prints each packet, including its link level header, in hexadecimal.
-X	Prints each packet (minus its link level header) in hexadecimal and ASCII. This is very handy for analyzing new protocols.
-y datalinktype	Sets the data link type to use while capturing packets to <i>datalinktype</i> .
-z command	When used in conjunction with the -C or -G option, causes the tcpdump command to run the specified command on the <i>savefile</i> . For example, specifying -z gzip or -z bzip2 compresses each <i>savefile</i> by using the gzip or bzip2 command. Note: The tcpdump command runs the -z command in parallel to the capture by using the lowest priority so that this does not disturb the capture process.
-Z user	Runs the tcpdump command with the system privileges of the specified user.

Parameters

expressions

Selects the packets that are to be dumped. If an expression is provided, only the packets for which the expressions is true are dumped; otherwise, all the packets on the net are dumped.

The expression consists of one or more primitives. Primitives usually consist of an id (name or number) preceded by one or more qualifiers. There are three different kinds of qualifier:

type qualifiers say what type of primitive the id name or number refers to. Possible types are host, net and port. For example, 'host foo', 'net 128.3', 'port 20'. If there is no type qualifier, host is assumed.

dir qualifiers specify a particular transfer direction to and/or from id. Possible directions are src, dst, src or dst and src and dst. If there is no dir qualifier, src or dst is assumed. For some link layers, such as SLIP and for some other device types, the inbound and outbound qualifiers can be used to specify a desired direction.

proto qualifiers restrict the match to a particular protocol. Possible protos are fddi, tr, wlan, ip, ip6, arp, rarp, decnet, tcp and udp. If there is no proto qualifier, all protocols consistent with the type are assumed.

fddi is an alias for ether. The parser treats it as meaning "the data link level used on the specified network interface." FDDI headers contain Ethernet-like source and destination addresses, and often contain Ethernet-like packet types, so you can filter on these FDDI fields just as with the analogous Ethernet fields. FDDI headers also contain other fields, but they cannot be named in a filter expression.

Like fddi, tr and wlan are aliases for ether. The previous paragraph's statements about FDDI headers also apply to Token Ring and 802.11 wireless LAN headers. For 802.11 headers, the destination address is the DA field and the source address is the SA field; the BSSID, RA, and TA fields aren't tested.

In addition to the above, there are some special 'primitive' keywords that don't follow the pattern: gateway, broadcast, less, greater and arithmetic expressions. All of these are described below.

More complex filter expressions are built by using the words and, or, and not to combine primitives.

Environment Variables

LIBPATH environmental variable must be set or libcrypto library should be in /usr/lib for the -E flag to work. For example:

```
ksh$ LIBPATH=/opt/freeware/lib tcpdump -E"algo:secret"
```

Exit Status

Item	Description
0	Success
non-zero	Error

Security

Reading packets from a network interface requires read access to /dev/bpf*, which is typically root-only. Reading packets from a file does not require any special privileges except file read permission.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To print all packets arriving at or departing from sundown, enter:

```
tcpdump host sundown
```

2. To print traffic between helios and either hot or ace, enter:

```
tcpdump host helios and \( hot or ace \)
```

3. To print all IP packets between ace and any host except helios, enter:

```
tcpdump ip host ace and not helios
```

4. To print all traffic between local hosts and hosts at Berkeley, enter:

```
tcpdump net ucb-ether
```

5. To print all ftp traffic through internet gateway snup, enter:

```
tcpdump 'gateway snup and (port ftp or ftp-data)'
```

Note: The expression is quoted to prevent the shell from mis-interpreting the parentheses.

6. To print traffic neither sourced from nor destined for local hosts (if you gateway to one other net, this should never make it onto your local net), enter:

```
tcpdump ip and not net localnet
```

7. To print the start and end packets (the SYN and FIN packets) of each TCP conversation that involves a non-local host, enter:

```
tcpdump 'tcp[tcpflags] & (tcp-syn|tcp-fin) != 0 and not src and d dst net localnet'
```

8. To print IP packets longer than 576 bytes sent through gateway snup, enter:

```
tcpdump 'gateway snup and ip[2:2] > 576'
```

9. To print IP broadcast or multicast packets that were not sent via ethernet broadcast or multicast, enter:

```
tcpdump 'ether[0] & 1 = 0 and ip[16] >= 224'
```

10. To print all ICMP packets that are not echo requests/replies (for instance, not ping packets), enter:

```
tcpdump 'icmp[icmptype] != icmp-echo and icmp[icmptype] != icmp-e choreply'
```

Standard Error

All errors and warnings are sent to `stderr`.

Limitations

A packet trace that crosses a Daylight Saving Time change gives skewed time stamps (the time change is ignored).

Filter expressions on fields other than those in Token Ring headers handles the source-routed Token Ring packets incorrectly.

Filter expressions on fields other than those in 802.11 headers handles the 802.11 data packets with both To DS and From DS set incorrectly.

ip6 proto should chase header chain, but at this moment it does not. ip6 protochain is supplied for this behavior.

Arithmetic expression against transport layer headers, like `tcp[0]`, does not work against IPv6 packets. It only looks at IPv4 packets.

Packet tracing does not work in WPAR environment because the underlying BPF driver is not WPAR aware.

Files

Item	Description
/usr/sbin/tcpdump	Location of the tcpdump command. Optional
/usr/lib/libpcap.a	
/dev/bpf*	
/opt/freeware/lib/libcrypto.a(libcrypto.so)	

tcptr Command

Purpose

Configures or displays TCP Traffic Regulation (TR) policy information to control the maximum incoming socket connections for ports.

Syntax

tcptr -add < *start port* > < *end port* > < *max connection* > [*divisor*]

tcptr -delete < *start port* > < *end port* >

tcptr -show

Description

The **tcptr** command assigns a maximum limit of incoming TCP connections to a given network port or a range of ports. You can run this command to add new pools of connection resources to be shared collectively by incoming socket requests remotely accessing the AIX TCP-layer.

The system automatically ensures that resources are shared across multiple remote IP addresses that are attempting to connect through TCP to a specific port. Root users can control system resources related to TCP Traffic Regulation (TR).

Notes:

- By default, the **tcptr** command is not enabled.
- The **tcptr** command does not limit the rate of connections from a particular IP address. The total pool of connections from any client for a specific port or port-range is controlled.
- When the limit is reached, the connection to the server is lost. Message is not logged and the connection is lost, because the server is regulating the traffic and the system is following the instructions from the server.
- The TCP TR policies that are added by using the **tcptr** command are not activated until the **tcptr_enable** network attribute is set to a value of 1 by using the **no** command. These policies automatically persist after a system restart, but they are not activated until the network flag is enabled by using the **-p** flag as specified in the following command:

```
no -p -o tcptr_enable=1
```

Flags

Item	Description
-add	Adds new TCP TR policies to the system. You should specify the maximum allowable connections for the current policy, the start port, and the end port with the -add flag. The start port and the end port can be the same port when a port range is not specified. Optionally, you can specify a divisor to allow a greater diversity of resource sharing on the pool of available TCP connections.
-delete	Deletes existing TCP TR policies that are defined for the system. This flag requires the user specify the maximum allowable connections for the current policy, the start port, and the end port (can be the same as start port if not specifying a port-range).
-show	Displays all existing TCP TR policies defined on the system. You might use the -show flag to see the active policies before you use the -delete flag.

Parameters

Item	Description
<i>max connection</i>	Specifies the maximum incoming TCP connections for the given TR policy.
<i>start port</i>	Specifies the beginning port for the current TR policy.
<i>end port</i>	Specifies the end port for the current TR policy. If the port is a range, the value specified must be larger than the start port. If the TR policy is for a single port, the value specified must be equal to the value specified for the start port.
<i>divisor</i>	Specifies a divisor to compare the number of available incoming TCP connections with the number of consumed incoming TCP connections for an IP, and corresponds to a division of the overall available connections by a power of two. The divisor is the power of two that is used in the division. This parameter is optional, and if it is not specified, the default value is one. In that case, half of the number of available connections are used.

Algorithm for tcptr traffic regulation

When a new connection request is received, the **tcptr** command uses the following algorithm to allow or deny the new socket connections:

```
If a new connection request is received and (N-X) = 0, the request is rejected.
If a new connection request is received and (N-X) > 0 and
the request is from a source that already has connections
with this port(range), then:
    if X+1 < [(N-X)/2^divisor] then
        Allow the new connection
    else
        Deny the new connection
```


N

Maximum allowed connections for a port (range).

X

Currently used connections for a particular IP address.

divisor

Optional, default value is 1 (one).

Examples

1. To add a TCP Traffic Regulation Policy that covers only TCP port 23, and to set a maximum incoming connection pool of 256 with an available connections divisor of 3, enter the following command:

```
# tcptr -add 23 23 256 3
```

2. To add a TCP Traffic Regulation Policy that covers a TCP port that ranges from 5000 to 6000, and to set a maximum incoming connection pool of 5000 with an available connections divisor of 2, enter the following command:

```
# tcptr -add 5000 6000 5000 2
```

3. To show TCP Traffic Regulation Policies set for the system, enter the following command:

```
# tcptr -show
```

4. To delete the TCP Traffic Regulation Policy that covers a TCP port that ranges from 5000 to 6000, enter the following command:

```
# tcptr -delete 5000 6000
```

5. To add a TCP Traffic Regulation Policy with the IP address 10.20.30.1 that makes $256/2^3=32$ connections to port 80, enter the following command:

```
tcptr -add 80 80 256 3
```

In this case, the next connection attempt from this IP address to port 80 is rejected and a TCP RST is received.

tcspd Daemon

Purpose

Manages trusted computing resources.

Syntax

tcspd [**-f**]

Description

TrouSerS is an open source Trusted Computing Group Software Stack (TSS) that is released under the Common Public License. TrouSerS aims to be compliant with 1.1b and 1.2 TSS specifications.

According to the TSS specification, the **tcspd** daemon is a user-space daemon that must be the only portal to the Trusted Platform Module (TPM) device driver. At boot time, the system must start the **tcspd** daemon, and then the **tcspd** daemon communicates with the TPM device driver. From that point onwards, all requests to the TPM are routed through the TSS. The **tcspd** daemon manages the TPM resources and handles both local and remote requests from the TCG Service Provider (TSP).

Flags

Item	Description

-f	Runs the tcsd daemon in the foreground.

Access Control

There are two types of access control for the **tcsd** daemon: access to the daemon socket and access to specific commands that are internal to the **tcsd** daemon.

Access to the **tcsd** daemon port is controlled by the system administrator by using firewall rules.

Access to individual commands that are internal to the **tcsd** daemon is configured by the **remote_ops** directive of the **tcsd** configuration file. Each function call in the TCG Core Services (TCS) API is reachable by a unique ordinal. Each labeled **remote_op** directive defines a set of ordinals (usually more than one) that are necessary to accomplish the operation. For example, the **random** operation enables the ordinals for opening and closing a context, calling the **TCS_StirRandom**, the **TCS_GetRandom**, and the **TCS_FreeMemory** functions. By default, connections from a local host allow any ordinals.

Data Files

TSS applications have access to the following types of persistent storage:

User persistent storage

User persistent storage has a lifetime similar to the lifetime of the application that uses it; therefore, it is destroyed when an application exits. User persistent storage is controlled by the TSP of the application. By default, user persistent storage files are stored as `/var/tss/lib/tpm/user.{pid}`.

System persistent storage

System persistent storage is controlled by the TCS and stays valid across application lifetimes, the **tcsd** daemon restarts, and system resets. The data registered in system persistent storage remains valid until an application requests its removal. By default, system persistent storage files are stored as `/var/tss/lib/tpm/system.data`. The system persistent storage file is initially created when ownership of the TPM is received.

Files

Item	Description
<code>/etc/security/tss/tcsd.conf</code>	Contains all the default options and configurations for the tcsd daemon.

Conforming To

The **tcsd** daemon conforms to the TSS specification Version 1.10 Golden.

tctl Command

Purpose

Gives subcommands to a streaming tape device.

Syntax

```
tctl [ -f Device ] [ eof | weof | fsf | bsf | fsr | bsr | rewind | offline | rewoffl | erase | retension | reset | status ] [ Count ]
```

tctl [**-b** *BlockSize*] [**-f** *Device*] [**-p** *BufferSize*] [**-v**] [**-n**] [**-B**] { **read** | **write** }

Description

The **tctl** command gives subcommands to a streaming tape device. If you do not specify the *Device* variable with the **-f** flag, the **TAPE** environment variable is used. If the environment variable does not exist, the **tctl** command uses the **/dev/rmt0.1** device. (When the **tctl** command gives the **status** subcommand, the default device is **/dev/rmt0**.) The *Device* variable must specify a raw (not block) tape device. The *Count* parameter specifies the number of end-of-file markers, number of file marks, or number of records. If the *Count* parameter is not specified, the default count is 1.

Subcommands

Item	Description
eof or weof	Writes the number of end-of-file markers specified by the <i>Count</i> parameter at the current position on the tape. On an 8 mm tape drive, an end-of-file marker can be written in three places: • Before blank tape • Before an extended file mark • At the beginning-of-tape mark On a 9-track tape drive, the end-of-tape marker can be written at any location on the tape. However, this subcommand does not support overwriting single blocks of data.
fsf	Moves the tape forward the number of file marks specified by the <i>Count</i> parameter and positions it on the end-of-tape (EOT) side of the file mark.
bsf	Moves the tape backward the number of file marks specified by the <i>Count</i> parameter and positions it on the beginning-of-tape (BOT) side of the file mark. If the bsf subcommand moves the tape past the beginning, the tape rewinds, and the tctl command returns EIO .
fsr	Moves the tape forward the number of records specified by the <i>Count</i> parameter.
bsr	Moves the tape backwards the number of records specified by the <i>Count</i> parameter.
rewind	Rewinds the tape. The <i>Count</i> parameter is ignored.
offline or rewoffl	Rewinds the tape and takes the tape drive offline. This will unload the tape when appropriate. The tape must be re-inserted before the device can be used again.
erase	Erases all contents on the tape and rewinds it.
read	Reads from the specified tape device (using the specified block size) until the internal buffer is full, and then writes the data to standard output, continuing to read and write this way until an end-of-file (EOF) mark is reached.
reset	Sends a bus device reset (BDR) to the tape device. The BDR will only be sent if the device cannot be opened and is not busy.

Item	Description
retension	Moves the tape to the beginning, then to the end, and then back to the beginning of the tape. If you have excessive read errors during a restore operation, you should run the retension subcommand. If the tape has been exposed to environmental extremes, you should run the retension subcommand before writing to tape. The 8 mm tape drive will not respond to this command.
status	Prints status information about the specified tape device.
write	Opens the tape device, reads from standard input, and writes the data to the tape device.

Tip: When you specify the **read** or **write** subcommand, the **tctl** command opens the tape device and sets up the tape block size as specified by the **-b** or **-n** flag. If neither flag is specified, the **tctl** command uses a default block size of 512 bytes.

Restrictions:

- The **-b**, **-n**, **-p**, and **-v** flags apply only when using the **read** and **write** subcommands.
- The **-B** flag applies only when using the **read** subcommand.

Flags

Item	Description
-b <i>BlockSize</i>	Specifies, in bytes, the size of buffer used to read and write to the tape device, and also specifies, in the absence of the -n flag, the tape block size. If the block size is 0, variable-length blocks are used and the size of the tape buffer is 32,768. If the -b flag is not specified, the default block size and the size of the tape buffer is 512 bytes.
-B	Writes the contents of the buffer each time the tape is read. Set this flag when reading variable-length records that are not of a regular and consistent size.
-f <i>Device</i>	Specifies the tape device.
-p <i>BufferSize</i>	Specifies the size of the buffer to be used on standard input and standard output. The default buffer size is 32,768 bytes. The <i>BufferSize</i> value must be a multiple of the tape block size.
-v	Verbose. Prints the sizes of each read and write to standard error.
-n	Specifies variable-length records when reading or writing to tape with the read or write subcommand.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Examples

1. To rewind the `rmt1` tape device, enter:

```
tctl -f /dev/rmt1 rewind
```

2. To move forward two file marks on the default tape device, enter:

```
tctl fsf 2
```

3. To write two end-of-file markers on the tape in /dev/rmt0.6, enter:

```
tctl -f /dev/rmt0.6 weof 2
```

4. To read a tape device formatted in 80-byte blocks and put the result in a file, enter:

```
tctl -b 80 read > file
```

5. To read variable-length records from a tape device formatted in 80-byte blocks and put the result in a file, enter:

```
tctl -b 80 -n read > file
```

6. To write variable-length records to a tape device using a buffer size of 1024 bytes, enter:

```
cat file | tctl -b 1024 -n -f/dev/rmt1 write
```

7. To write to a tape device in 512-byte blocks and use a 5120-byte buffer for standard input, enter:

```
cat file | tctl -v -f /dev/rmt1 -p 5120 -b 512 write
```

Note: The only valid block sizes for quarter-inch (QIC) tape drives are 0 and 512.

8. To write over one of several backups on an 8 mm tape, position the tape at the start of the backup file and issue these commands:

```
tctl bsf 1
```

```
tctl eof 1
```

The first command moves the tape to the beginning-of-tape side of the file mark. The second command rewrites the file mark, because writing is allowed before extended file marks. The erase head of the drive erases data before the write head reaches it, so the **write** subroutines can write over data already in the tape. However, all old data following is lost because its file markers are meaningless.

Note: The **write** subroutines cannot write over a short file mark unless blank tape follows the short file mark. To write over existing data, as in the case of this example, the tape must be written with extended file marks (as specified through the SMIT interface).

Files

Item	Description
/dev/rmt <i>n</i>	Specifies the raw streaming tape interface.
/usr/bin/tctl	Contains the tctl command.

tee Command

Purpose

Displays the output of a program and copies it into a file.

Syntax

```
tee [ -a ] [ -i ] [ File ... ]
```

Description

The **tee** command reads standard input, then writes the output of a program to standard output and simultaneously copies it into the specified file or files.

Flags

Item	Description

-a	Adds the output to the end of <i>File</i> instead of writing over it.
-i	Ignores interrupts.

Exit Status

This command returns the following exit values:

Item	Description

0	The standard input was successfully copied to all output files.
>0	An error occurred.

Note: If a write to any successfully opened *File* operand is not successful, writes to other successfully opened *File* operands and standard output will continue, but the exit value will be >0.

Examples

1. To view and save the output from a command at the same time:

```
lint program.c | tee program.lint
```

This displays the standard output of the command **lint program.c** at the workstation, and at the same time saves a copy of it in the file `program.lint`. If a file named `program.lint` already exists, it is deleted and replaced.

2. To view and save the output from a command to an existing file:

```
lint program.c | tee -a program.lint
```

This displays the standard output of the **lint program.c** command at the workstation and at the same time appends a copy of it to the end of the `program.lint` file. If the `program.lint` file does not exist, it is created.

Files

Item	Description
<code>/usr/bin/tee</code>	Contains the tee command.

telinit or init Command

Purpose

Initializes and controls processes.

Syntax

```
{ telinit | init } { 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | h | Q | q | S | s | M | m | N }
```

Description

The **init** command initializes and controls processes. Its primary role is to start processes based on records read from the **/etc/inittab** file. The **/etc/inittab** file usually requests that the **init** command run the **getty** command for each line on which a user can log in. The **init** command controls autonomous processes required by the system.

The process that constitutes the majority of the **init** command's process dispatching activities is **/usr/sbin/getty**. The **/usr/sbin/getty** process initiates individual terminal lines. Other processes typically dispatched by the **init** command are daemons and the shell.

The **telinit** command, which is linked to the **init** command, directs the actions of the **init** command. The **telinit** command takes a one-character argument and signals the **init** command by way of the **kill** subroutine to perform the appropriate action.

The **telinit** command sets the system at a specific run level. A run level is a software configuration that allows only a selected group of processes to exist. The system can be at one of the following run levels:

Item	Description
0-9	Tells the init command to place the system in one of the run levels 0-9 . When the init command requests a change to run levels 0-9 , it kills all processes at the current run levels and then restarts any processes associated with the new run levels.
0-1	Reserved for the future use of the operating system.
2	Contains all of the terminal processes and daemons that are run in the multiuser environment. In the multiuser environment, the /etc/inittab file is set up so that the init command creates a process for each terminal on the system. The console device driver is also set to run at all run levels so the system can be operated with only the console active.
3-9	Can be defined according to the user's preferences.
S,s,M,m	Tells the init command to enter the maintenance mode. When the system enters maintenance mode from another run level, only the system console is used as the terminal.

The following arguments also serve as directives to the **init** command:

Item	Description
a,b,c,h	Tells the init command to process only those records in the /etc/inittab file with a, b, c, or h in the run level field. These four arguments, a, b, c, and h, are not true run levels. They differ from run levels in that the init command cannot request the entire system to enter run levels a, b, c, or h. When the init command finds a record in the /etc/inittab file with a value of a, b, c, or h in the run level field, it starts the process. However, it does not kill any processes at the current run level; processes with a value of a, b, c, or h in the run level field are started in addition to the processes already running at the current system run level. Another difference between true run levels and a, b, c, or h is that processes started with a, b, c, or h are not stopped when the init command changes run levels. Three ways stop a, b, c, or h processes: • Type off in the <i>Action</i> field. • Delete the objects entirely. • Use the init command to enter maintenance state.
Q,q	Tells the init command to re-examine the /etc/inittab file.
N	Sends a signal that stops processes from being respawned.

During system startup, after the root file system has been mounted in the pre-initialization process, the following sequence of events occurs:

1. The **init** command is run as the last step of the startup process.

2. The **init** command attempts to read the **/etc/inittab** file.
3. If the **/etc/inittab** file exists, the **init** command attempts to locate an **initdefault** entry in the **/etc/inittab** file.
 - a. If the **initdefault** entry exists, the **init** command uses the specified run level as the initial system run level.
 - b. If the **initdefault** entry does not exist, the **init** command requests that the user enter a run level from the system console (**/dev/console**).
 - c. If the user enters an **S**, **s**, **M** or **m** run level, the **init** command enters maintenance run level. These are the only run levels that do not require a properly formatted **/etc/inittab** file.
4. If the **/etc/inittab** file does not exist, the **init** command places the system in the maintenance run level by default.
5. The **init** command rereads the **/etc/inittab** file every 60 seconds. If the **/etc/inittab** file has changed since the last time the **init** command read it, the new commands in the **/etc/inittab** file are executed during system startup.

When you request the **init** command to change the run level, the **init** command reads the **/etc/inittab** file to identify what processes should exist at the new run level. Then, the **init** command cancels all processes that should not be running at the new level and starts any processes that should be running at the new level.

The processes run by the **init** command for each of these run levels are defined in the **/etc/inittab** file. The run level is changed by having a root user run the **telinit** command, which is linked to the **init** command. This user-run **init** command sends appropriate signals to the original **init** command initiated by the system during startup. The default run level can be changed by modifying the run level for the **initdefault** entry in the **/etc/inittab** file.

In the maintenance run level, the **/dev/console** console terminal is opened for reading and writing. The password for root is prompted. When the root password is entered successfully, the **su** command is invoked. Two ways exist to exit from the maintenance run level:

- If the shell is terminated, the **init** command requests a new run level.

OR

- The **init** (or **telinit**) command can signal the **init** command and force it to change the run level of the system.

During a system startup attempt, apparent failure of the **init** command to prompt for a new run level (when **initdefault** is maintenance) may be due to the fact that the terminal console device (**/dev/console**) has been switched to a device other than the physical console. If this occurs and you wish to work at the physical console rather than the **/dev/console**, you can force the **init** command to switch to the physical console by pressing the DEL (delete) key at the physical console device.

When the **init** command prompts for a new run level, enter one of the digits **0** through **9** or any of the letters **S**, **s**, **M**, or **m**. If you enter **S**, **s**, **M**, or **m**, the **init** command operates in maintenance mode with the additional result that if control had previously been forced to switch to the physical console, the **/dev/console** file is switched to this device as well. The **init** command generates a message to this effect on the device to which the **/dev/console** file was previously connected.

If you enter a **0** through **9** run level, the **init** command enters the corresponding run level. The **init** command rejects any other input and re-prompts you for the correct input. If this is the first time the **init** command enters any run level other than maintenance, it searches the **/etc/inittab** file for entries with the **boot** or **bootwait** keywords. If the **init** command finds these keywords, it performs the corresponding task, provided the run level entered matches that of the entry. For example, if the **init** command finds the **boot** keyword, it boots the machine. Any special initialization of the system, such as checking and mounting file systems, takes place before any users are allowed on the system. The **init** command then scans the **/etc/inittab** file to find all entries that are processes for that level. It then resumes normal processing of the **/etc/inittab** file.

Run level **2** is defined by default to contain all of the terminal processes and daemons that are run in the multiuser environment. In the multiuser environment, the **/etc/inittab** file is set up so that the **init** command creates a process for each terminal on the system.

For terminal processes, the shell terminates either as a result of an end of file character (EOF) typed explicitly or as the result of disconnection. When the **init** command receives a signal telling it that a process has terminated, it records the fact and the reason it stopped in **/etc/utmp** file and **/var/adm/wtmp** file. The **/var/adm/wtmp** file keeps a history of the processes started.

To start each process in the **/etc/inittab** file, the **init** command waits for one of its descendant processes to stop, for a power fail signal **SIGPWR**, or until the **init** command is signaled by the **init** or **telinit** commands to change the system's run level. When one of the above three conditions occurs, the **init** command re-examines the **/etc/inittab** file. Even if new entries have been added to the **/etc/inittab** file, the **init** command still waits for one of the three conditions to occur. To provide for instantaneous response, re-examine the **/etc/inittab** file by running the **telinit -q** command.

If the **init** command finds that it is continuously running an entry in the **/etc/inittab** file (more than five times in 225 seconds), it assumes that an error in the entry command string exists. It then prints an error message to the console and logs an error in the system error log. After the message is sent, the entry does not run for 60 seconds. If the error continues to occur, the command will respawn the entry only five times every 240 seconds. The **init** command continues to assume an error occurred until the command does not respond five times in the interval, or until it receives a signal from a user. The **init** command logs an error for only the first occurrence of the error.

When the **init** command is requested to change run levels by the **telinit** command, the **init** command sends a **SIGTERM** signal to all processes that are undefined in the current run level. The **init** command waits 20 seconds before stopping these processes with the **SIGKILL** signal.

If the **init** command receives a **SIGPWR** signal and is not in maintenance mode, it scans the **/etc/inittab** file for special power fail entries. The **init** command invokes the tasks associated with these entries (if the run levels permit) before any further processing takes place. In this way, the **init** command can perform cleanup and recording functions whenever the system experiences a power failure. It is important to note that these power fail entries should not use devices that need to be initialized first.

Environments

Because the **init** command is the ultimate ancestor of every process on the system, every other process on the system inherits the **init** command's environment variables. As part of its initialization sequence, the **init** command reads the **/etc/environment** file and copies any assignments found in that file into the environment passed to all of its subprocesses. Because **init** subprocesses do not run from within a login session, they do not inherit a **umask** setting from **init**. These processes may set the **umask** to whatever value they require. A command that is executed by **init** from the **/etc/inittab** file uses **init**'s **ulimit** values and not the default values as given in **/etc/security/limits**. The result is that a command that is successfully executed from the command line may not execute correctly when invoked by **init**. Any command that has specific **ulimit** requirements should include specific actions to set the **ulimit** values as required.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To request the **init** command to reexamine the **/etc/inittab** file, enter:

```
telinit q
```

2. To request the **init** command to enter maintenance mode, enter:

```
telinit s
```

Files

Item	Description
/etc/inittab	Specifies the init command control file.
/etc/utmp	Specifies the record of logged-in users.
/var/adm/wtmp	Specifies the permanent login accounting file.
/sbin/rc.boot	Specifies the pre-initialization command file.
/etc/rc	Specifies the initialization command file.
/etc/environment	Specifies system environment variables.
/dev/console	Specifies the console device driver.

telnet, tn, or tn3270 Command

Purpose

Connects the local host with a remote host, using the Telnet interface.

Syntax

```
{ telnet | tn | tn3270 } [ -d ] [ -p ] [ -n TraceFile ] [ -e TerminalType ] [ -f | -F ] [ -k realm ] [ -l user ] [ Host [ Port ] ]
```

Description

The **telnet** command, which is also referred to as the **tn** or **tn3270** command, operates in two different modes: command mode and input mode.

System

The user is assigned a default login Sensitivity Label (SL) and Integrity Label (TL), which is SL and TL of the user's process after successful login.

If the user does not want to login using the default login SL, the user can choose to supply a different SL at the login time using the **-e** option. The SL that is supplied by the user must be dominated by the user's clearance and contained in the system accreditation range. The TL cannot be specified by the user at login time. The default login SL and TL are defined in the **/etc/security/user** file along with the username and clearance for each user. To use the **-e** option, the server side's kernel trusted network bit must be turned off.

Command Mode

When the **telnet** command is issued without arguments, it enters command mode, as indicated by the **telnet>**, **tn>**, or the **tn3270>** prompt. A user can also enter command mode from input mode by pressing Ctrl-] for the **telnet** command, Ctrl-T for the **tn** command, or Ctrl-C for the **tn3270** command. In command mode, subcommands can be entered to manage the remote system. Some of these subcommands return you to the remote session upon completion. For those subcommands that do not, pressing the Enter key returns you to the remote session.

Note: The default escape sequence for this command is Ctrl-] for the **telnet** command, Ctrl-T for the **tn** command, or Ctrl-C for the **tn3270** command. This default can be overridden by changing the **TNESC** environment variable.

To enter **telnet** command mode while connected to a remote host, type the Telnet escape key sequence. When in command mode, the standard operating system editing conventions, such as backspace, are available.

Input Mode

When the **telnet** command is issued with arguments, it performs an **open** subcommand with those arguments and then enters input mode. The type of input mode is either character-at-a-time or line-by-line, depending on what the remote system supports. In character-at-a-time mode, most text that is typed is immediately sent to the remote host for processing. In line-by-line mode, all text is echoed locally and completed lines are sent to the remote host.

In either input mode, if the **toggle localchars** subcommand has a value of True, the user's QUIT, INTR, and FLUSH characters are trapped locally and sent as Telnet Protocol sequences to the remote host. The **toggle autoflush** and **toggle autosynch** subcommands cause this action to flush subsequent output to the terminal until the remote host acknowledges the Telnet sequence and to flush previous terminal input (in the case of QUIT and INTR characters).

Arabic/Hebrew Support

The **telnet**, **tn**, and **tn3270** command supports the Arabic and Hebrew texts, allowing the user to type Arabic or Hebrew characters while in an emulation session. The **Ar_AA** locale displays the Arabic characters in their correct shapes. The following functions support the bidirectional Arabic and Hebrew texts:

Language Selection

This function allows you to toggle the language layer. Activate the Arabic/Hebrew language selection with the following key combinations:

Item	Description
Alt+N	From an AIX terminal
Esc+N	From an ASCII terminal
Alt+N or Esc+N	From a Latin AIX terminal

Activate the Latin language layer with the following key combinations:

Item	Description
Alt+L	From an Arabic or Hebrew AIX terminal
Esc+L	From an ASCII terminal
Alt+L or Esc+L	From an AIX terminal

Screen Reverse

This function reverses the screen image and invokes the default language of the new screen orientation. Thus, if the screen is reversed to right-to-left, the language is changed to Arabic/Hebrew. If the screen is reversed to left-to-right, the language is changed to Latin.

If symmetric character swapping is enabled, reversing the screen causes bidirectional characters to be replaced by their counterparts. For example, if numeric character swapping is enabled, reversing the screen causes Hindi numerals to be replaced by their Arabic counterparts and the Arabic numerals to be replaced by their Hindi counterparts.

Activate screen reverse with the following key combinations:

Item	Description
Alt+S	From an Arabic or Hebrew AIX terminal
Esc+S	From an ASCII terminal

Item	Description
Alt+S or Esc+S	From a Latin AIX terminal

Push/End Push

The Push function allows you to edit text whose direction is opposite the screen orientation. When you activate this function, the cursor orientation is reversed, the language layer is changed accordingly, and a Push segment is created.

The Push function has two secondary modes:

Item	Description
Boundary Mode	This mode is activated upon entering the Push mode. In this mode, the cursor remains in its position while you type additional characters. The text is pushed in the opposite direction of the screen orientation.
Edit Mode	This mode is activated when the cursor is moved from its boundary position into the Push segment area. In this mode, you can edit the text within the Push segment, while typing in the field's natural direction.

Activate this function with the following key combinations:

Item	Description
Alt+P	From an Arabic or Hebrew AIX terminal
Esc+P	From an ASCII terminal
Alt+P or Esc+P	From a Latin AIX terminal

The End Push function terminates the Push function. The cursor jumps to the end of the Push segment and its direction changes to the original direction. You can activate End Push by pressing any field exit keys such as cursor up, cursor down, or any attention identifier (AID) key such as the Enter key. You can also activate this function with the following key combinations:

Item	Description
Alt+E	From an Arabic or Hebrew AIX terminal
Esc+E	From an ASCII terminal
Alt+E or Esc+E	From a Latin AIX terminal

Field Reverse

This function toggles the field orientation to either the opposite of or the same as the screen orientation. This function does not invert the text in the field. The cursor orientation is set to the new field orientation and the language layer is selected accordingly.

For example, if the cursor is in the first logical position of a field or line when you activate the field reverse function, the cursor skips to the opposite side of that field or line. This position is now the first logical position. If the cursor is not in the first position of the field or line when you activate field reverse function, the cursor remains in its position and allows natural and correct editing of the existing text. Activate this function with the following key combinations:

Item	Description
Alt+R	From an Arabic or Hebrew AIX terminal
Esc+R	From an ASCII terminal
Alt+R or Esc+R	From a Latin AIX terminal

Autopush

This function assists you in typing mixed left-to-right and right-to-left text. When enabled, reversed segments are automatically initiated and terminated according to the typed characters or the selected language layer. Thus, this mode automatically invokes the Push mode and relieves you of invoking the Push function.

When you type a digit or Latin character in a right-to-left field, the Autopush function automatically initiates the Push function without changing the language. If you type additional digits or Latin character, the Push function continues; otherwise, the Push function automatically terminates. Thus, you can type Arabic/Hebrew text with embedded digits or Latin characters without invoking the Push/End Push functions.

When you type an Arabic/Hebrew character in a left-to-right field, the Autopush function automatically initiates the Push function without a language change. If you then type a digit or Latin character, the Autopush function automatically terminates. Thus, you can type Latin text with embedded Arabic/Hebrew text using the Language Selection function rather than the Push/End Push functions.

Activate this function with the following key combinations:

Item	Description
Alt+A	From an Arabic or Hebrew AIX terminal
Esc+A	From an ASCII terminal
Alt+A or Esc+A	From a Latin AIX terminal

Field Shape

This function shapes the Arabic characters in the current field or line. Activate this function with the following key combinations:

Item	Description
Alt+H	From an Arabic AIX terminal
Esc+H	From an ASCII terminal
Alt+H or Esc+H	From a Latin AIX terminal

Field Deshape

This function deshapes Arabic text in the current field or line. Activate this function with the following key combinations:

Item	Description
Alt+B	From an Arabic AIX terminal
Esc+B	From an ASCII terminal
Alt+B or Esc+B	From a Latin AIX terminal

Contextual Shape Determination

This function determines the shape of an Arabic character based on the surrounding text. Use the Contextual Shape Determination function only when typing or editing right-to-left text. This function is terminated when any of the specific shape selection keys is pressed. This is the default function. Activate this function with the following key combinations:

Item	Description
Alt+C	From an Arabic AIX terminal
Esc+C	From an ASCII terminal
Alt+C or Esc+C	From a Latin AIX terminal

Initial Shape Determination

This function shapes Arabic characters in their initial shapes. Activate this function with the following key combinations:

Item	Description
Alt+I	From an Arabic AIX terminal
Esc+I	From an ASCII terminal
Alt+I or Esc+I	From a Latin AIX terminal

Middle Shape Determination

This function shapes Arabic characters in their middle shapes. Activate this function with the following key combinations:

Item	Description
Alt+M	From an Arabic AIX terminal
Esc+M	From an ASCII terminal
Alt+M or Esc+M	From a Latin AIX terminal

Isolated Shape Determination

This function shapes Arabic characters in their isolated shapes. Activate this function with the following key combinations:

Item	Description
Alt+O	From an Arabic AIX terminal
Esc+O	From an ASCII terminal
Alt+O or Esc+O	From a Latin AIX terminal

Final Shape Determination

This function shapes Arabic characters in their final shapes. Activate this function with the following key combinations:

Item	Description
Alt+Y	From an Arabic AIX terminal
Esc+Y	From an ASCII terminal
Alt+Y or Esc+Y	From a AIX terminal

Miscellaneous Functions

To activate numeric swapping, type the following line at the command line:

```
export ARB_NUM_SWAP=1
```

To activate symmetric swapping, that is, to swap bidirectional characters such as braces, brackets, and so on, type the following line at the command line:

```
export ARB_SYM_SWAP=1
```

To specify the code page that the host uses, type the following line at the command line:

```
export RM_HOST_LANG=IBM-420
```

Terminal Type Negotiation

The **telnet** command negotiates the terminal type, using the Telnet protocol, and it sets the **TERM** environment variable according to what has been negotiated.

To override the terminal negotiation from the console, use the **EMULATE** environment variable or the **-e** flag; or invoke the **tn3270** command if you require 3270 emulation. To determine whether terminal-type negotiation is performed, the following list describes the order of the **telnet** command processing:

1. The **-e** command-line flag. (No negotiation.)
2. The **EMULATE** environment variable. (No negotiation.)
3. The **tn3270** command. (No negotiation.)
4. If steps 1, 2, and 3 are not present, terminal-type negotiation occurs automatically.

If the client and the server negotiate to use a 3270 data stream, the keyboard mapping is determined by the following precedence:

Item	Description
\$HOME/.3270keys	Specifies the user's 3270 keyboard mapping when the tn or telnet command is invoked. If you are using a color display, you can also change this file to customize the colors for 3270 displays.
/etc/map3270	Specifies the user's 3270 keyboard mapping when the tn3270 command is invoked. The /etc/map3270 file defines keyboard mapping and colors for the tn3270 command.
/etc/3270.keys	Specifies the base 3270 keyboard mapping for use with limited function terminals.

Secure Attention Key (SAK) Option

In addition to terminal negotiation, the **telnet** command allows negotiation for the Secure Attention Key (SAK) option. This option, when supported, provides the local user with a secure communication path to the remote host for tasks such as changing user IDs or passwords. If the remote host supports the **SAK** function, a trusted shell is opened on the remote host when the **telnet send sak** subcommand is issued. The **SAK** function can also be assigned to a single key available in **telnet** input mode, using the **set sak** subcommand.

End-of-Line Convention

The Telnet protocol defines the carriage-return line-feed (CR-LF) sequence to mean "end-of-line." For terminal input, this corresponds to a command-completion or end-of-line key being pressed on a user terminal. On an ASCII terminal, this is the CR key, but it may also be labeled "Return" or "Enter."

When a Telnet server receives the Telnet end-of-line sequence, CR-LF, as input from a remote terminal, the effect is the same as if the user had pressed the end-of-line key on a local terminal.

On ASCII servers, receiving the Telnet sequence CR-LF causes the same effect as a local user pressing the CR key on a local terminal. CR-LF and CR-NUL have the same effect on an ASCII server when received as input over a Telnet connection.

Note: A Telnet user must be able to send CR-LF, CR-NULL, or LF. An ASCII user must be able to send CR-LF or CR-NULL.

A Telnet user on an ASCII host should have a user-controllable mode to send either CR-LF or CR-NULL when the user presses the end-of-line key. The CR-LF should be the default. The Telnet end-of-line sequence, CR-LF, must be used to send Telnet data that is not terminal-to-computer. This occurs, for example, when a Telnet server sends output or when the Telnet protocol incorporates another application protocol.

The **telnet** command "execs" (using the **exec** command) the **/usr/sbin/login** command to validate a user. This 1) allows all user and device attributes to take effect on telnet connections and 2) causes telnet connections to count against the maximum number of login sessions allowable at a time (determined by the **maxlogins** attribute). Attributes are defined in the **/etc/security/user** and **/etc/security/login.cfg** files.

Restrictions

- Earlier versions of the **telnet** command are not compatible with AIX Version 4 and later of the **telnet** command in sending escapes that emulate a high function terminal (HFT). The present version of the **telnet** command sends only one escape when the escape key is hit, while prior versions send two escape characters.
- The **telnet** command must allow transmission of 8-bit characters that are not in binary mode to implement ISO 8859 Latin code page. This is necessary for internationalization of the TCP/IP commands.
- In order to support new character sets, the following was added to the **hft-m**, **ibm5081**, **hft**, **hft-nam**, **hft-c**, **aixterm-m**, and **aixterm** entries in the **terminfo** file:

```
box1=\154\161\153\170\152\155\167\165\166\164\156,      batt1=f1,
box2=\154\161\153\170\152\155\167\165\166\164\156,      batt2=f1md,
font0=\E(B,          font1=\E(0,
```

- The **rlogind** and **telnetd** daemons use POSIX line discipline to change the line discipline on the local TTY. If POSIX line discipline is not used on the local TTY, echoing other line disciplines may result in improper behavior. AIX TCP/IP must have POSIX line discipline to function properly.
- The mouse cannot be used as an input device with the **telnet** command.
- The **telnet** command does not support the APL data stream.

Environment Variables

The following environment variables can be used with the **telnet** command:

Item	Description
EMULATE	Overrides terminal-type negotiation in the same way as the -e flag. If the value of the EMULATE environment variable is defined as vt100 or 3270 , the telnet command emulates a DEC VT100 terminal or 3270 terminal, respectively. If the EMULATE variable is not defined or has a value of none , the telnet command operates normally. If the EMULATE variable is set to vt100 or 3270 , the TERM environment variable in the remote login connection should be set to the same value. You can check this by using the env command after the connection is open.
TNESC	Specifies an alternate TELNET escape character, other than the default, Ctrl-] for the telnet command, Ctrl-T for the tn command, or Ctrl-C for the tn3270 command. To change the telnet escape sequence, set TNESC to the octal value of the character you want to use. Then export TNESC . For example, set TNESC to 35 to change the TELNET escape sequence to Ctrl-].
MAP3270	Specifies an alternate file that contains the user's 3270 keyboard mapping. The MAP3270 variable must contain the full path name to the alternate file. Create the alternate file using the same format as the default /etc/map3270 file.
RM_HOST_LANG	Specifies the EBCDIC code page being used on the remote 3270 host. Set the RM_HOST_LANG environment variable to the correct code page before you telnet (using the telnet command) to a non-English-speaking 3270 host. The default is English. Refer to the Converters Overview for Programming in Globalization Guide and Reference for possible code pages to use. Format the RM_HOST_LANG environment variable by specifying the desired code page. Restriction: The tn3270 command does not support DBCS, because terminal types for DBCS are not supported. The telnet command converts characters by using the iconv command. Users can change the default conversion tables by using the genxlt command.

Flags

Item	Description
-d	Turns debugging mode on.
-e <i>TerminalType</i>	Overrides terminal-type negotiation. Possible values are vt100 , 3270 , or none .
-n <i>TraceFile</i>	Records network trace information in the file specified by the <i>TraceFile</i> variable.
-p	Preserves current TTY attributes.
-f	Causes the credentials to be forwarded. This flag will be ignored if Kerberos 5 is not the current authentication method. Authentication will fail if the current DCE credentials are not marked forwardable.
-F	Causes the credentials to be forwarded. In addition, the credentials on the remote system will be marked forwardable (allowing them to be passed to another remote system). This flag will be ignored if Kerberos 5 is not the current authentication method. Authentication will fail if the current DCE credentials are not marked forwardable.
-k <i>realm</i>	Allows the user to specify the realm of the remote station if it is different from the local systems realm. For these purposes, a <i>realm</i> is synonymous with a DCE cell. This flag will be ignored if Kerberos 5 is not the current authentication method.
-l <i>user</i>	Specifies the remote user the telnet wants to login as. This option is ignored if Kerberos 5 is not the current authentication method.

Subcommands

Before entering each subcommand, press the escape key sequence. The escape sequence tells the program that non-text information follows. Otherwise, the program interprets subcommands as text.

For each of the subcommands in the following list, you only need to type enough letters to uniquely identify the subcommand. (For example, **q** is sufficient for the **quit** subcommand.) This is also true for the arguments to the **display**, **emulate**, **mode**, **set**, and **toggle** subcommands.

The **telnet** subcommands are:

Item	Description
? [<i>Subcommand</i>]	Requests help on telnet subcommands. Without arguments, the ? subcommand prints a help summary. If a <i>Subcommand</i> variable is specified, help information is displayed for that subcommand.
close	Closes the TELNET connection and returns to telnet command mode when the open subcommand is used to establish the connection. When the telnet command is invoked and a host is specified, the close subcommand closes the TELNET connection and exits the telnet program (identical to the quit subcommand).
display [<i>Argument</i>]	Displays all of the set and toggle values if no <i>Argument</i> variable is specified; otherwise, lists only those values that match the <i>Argument</i> variable.

Item	Description
emulate <i>TerminalType</i>	Overrides terminal-type negotiation with the specified terminal type. Possible choices are: ? Prints help information. 3270 Emulates a 3270 terminal. none Specifies no emulation. vt100 Emulates a DEC VT100 terminal.

All output received from the remote host is processed by the specified emulator. The initial terminal type to emulate can be specified through the **EMULATE** environment variable or the **-e** flag to the **telnet** command.

Restriction: Only standard ASCII characters are allowed in emulation mode.

Item	Description
mode <i>Type</i>	Specifies the current input mode. When the <i>Type</i> variable has a value of line , the mode is line-by-line. When the <i>Type</i> variable has a value of character , the mode is character-at-a-time. Permission is requested from the remote host before entering the requested mode, and if the remote host supports it, the new mode is entered.
open <i>Host</i> [<i>Port</i>]	Opens a connection to the specified host. The <i>Host</i> specification can be either a host name or an Internet address in dotted-decimal form. If no <i>Port</i> variable is specified, the telnet subcommand attempts to contact a TELNET server at the default port.
quit	Closes a TELNET connection and exits the telnet program. A Ctrl-D in command mode also closes the connection and exits.

Item	Description
send <i>Arguments</i>	Sends one or more arguments (special character sequences) to the remote host. Multiple arguments are separated by spaces. The following arguments can be used:
?	Prints help information for the send subcommand.
ao	Sends the TELNET AO (Abort Output) sequence, which causes the remote host to flush all output from the remote system to the local terminal.
ayt	Sends the TELNET AYT (Are You There) sequence, to which the remote system can respond.
brk	Sends the TELNET BRK (Break) sequence, which causes the remote system to perform a kill operation.
ec	Sends the TELNET EC (Erase Character) sequence, which causes the remote host to erase the last character entered.
el	Sends the TELNET EL (Erase Line) sequence, which causes the remote system to erase the line currently being entered.
escape	Sends the current telnet escape character. The default escape sequence is Ctrl-] for the telnet command, Ctrl-T for the tn command, or Ctrl-C for the tn3270 command.
ga	Sends the TELNET GA (Go Ahead) sequence, which provides the remote system with a mechanism to signal the local system to return control to the user.
ip	Sends the TELNET IP (Interrupt Process) sequence, which causes the remote system to cancel the currently running process.
nop	Sends the TELNET NOP (No Operation) sequence.
sak	Sends the TELNET SAK (Secure Attention Key) sequence, which causes the remote system to invoke the trusted shell. If the SAK is not supported, then an error message is displayed that reads: Remote side does not support SAK.
synch	Sends the TELNET SYNC sequence, which causes the remote system to discard all previously typed input that has not yet been read. This sequence is sent as TCP/IP urgent data.

Item	Description
set <i>VariableValue</i>	Sets the specified TELNET variable to the specified value. The special value off turns off the function associated with the variable entered. The display subcommand can be used to query the current setting of each variable. The variables that can be specified are: echo Toggles between local echo of entered characters and suppressing local echo. Local echo is used for normal processing, while suppressing the echo is convenient for entering text that should not be displayed on the screen, such as passwords. This variable can only be used in line-by-line mode. eof Defines the character for the telnet command. When the telnet command is in line-by-line mode, entering the eof character as the first character on a line sends the character to the remote host. The initial value for the eof character is the local terminal End-Of-File character. erase Defines the erase character for the telnet command. When the telnet command is in character-at-a-time mode and localchars has a value of true, typing the erase character sends the TELNET EC sequence to the remote host. The initial value for the erase character is the local terminal ERASE character. escape Specifies the telnet escape character, which puts the telnet command into command mode when connected to a remote host. This character can also be specified in octal in the TNESC environment variable. flushoutput Defines the flush character for the telnet command. When localchars has a value of true, typing the flushoutput character sends the TELNET AO sequence to the remote host. The initial value for the flush character is Ctrl-O. If the remote host is running AIX, the flushoutput variable, unlike the other special characters defined by the set subcommand, only works in localchars mode since it has no termio equivalent. interrupt Defines the interrupt character for the telnet command. When localchars has a value of true, typing the interrupt character sends the TELNET IP sequence to the remote host. The initial value for the interrupt character is the local terminal interrupt (INTR) character. kill Defines the kill character for the telnet command. When the telnet command is in character-at-a-time mode and localchars has a value of true, typing the kill character sends the TELNET EL sequence to the remote host. The initial value for the kill character is the local terminal KILL character. quit Defines the quit character for the telnet command. When localchars has a value of true, typing the quit character sends the TELNET BRK sequence to the remote host. The initial value for the quit character is the local terminal QUIT character. sak Defines the Secure Attention Key (SAK) for the telnet command. When the sak character is entered, the remote system is asked to create a trusted shell. If the remote host does not support the SAK, this sequence has no effect.
status	Shows the status of the telnet command, including the current mode and the currently connected remote host.

Item	Description
toggle <i>Arguments</i>	Toggles one or more arguments that control how the telnet command responds to events. Possible values are true and false. Multiple arguments are separated by spaces. The display subcommand can be used to query the current setting of each argument. The following arguments can be used: ? Displays valid arguments to toggle. autoflush If autoflush and localchars both have a value of true and the AO, INTR, and QUIT characters are recognized and transformed into TELNET sequences, the telnet command does not display any data on the user's terminal until the remote system acknowledges (with a TELNET timing mark option) that it has processed those TELNET sequences. The initial value of autoflush is true if the terminal has not done an stty noflsh, and false if it has. autosynch If autosynch and localchars are both true, then typing the INTR or QUIT character sends that character's TELNET sequence, followed by the TELNET SYNC sequence. This procedure causes the remote host to discard all previously typed input until both of the TELNET sequences have been read and acted upon. The initial value of this toggle is false. crmod Toggles carriage return mode. When set to true, most carriage return characters received from the remote host are mapped into a carriage return followed by a line feed. This mode does not affect the characters typed by the user, only those received from the remote host. This mode is useful when the remote host sends only a carriage return and not a line feed. The initial value of this toggle is false. debug Toggles debugging at the socket level. The initial value of this toggle is false. localchars Determines the handling of TELNET special characters. When this value is true, the ERASE, FLUSH, INTERRUPT, KILL, and QUIT characters are recognized locally and transformed into the appropriate TELNET control sequences (EC, AO, IP, BRK, and EL, respectively). When this value is false, these special characters are sent to the remote host as literal characters. The initial value of localchars is true in line-by-line mode and false in character-at-a-time mode. netdata Toggles the display of all network data (in hexadecimal format). The data is written to standard output unless a <i>TraceFile</i> value is specified with the -n flag on the telnet command line. The initial value of this toggle is false. options Toggles the display of internal TELNET Protocol processing options, such as terminal negotiation and local or remote echo of characters. The initial value of this toggle is false, indicating that the current options should not be displayed. lineterm Toggles the default end-of-line terminator to CR-LF (ASCII carriage-return line-feed). A telnet client running on an ASCII host should have the user configurable option to send either the CR-NUL or CR-LF terminator when the user presses the end-of-line key. The initial value of this toggle is false.

Item	Description
z	Suspends the TELNET process. To return to the TELNET process, use the fg built-in command of the csh or ksh command. Note: The z subcommand has the same effect as a Ctrl-Z key sequence for any other process. It suspends Telnet execution and returns you to your original login shell.

Authentication

If the system is configured for Kerberos 5 authentication, the telnet client will attempt authentication negotiation. The authentication negotiation used by telnet and the definitions of the options and suboptions for this are defined in rfc 1416.

If the client and server agree on an authentication type, they will exchange authentication information including the account the client wants to access. This will be the local user unless the **-l** flag is set.

If they cannot agree on the authentication information or if it fails, the telnet connection will continue with the standard connection (provided Standard AIX is configured).

The remote host allows access only if all of the following conditions are satisfied:

- The local user has current DCE credentials.
- The remote system accepts the DCE credentials as sufficient for access to the remote account. See the **kvalid_user** function for additional information.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

In the following examples, if you enter the **tn** command instead of the **telnet** command, the command mode prompt is displayed as tn>.

1. To log in to the remote host **host1** and perform terminal negotiation, enter:

```
telnet host1
```

2. To log in to **host1** as a **vt100** terminal (no terminal type negotiation), choose one of the following methods:

- a. Use the following commands to set the **EMULATE** environment variable for this login session, then enter the **telnet** command:

```
EMULATE=vt100; export EMULATE
telnet host1
```

- b. Use the **-e** flag to set the terminal type for this **telnet** session only:

```
telnet -e vt100 host1
```

3. To log in to a remote host and then check the status of the **telnet** program, enter:

```
telnet host3
```

When the login prompt appears, enter your login ID and password. Press the Ctrl-T key sequence to receive the **telnet>** prompt. Enter the following at the **telnet>** prompt:

```
status
```

Information similar to the following is displayed on your screen:

```
Connected to host3.  
Operating in character-at-a-time mode.  
Escape character is '^['.
```

Upon completion of the **status** subcommand, press the Enter key to return to the remote prompt.

Once you have completed your login, you can issue commands. To log out of the system and close the connection, press the Ctrl-D key sequence, or exit.

4. To log in to a remote host using the **tn3270** command, enter:

```
tn3270 hostname
```

The host login screen should be displayed. You can now enter your login ID and password. Once you have completed your login, you can issue commands. To log out of the system and close the connection, press Ctrl-D or exit.

5. To connect to the **icehouse.austin.ibm.com** remote host with the **telnet** command with a user name david of specific SLs sec a b, enter the following commands:
 - a. In the command line, enter `telnet icehouse.aoot.austin.ibm.com` to connect to the **icehouse.austin.ibm.com**
 - b. In the login field, enter `david -e "sec a b"`
 - c. In the passwords field, enter david's passwords.

To disconnect from the remote server, use the **Ctrl-T** key sequence.

Files

Item	Description
/etc/3270.keys	Defines base 3270-keyboard mapping for use with limited function terminals.

telnetd Daemon

Purpose

Provides the server function for the TELNET protocol.

Syntax

/usr/sbin/telnetd [-a] [-c] [-n] [-s]

Description

Note: The **telnetd** daemon is normally started by the **inetd** daemon. It can also be controlled from the command line, using SRC commands.

The **/usr/sbin/telnetd** daemon is a server that supports the Defense Advanced Research Product Agency (DARPA) standard Telnet Protocol (TELNET). Changes to the **telnetd** daemon should be made using the System Management Interface Tool (SMIT).

Changes to the **telnetd** daemon can be made using the System Management Interface Tool (SMIT) or System Resource Controller (SRC), by editing the **/etc/inetd.conf** or **/etc/services** file. Typing **telnetd** at the command line is not recommended. The **telnetd** daemon is started by default when it is uncommented in the **/etc/inetd.conf** file. By default, the **-a** flag is also turned on.

The **inetd** daemon get its information from the **/etc/inetd.conf** file and the **/etc/services** file.

After changing the **/etc/inetd.conf** or **/etc/services** file, run the **refresh -s inetd** or **kill -1 InetdPID** command to inform the **inetd** daemon of the changes to its configuration file.

When a **telnet** session is started, the **telnetd** daemon sends TELNET options to the client (remote) host to indicate an ability to perform options.

Terminal Negotiation

The **telnetd** daemon requests the terminal type from the client host. On receipt, the **telnetd** daemon checks whether the indicated type is supported on the local system. If not, the daemon requests a terminal type again.

This terminal type negotiation continues until the remote client sends an acceptable terminal type or until the client sends the same type twice in a row, indicating that it has no other types available. When necessary, the **telnetd** daemon refers to the **/etc/telnet.conf** file to translate a client's terminal-type strings into **terminfo** file entries.

Note: Because the **telnetd** daemon allows the sending and receiving of 8-bit ASCII, NLS is supported.

If the remote client sends the TELNET **SAK** command, the **telnetd** daemon passes the local SAK characters through the PTY to invoke the trusted shell.

The **telnetd** daemon supports the following telnet options:

- Binary
- Echo/no echo
- Support SAK
- Suppress go ahead
- Timing mark
- Negotiate About Window Size (NAWS)
- Authentication

The **telnetd** daemon also recognizes the following options for the remote client:

- Binary
- Suppress go ahead
- Echo/no echo
- Terminal type

The **telnetd** daemon should be controlled using the System Management Interface Tool (SMIT) or by changing the **/etc/inetd.conf** file. Typing **telnetd** at the command line is not recommended.

Authentication Negotiation

If the system has Kerberos 5 authentication configured, **telnetd** will accept authentication option negotiation. If both agree on Kerberos 5 authentication, the client will pass over the DCE principal and **telnetd** will use the **kvalid_user** routine to determine if the DCE principal should have access to the account. If it passes, no password will be requested.

Manipulating the telnetd Daemon with the System Resource Controller

The **telnetd** daemon is a subserver of the **inetd** daemon, which is a subsystem of the System Resource Controller (**SRC**). The **telnetd** daemon is a member of the **tcpip** SRC subsystem group. This daemon is enabled by default in the **/etc/inetd.conf** file and can be manipulated by the following SRC commands:

Item	Description
startsrc	Starts a subsystem, group of subsystems, or a subserver.
stopsrc	Stops a subsystem, group of subsystems, or a subserver.
lssrc	Gets the status of a subsystem, group of subsystems, or a subserver.

Flags

Item	Description
-a	Causes the PTY and socket to be linked directly in the kernel so that the data handling remains in the kernel to improve the performance.
-c	Suppresses the reverse host name lookup.
-n	Disables transport-level keep-alive messages. Messages are enabled by default.
-s	Turns on socket-level debugging.

Note: Unrecognized flags will be ignored by the daemon and logged to the syslog if syslog is enabled.

Security

The **telnetd** daemon is a PAM-enabled application with a service name of *telnet*. System-wide configuration to use PAM for authentication is set by modifying the value of the **auth_type** attribute, in the **usw** stanza of **/etc/security/login.cfg**, to PAM_AUTH as the root user.

The authentication mechanisms used when PAM is enabled depend on the configuration for the **telnet** service in **/etc/pam.conf**. The **telnetd** daemon requires **/etc/pam.conf** entries for the **auth**, **account**, **password**, and **session** module types. Listed below is a recommended configuration in **/etc/pam.conf** for the **telnet** service:

```
#
# AIX telnet configuration
#
telnet auth      required    /usr/lib/security/pam_aix
telnet account   required    /usr/lib/security/pam_aix
telnet password  required    /usr/lib/security/pam_aix
telnet session   required    /usr/lib/security/pam_aix
```

Examples

Note: The arguments for the **telnetd** daemon can be specified by using SMIT or by editing the **/etc/inetd.conf** file.

1. To start the **telnetd** daemon, type the following:

```
startsrc -t telnet
```

This command starts the **telnetd** subserver.

2. To stop the **telnetd** daemon normally, type the following:

```
stopsrc -t telnet
```

This command allows all pending connections to start and existing connections to complete but prevents new connections from starting.

3. To force stop the **telnetd** daemon and all **telnetd** connections, type the following:

```
stopsrc -f -t telnet
```

This command terminates all pending connections and existing connections immediately.

4. To display a short status report about the **telnetd** daemon, type the following:

```
lssrc -t telnet
```

This command returns the daemon's name, process ID, and state (active or inactive).

File

Item	Description
terminfo	Describes terminal by capability.

termdef Command

Purpose

Queries terminal characteristics.

Syntax

termdef [**-c** | **-l** | **-t**]

Description

The **termdef** command identifies the current display type, the active lines setting, or the current columns setting. This simplifies resetting the lines and columns when you switch fonts as well as resetting the **TERM** environment variable when you switch displays. The **terminfo** database defines the default number of lines and columns for each display, but the lines and columns can change depending upon which font is currently active. Also, the **TERM** environment variable does not automatically reflect the currently active display.

The flags for the **termdef** command are mutually exclusive. If you use more than one flag with the command, the **termdef** command recognizes and returns the current value for the first flag only. Any other flags are ignored. For example, the **termdef -lc** command returns only the active lines setting for the current display.

Flags

Item	Description
-c	Returns the current column value.
-l	Returns the current line value.
-t	Returns the name of the current display (the default action).

Example

To determine the current value of the **TERM** environment variable, enter:

```
termdef -c
```

File

Item	Description
/usr/bin/termdef	Contains the termdef command.

test Command

Purpose

Evaluates conditional expressions.

Syntax

test *Expression*

OR

[*Expression*]

Description

The **test** command evaluates the *Expression* parameter, and if the expression value is True, returns a zero (True) exit value. Otherwise, the **test** command returns a nonzero (False) exit value. The **test** command also returns a nonzero exit value if there are no parameters.

Requirements:

- In the second form of the command, the [] (brackets) must be surrounded by blank spaces.
- You must test explicitly for file names in the C shell. File-name substitution (globbing) causes the shell script to exit.

Functions and operators are treated as separate parameters by the **test** command. The *Expression* parameter refers to a statement that is checked for a true or false condition. The following functions are used to construct this parameter:

Item	Description
-b <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists and is a block special file.
-c <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists and is a character special file.
-d <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists and is a directory.
-e <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists.
-f <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists and is a regular file.
-g <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists and its Set Group ID bit is set.
-h <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists and is a symbolic link.
-k <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists and its sticky bit is set.
-L <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists and is a symbolic link.
-n <i>String1</i>	Returns a True exit value if the length of the <i>String1</i> variable is nonzero.
-p <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists and is a named pipe (FIFO).
-r <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists and is readable by the current process.
-s <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists and has a size greater than 0.
-t <i>FileDescriptor</i>	Returns a True exit value if the file with a file descriptor number of <i>FileDescriptor</i> is open and associated with a terminal.

Item	Description
-u <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists and its Set User ID bit is set.
-w <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists and the write flag is on. However, the <i>FileName</i> will not be writable on a read-only file system even if test indicates true.
-x <i>FileName</i>	Returns a True exit value if the specified <i>FileName</i> exists and the execute flag is on. If the specified file exists and is a directory, the True exit value indicates that the current process has permission to search in the directory.
-z <i>String1</i>	Returns a True exit value if the length of the <i>String1</i> variable is 0 (zero).
<i>String1</i> = <i>String2</i>	Returns a True exit value if the <i>String1</i> and <i>String2</i> variables are identical.
<i>String1</i> != <i>String2</i>	Returns a True exit value if the <i>String1</i> and <i>String2</i> variables are not identical.
<i>String1</i>	Returns a True exit value if the <i>String1</i> variable is not a null string.
<i>Integer1</i> -eq <i>Integer2</i>	Returns a True exit value if the <i>Integer1</i> and <i>Integer2</i> variables are algebraically equal. Any of the comparisons -ne , -gt , -ge , -lt , and -le can be used in place of -eq .
<i>file1</i> -nt <i>file2</i>	True if <i>file1</i> is newer than <i>file2</i> .
<i>file1</i> -ot <i>file2</i>	True if <i>file1</i> is older than <i>file2</i> .
<i>file1</i> -ef <i>file2</i>	True if <i>file1</i> is another name for <i>file2</i> .

These functions can be combined with the following operators:

Item	Description
!	Unary negation operator
-a	Binary AND operator
-o	Binary OR operator (that is, the -a operator has higher precedence than the -o operator)
\(Expression\)	Parentheses for grouping

Exit Status

This command returns the following exit values:

Ite m	Description
0	The <i>Expression</i> parameter is true.
1	The <i>Expression</i> parameter is false or missing.
>1	An error occurred.

Examples

1. To test whether a file exists and is not empty, enter the following command:

```
if test ! -s "$1"
then
```

```
    echo $1 does not exist or is empty.
fi
```

If the file specified by the first positional parameter to the shell procedure, **\$1**, does not exist, the **test** command displays an error message. If **\$1** exists and has a size greater than 0, the **test** command displays nothing.

Note: There must be a space between the **-s** function and the file name.

The quotation marks around **\$1** ensure that the test works properly even if the value of **\$1** is a null string. If the quotation marks are omitted and **\$1** is the empty string, the **test** command displays the error message `test: argument expected`.

2. To do a complex comparison, type:

```
if [ $# -lt 2 -o ! -e "$1" ]
then
    exit
fi
```

If the shell procedure is given fewer than two positional parameters or the file specified by **\$1** does not exist, then the shell procedure exits. The special shell variable **\$#** represents the number of positional parameters entered on the command line that starts this shell procedure.

The **Shells** in *Operating system and device management* describes shells in general, defines terms that are helpful in understanding shells, and describes the more useful shell functions.

File

Item	Description
<code>/usr/bin/test</code>	Contains the test command.

tetoldif Command

Purpose

Prints certain Trusted Signature Database (TSD) and TE Policies that are defined locally to **stdout** in an ldif format.

Syntax

```
tetoldif -d < baseDN > [-s [ filename ] ] [-p [ filename ] ]
```

Description

The **tetoldif** command reads data from a locally defined TSD and TE policies database files and prints the result to **stdout** in ldif format. If the results are redirected to a file, they can be added to a LDAP server with the **ldapadd** command with the **-b** flag or the **ldif2db** command.

The **tetoldif** command reads the `/etc/security/ldap/sectoldif.cfg` file to determine what to name the trusted signature database and the TE policies database sub-trees where the data is exported to. The **tetoldif** command only exports data to the TSDDAT types and TEPOLICIES types defined in the `/etc/security/ldap/sectoldif.cfg` file. The names specified in the `/etc/security/ldap/sectoldif.cfg` file will be used to create sub-trees under the base distinguished name (DN) specified with the **-d** flag.

The **tetoldif** command reads the Trusted Execution LDAP database reference names from the `/etc/nscontrol.conf` file if it is present. If the specified names are unavailable in the `/etc/nscontrol.conf` file, then the default names will be used. The default names are *TSD* for the TSD and *TEPOL* for the TE Policy.

Flags

Item	Description
-d < <i>BaseDN</i> >	Specifies the base distinguished names (DN) under which to place the TSD and TE policies data. For example, <i>cn=aixdata</i> .
-s [<i>filename</i>]	Specifies the signature database. It will print only the TSD database to ldif format. If the filename is used, the default TSD /etc/security/tsd/tsd.dat data file can be changed to the filename.
-p [<i>filename</i>]	Specifies the TE policies database. It will print only the TE policies database to LDIF format. If the filename is used, the default TE Policies //etc/security/tsd/tepolices.dat file is changed to the filename.

Exit Status

Item	Description
0	Successful completion.
>0	An error occurred.

Security

Access Control: This command should grant execute (x) access only to the root user.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Files:

Item	Description
/etc/security/tsd/tsd.dat	Contains the TSD attributes for the binaries which are configured.
/etc/security/tsd/tepolices.dat	Contains the TE policies configured.

Examples

1. To export the TSD and TE policies database content to a ldif format with the base DN of cn=aixdata, run the following command:

```
tetoldif -d cn=aixdata
```

2. To export only a TSD database to a ldif format with the base DN of cn=aixdata, run the following command:

```
tetoldif -d cn=aixdata -s
```

3. To export only a TE policies database content to a ldif format with the base DN of cn=aixdata, run the following command:

```
tetoldif -d cn=aixdata -p
```

4. To export only a TSD database from a different file than the default **/etc/security/tsd/tepolices.dat** file to a ldif format with the base DN of cn=aixdata, run the following command:

```
tetoldif -d cn=aixdata -s filename
```

5. To export TE policies from a different file than the default **/etc/security/tsd/tepolicies.dat** file to a ldif format with the base DN of cn=aixdata, run the following command:

```
tetoldif -d cn=aixdata -p filename
```

tftp or utftp Command

Purpose

Transfers files between hosts using the Trivial File Transfer Protocol (TFTP).

Syntax

```
{tftp | utftp} { -g | -o | -p | -r | -w } LocalName HostPort RemoteName [ netascii | image ] [blksize #] [timeout #] [tsize]
```

Interactive Form Syntax

Command Line Form Syntax

Description

The **/usr/bin/tftp** and **utftp** commands transfer files between hosts using the Trivial File Transfer Protocol (TFTP). Since TFTP is a minimal file transfer protocol, the **tftp** and **utftp** commands do not provide all of the features of the **ftp** command. For example, the **tftp** and **utftp** commands do not provide the ability to list remote files or change directories at the remote host, and only limited file access privileges are given to the remote TFTP server. The **utftp** command is a form of the **tftp** command for use in a pipe.

The remote host must have a **tftpd** daemon started by its **inetd** daemon and have an account defined that limits the access of the **tftpd** daemon. Use the procedure defined by the **tftpd** command to setup the TFTP environment and the nobody account.

Note: The **tftp** and **utftp** commands should not be available when your host is operating in secure mode.

The **tftp** command ignores duplicate acknowledgments for any block sent and sends an error packet and exit if a block with an inappropriate (future) block number arrives. It also ignores duplicate data blocks if they have already been received and sends an error packet and exits.

RFC2349 Option Negotiation

The **tftp** client is capable of negotiating the following TFTP options with the server: block size (blksize), transfer size (tsize), and timeout (timeout). Larger transfer block size can improve transfer performance, tsize reports the file size before the transfer to check for available space, and timeout negotiates the retransmit timeout. The TFTP server must support RFC2349 for option negotiation to take place.

Access Control

The **/etc/tftpaccess.ctl** file is searched for lines that start with **allow:** or **deny:**. Other lines are ignored. If the file doesn't exist, access is allowed. The allowed directories and files can be accessed and the denied directories cannot be accessed. For example, the **/usr** directory might be allowed and the **/usr/ucb** directory might be denied. This means that any directory or file in the **/usr** directory, except the **/usr/ucb** directory, can be accessed. The entries in the **/etc/tftpaccess.ctl** file must be absolute path names.

The **/etc/tftpaccess.ctl** file should be write-only by the root user and readable by all groups and others (that is, owned by root with permissions of 644). The user nobody must be able to read the **/etc/tftpaccess.ctl** file. Otherwise, the **tftpd** daemon is not able to recognize the existence of the file and allows access to the entire system. For more information, refer to the sample **tftpaccess.ctl** file, which resides in the **/usr/samples/tcpip** directory.

The search algorithm assumes that the local path name used in the **tftp** command is an absolute path name. It searches the **/etc/tftpaccess.ctl** file looking for **allow: /**. It repeatedly searches for allowed path names with each partial path name constructed by adding the next component from the file path name. The longest path name matched is the one allowed. It then does the same with denied names, starting with the longest allowed path name matched.

For example, if the file path name were **/a/b/c** and the **/etc/tftpaccess.ctl** file contained **allow: /a/b** and **deny: /a**, one allowed match would be made (**/a/b**) and no denied match starting with **/a/b** would be made, and access would be allowed.

If the **/etc/tftpaccess.ctl** file contained **allow: /a** and **deny: /a/b**, one allowed match would be made (**/a**) and one denied match starting with **/a** (**/a/b**) would be made, and access would be denied. If the **/etc/tftpaccess.ctl** file contained **allow: /a/b** and also contained **deny: /a/b**, access would be denied because allowed names are searched first.

Note: Further information and example configurations for Xstations, Diskless clients, and restricted entry can be found in the **/usr/samples/tcpip/tftpaccess.ctl** file.

The **tftp** and **utftp** commands have two forms: interactive form and command-line form.

Interactive Form

In the interactive form, the **tftp** and **utftp** commands are issued alone or with a *Host* parameter that specifies the default host to use for file transfers during this session. If you choose, you can also specify with the *Port* parameter which port the **tftp** or **utftp** connection should use, such as the one specified for **mail** in the **/etc/services** file. When you enter the interactive form of either of these commands, the **tftp>** prompt is displayed.

When transferring data to a remote host, the transferred data is placed in the directory specified by the *RemoteName* parameter. The remote name must be a fully specified file name, and the remote file must both exist and have write permission set for others. The **tftp** command attempts to write the data to the specified file. However, if the remote TFTP server does not have the appropriate privileges to write the remote file or if the file does not already exist, the transfer is unsuccessful. This can be overridden using the **tftpd** daemon.

Command-Line Form

The command-line forms of the **tftp** and **utftp** commands are equivalent, except that the **utftp** command does not overwrite a local file. The **tftp** command can overwrite a file, but prompts the user before doing so. Because it is not interactive, the command line form of the **utftp** command can be more useful than the **tftp** command in a pipe. In the command line form, all of the arguments to either command are specified on the command line, and no prompt is displayed.

Subcommands

The **tftp** and **utftp** subcommands can be entered in either their interactive form or in their command-line form.

Subcommands Used in the Interactive Form

Once the **tftp>** prompt is displayed, the following subcommands can be issued:

Item	Description
? [Subcommand]	Displays help information. If a <i>Subcommand</i> parameter is specified, only information about that subcommand is displayed.
ascii	Synonym for the mode ascii subcommand.
binary	Synonym for the mode binary subcommand. This subcommand is used in the interactive mode. The image subcommand accomplishes the same thing as the mode binary subcommand, but is used on the command line.

Item	Description
blksize <i>Number of Bytes</i>	Enables the blksize option negotiation with the server. If successfully negotiated, this can substantially improve transfer rates. The transfer block size must be at least 8 octets and can be as high as 65464 octets. The default is 512 octets.
connect <i>Host [Port]</i>	Sets the remote host, and optionally the port, for file transfers. Since the TFTP protocol does not maintain connections between transfers, the connect subcommand does not create a connection to the specified host, but stores it for transfer operations. Because the remote host can be specified as part of the get or put subcommand, which overrides any host previously specified, the connect subcommand is not required.

get *RemoteFile [LocalFile]*

Item	Description
get <i>RemoteFile RemoteFile RemoteFile [RemoteFile . . .]</i>	Gets a file or set of files from the remote host to the local host. Each of the <i>RemoteFile</i> parameters can be specified in one of the following two ways: • As a file (<i>File</i>) that exists on the remote host if a default host has already been specified. • As a host file (<i>Host:File</i>), where <i>Host</i> is the remote host and <i>File</i> is the name of the file to copy to the local system. If this form of the parameter is used, the last host specified becomes the default host for later transfers in this tftp session.
mode <i>Type</i>	Sets the type (<i>Type</i>) of transfer mode to either ascii or binary . A transfer mode of ascii is the default.

put *LocalFile [RemoteFile]*

Item	Description
put <i>LocalFile LocalFile LocalFile [LocalFile . . .] RemoteDirectory</i>	Puts a file or set of files from the local host onto the remote host. The <i>RemoteDirectory</i> and <i>RemoteFile</i> parameters can be specified in one of the following two ways: • As a file or directory that exists on the remote host if a default host has already been specified. • With <i>Host:RemoteFile</i> parameter, where <i>Host</i> is the remote host and <i>RemoteFile</i> is the name of the file or directory on the remote system. If this form of the parameter is used, the last host specified becomes the default host for later transfers in this tftp session. In either case, the remote file or directory name must be a fully specified path name, even if the local and remote directories have the same name. If a remote directory is specified, the remote host is assumed to be a UNIX machine. The default value of the put subcommand is write-replace, but you can add an option in the tftpd daemon to allow write-create.

Item	Description
quit	Exits the tftp session. An End-Of-File key sequence also exits the program.
status	Shows the current status of the tftp program, including, for example, the current transfer mode (ascii or binary), connection status, and time-out value.
timeout <i>Value</i>	Sets the total transmission time out to the number of seconds specified by the <i>Value</i> parameter. The <i>Value</i> parameter must be 1 second or greater (the default is 5 seconds).
trace	Turns packet tracing on or off.
tsize	Enables the tsize option negotiation with the server. This allows the file size to be known before the transfer starts. If allocation is exceeded, an error is returned and the file transfer does not occur.
verbose	Turns verbose mode, which displays additional information during file transfer, on or off.

Subcommands Used in the Command Line Form

In this form, if the *Action* flag is:

Item	Description
-w or -p	Writes (or puts) local data, specified by the <i>LocalName</i> parameter, to the file specified by the <i>RemoteName</i> parameter on the remote host specified by the <i>Host</i> parameter. If the <i>LocalName</i> parameter is a file name, the tftp command transfers the specified local file. If the <i>LocalName</i> parameter is specified as a - (dash), the tftp command transfers data from local standard input to the remote host. When the <i>LocalName</i> parameter is standard input, the tftp command allows 25 seconds for all input to be entered before it times out.
-r or -g or -o	Reads (or gets) remote data from the file specified by the <i>RemoteName</i> parameter at the remote host specified by the <i>Host</i> parameter and writes it to the file specified by the <i>LocalName</i> parameter. If the <i>LocalName</i> parameter is a file name, the tftp command writes the data to the specified local file. For the -r and -g actions, the tftp command prompts for verification before overwriting an existing local file. For the -o action, the tftp command overwrites an existing local file without prompting. If the <i>LocalName</i> parameter is specified as a - (dash), the tftp command writes the data to local standard output. Note: Since the tftp -g and tftp -r commands prompt before overwriting an existing local file, it may be impractical to use the tftp command in a pipe. The utftp command performs the same -r and -g actions as the tftp command, but simply stops before overwriting a local file. Thus, the utftp command may be more appropriate for use in a pipe.

For both of the following modes of file transfer, the *RemoteName* parameter is the name of a file that has write permission set for others. Note that the *RemoteName* parameter must be in double quotes (" ") if it contains shell special characters.

The mode of transfer is one of the following:

Item	Description
netascii	Transfers the data as 7-bit ASCII characters in 8-bit transfer bytes. This is the default.
image	Transfers the data as 8-bit binary data bytes in 8-bit transfer bytes, with no conversion. image transfer can be more efficient than netascii transfer when transferring between two hosts. It is recommended that netascii be used when transferring ASCII files from a workstation to a different type of host.

Examples

The following examples distinguish the differences between the interactive form and the command line form of the **tftp** command:

Using the Interactive Form of the tftp Command

To enter the **tftp** command, check the current status, connect to a remote host, and transfer a file from a remote host to your local host, enter:

```
tftp
```

The **tftp>** prompt is displayed. Enter the **status** subcommand following this prompt:

```
status
```

A message similar to the following is displayed on your screen:

```
Not connected.
Mode: netascii  Verbose: off  Tracing: off
Max-timeout: 25 seconds
tftp> _
```

After the **tftp>** prompt, enter the **connect** subcommand and the name of the remote system to which you want to connect:

```
tftp> connect host1
```

The **tftp>** prompt is displayed as an indication that you are connected to **host1**. Following the **tftp>** prompt, enter the **get** subcommand to transfer the file **update** from the remote host to your local host.

```
get /home/alice/update update
```

The **/home/alice** directory on the remote host must have read permission set for others. The **/home/alice/update** file from **host1** was transferred to the **update** file on your local system. In this example, the user is connected to **host1** and the **update** file is transferred from **host1** to the local host.

Using the Command Line Form of the tftp Command

1. To copy a text file from a remote host and write it to a local file, enter:

```
tftp -g newsched host1 /home/john/schedule
$ _
```

In this example, the **/home/john/schedule** file was copied from the remote host **host1** and written to the local file **newsched**.

2. To copy a file from a remote host and redirect the output to standard output of the local host, enter:

```
tftp -g - host3 /etc/hosts
```

If the copy is successful, information similar to the following is displayed on your screen:

```
192.100.13.3 nameserver
192.100.13.3 host2
192.100.13.5 host1
192.100.13.7 host3
192.100.13.3 timeserver
```

```
Received 128 bytes in 0.4 seconds
$ _
```

In this example, the `/etc/hosts` file from remote host `host3` was copied and the output redirected to standard output of the local host.

3. To copy a file from a remote host, pipe it to the **grep** command, and write it to a local file, enter:

```
utftp -g - host1 /home/john/schedule | grep Jones > jones.todo
$ _
```

In this example, the `/home/john/schedule` file was copied from the remote host `host1`. This file was then piped to the **grep** command and written into the local file `jones.todo`.

4. To copy a file to another system, enter:

```
tftp -p /home/jeanne/test host2 /tmp/test
```

If the copy is successful, information similar to the following is displayed on your screen:

```
Sent 94146 bytes in 6.7 seconds
```

In this example, the `/home/jeanne/test` file was sent to the `/tmp` directory on the remote host `host2`.

5. To copy a binary file to another system, enter:

```
tftp -p core host3 /tmp/core image
```

If the copy is successful, information similar to the following is displayed on your screen:

```
Sent 309295 bytes in 15 seconds
```

In this example, the binary file `core` from the current directory was sent to the `/tmp` directory on remote host `host3`.

Files

Item	Description
<code>/etc/tftpdaccess.ctl</code>	Allows or denies access to files and directories.

tftpd Daemon

Purpose

Provides the server function for the Trivial File Transfer Protocol.

Syntax

```
/usr/sbin/tftpd [ -c ] [ -n ] [ -p ] [ -v ] [ -t ] [ -s ] [ -x ] [ -z ] [ -d Directory ] [ -r Option ]
```

Description

Note: The **tftpd** daemon is normally started by the **inetd** daemon. It can also be controlled from the command line, using SRC commands.

The `/usr/sbin/tftpd` daemon runs the Trivial File Transfer Protocol (TFTP) server. Files sent using TFTP can be found in the directory specified by the full path name given on the **tftp** or **utftp** command line.

Note: The **tftp** command, **utftp** command, and **tftpd** server are not available when the auditing system is in use. For more information, see **TCP/IP Security**, the **Auditing overview**, and the **audit** command.

Changes to the **tftpd** daemon can be made using the System Management Interface Tool (SMIT) or System Resource Controller (SRC), by editing the **/etc/inetd.conf** or **/etc/services** file. The **tftpd** daemon is started by default when it is uncommented in the **/etc/inetd.conf** file.

The **inetd** daemon get its information from the **/etc/inetd.conf** file and the **/etc/services** file.

After changing the **/etc/inetd.conf** or **/etc/services** file, run the **refresh -s inetd** or **kill -1 InetdPID** command to inform the **inetd** daemon of the changes to its configuration file.

The **tftpd** server should have a user ID with the least privileges possible. The **nobody** ID allows the least permissions, and is the default user ID.

The **tftpd** daemon should be controlled using the System Management Interface Tool (SMIT) or by changing the **/etc/inetd.conf** file. Entering **tftpd** at the command line is not recommended.

The **tftpd** server is a multithreaded application and is able to handle option negotiation (RFC2349). This capability allows a client to negotiate a file size to be transferred. It also allows for a timeout and a larger block size. Block size (**blksize**) is negotiated for the read requests (RRQ) only. As a result, the boot time performance of diskless nodes using TFTP can improve significantly.

The Transfer Size option (**tsize**) negotiation for both read and write requests allows the file size to be known before the transfer, resulting in an error message if allocation exceeded before the transfer started. The timeout option (**timeout**) allows for the client and the server to negotiate a retransmit timeout (between 1 and 255 seconds). The **tftp** client must also support RFC2349 for the option negotiation to take place.

tftpdaccess.ctl File

The **/etc/tftpdaccess.ctl** file is searched for lines that start with **allow:** or **deny:**. Other lines are ignored. If the file doesn't exist, access is allowed. The allowed directories and files minus the denied directories and files can be accessed. For example, the **/usr** directory might be allowed and the **/usr/ucb** directory might be denied. This means that any directory or file in the **/usr** directory, except the **/usr/ucb** directory, can be accessed. The entries in the **/etc/tftpdaccess.ctl** file must be absolute path names.

The **/etc/tftpdaccess.ctl** file should be write-only by the root user and readable by all groups and others (that is, owned by root with permissions of 644). The user **nobody** must be able to read the **/etc/tftpdaccess.ctl** file. Otherwise, the **tftpd** daemon is not able to recognize the existence of the file and allows access to the entire system. For more information, refer to the sample **tftpdaccess.ctl** file, which resides in the **/usr/samples/tcpip** directory.

The search algorithm assumes that the local path name used in the **tftp** command is an absolute path name. It searches the **/etc/tftpdaccess.ctl** file looking for **allow:/.** It repeatedly searches for allowed path names with each partial path name constructed by adding the next component from the file path name. The longest path name matched is the one allowed. It then does the same with denied names, starting with the longest allowed path name matched.

For example, if the file path name were **/a/b/c** and the **/etc/tftpdaccess.ctl** file contained **allow:/a/b** and **deny:/a**, one allowed match would be made (**/a/b**) and no denied match starting with **/a/b** would be made, and access would be allowed.

If the **/etc/tftpdaccess.ctl** file contained **allow:/a** and **deny:/a/b**, one allowed match would be made (**/a**) and one denied match starting with **/a** (**/a/b**) would be made, and access would be denied. If the **/etc/tftpdaccess.ctl** file contained **allow:/a/b** and also contained **deny:/a/b**, access would be denied because allowed names are searched first.

Manipulating the tftpd Daemon with the System Resource Controller

The **tftpd** daemon is a subserver of the **inetd** daemon, which is a subsystem of the System Resource Controller (SRC). The **tftpd** daemon is a member of the **tcpip** SRC subsystem group. This daemon is enabled when it is uncommented in the **/etc/inetd.conf** file and can be manipulated by the following SRC commands:

Item	Description
startsrc	Starts a subsystem, group of subsystems, or a subserver.

Item	Description
stopsrc	Stops a subsystem, group of subsystems, or a subserver.
lssrc	Gets the status of a subsystem, group of subsystems, or a subserver.

Flags

Item	Description
-c	Specifies the maximum number of concurrent threads per process, excluding the initial thread.
-d <i>Directory</i>	Specifies default destination directory. The <i>Directory</i> specified will be used as the home directory for storing files only. This default directory will be used only if a full pathname is not specified. The default directory for retrieving files is still /tftpboot .
-i	Logs the IP address of the calling machine with error messages.
-n	Allows the remote user to create files on your machine. Remote users are only allowed to read files with read permission for other if this flag is not specified.
-p	Specifies the port number for the incoming request.
-r <i>Option</i>	Specifies a tftp option negotiation to disable. Multiple -r flags can be used. For example, the following line in the <code>/etc/inetd.conf</code> file disables option negotiation for <code>tsize</code> and <code>blksize</code> : <pre>tftp dgram udp6 SRC nobody /usr/sbin/tftpd tftpd -n -r tsize -r blksize</pre>
-s	Turns on socket-level debugging.
-t	Specifies the timeout value for datagrams.
-v	Logs information messages when any file is successfully transferred by the tftpd daemon. This logging keeps track of who is remotely transferring files to and from the system with the tftpd daemon.
-x	Specifies the maximum of timeouts waiting for a datagram.
-z	Specifies the maximum allowed segment size for transfers.

Examples

Note: The arguments for the **tftpd** daemon can be specified by using SMIT or by editing the `/etc/inetd.conf` file.

1. To start the **tftpd** daemon, enter the following:

```
startsrc -t tftp
```

This command starts the **tftpd** subserver.

2. To stop the **tftpd** daemon normally, enter the following:

```
stopsrc -t tftp
```

This command allows all pending connections to start and existing connections to complete but prevents new connections from starting.

3. To force stop the **tftpd** daemon and all **tftpd** connections, enter the following:

```
stopsrc -f -t tftp
```

This command terminates all pending connections and existing connections immediately.

4. To display a short status report about the **tftpd** daemon, enter the following:

```
lssrc -t tftp
```

This command returns the daemon's name, process ID, and state (active or inactive).

tic Command

Purpose

Translates the terminfo description files from source to compiled format.

Syntax

```
tic [ -v [Number] ] [ -c ] FileName
```

Description

The **tic** command translates the terminfo files from the source format into the compiled format. The **tic** command places the results in the **/usr/share/lib/terminfo** directory. If the **TERMINFO** environment variable is set, the results are placed there instead of in the **/usr/share/lib/terminfo** directory.

The **tic** command compiles all terminfo descriptions in *FileName*. When the **tic** command finds a `use=entry-name` field, it searches the current file first. If unable to find the entry *-name*, it obtains the entry from the binary file in **/usr/share/lib/terminfo**. If **TERMINFO** is set, the terminfo directory is searched before **/usr/share/lib/terminfo**.

The total compiled entries cannot exceed 4096 bytes, and the name field cannot exceed 128 bytes.

Flags

Item	Description
-v [<i>Number</i>]	Writes trace information on the progress of the tic command. <i>Number</i> is an integer from 1 to 10 inclusive that increases the level of the verbosity. If <i>Number</i> is omitted, the default level is 1. The amount of information output increases as <i>Number</i> increases.
-c	Only checks <i>FileName</i> for errors. Errors in <code>use=entry-name</code> are not detected.

Files

Item	Description
/usr/share/lib/terminfo/?/*	Contains the compiled terminal capability database.

time Command

Purpose

Prints the time of the execution of a command.

Syntax

```
time [ -p ] Command [ Argument ... ]
```

Description

The **time** command prints the elapsed time during the execution of a command, time in the system, and execution time of the **time** command in seconds to standard error.

Note: Sleep time is not charged to either system or user time.

The **time** command is also built into the C shell (**cs****h**) and Korn shell (**ks****h**) with a different format. To run the **time** command while in the **cs****h** and **ks****h** shells, enter:

```
/usr/bin/time
```

Flags

Item	Description

-p	Writes the timing output to standard error. Seconds are expressed as a floating-point number with at least one digit following the radix character.

The standard format for this flag is as follows:

```
"real %f\nuser %f\nsys %f\n", <real seconds>, <user seconds>, <system seconds>
```

Exit Status

If you use the *Command* parameter, the exit status of the **time** command is the exit status of the specified command. Otherwise, the **time** command exits with one of the following values:

Item	Description

1-125	Indicates an error occurred in the time command.
126	Indicates the command specified by the <i>Command</i> parameter was found but could not be invoked.
127	Indicates the command specified by the <i>Command</i> parameter could not be found.

Examples

1. To measure the time required to run a program, enter:

```
/usr/bin/time -p a.out
```

This command runs the program **a.out** and writes the amount of real, user, and system time to standard error, in the format specified by the **-p** flag; for example:

```
real    10.5
user     0.3
sys      3.6
```

2. To save a record of the **time** command information in a file, enter:

```
/usr/bin/time a.out 2> a.time
```

Files

Item	Description
/usr/bin/time	Specifies the path of the time command.

timed Daemon

Purpose

Invokes the time server daemon.

Syntax

`/usr/sbin/timed [ -c ] [ -M ] [ -t ] [ [ -n Network ] ... | [ -i Network ] ... ]`

Note: Use the **rc.tcpip** file to start the daemon with each initial program load. You can specify the **timed** daemon at the command line. You can also use SRC commands to control the **timed** daemon from the command line.

Description

The **timed** daemon synchronizes one machine's clock with those of other machines on the local area network that are also running the **timed** daemon. The **timed** daemon slows the clocks of some machines and speeds up the clocks on other machines to create an average network time.

When the **timed** daemon is started without the **-M** flag, the machine locates the nearest master time server and asks for the network time. Then the machine uses the **date** command to set the machine's clock to the network time. The machine accepts synchronization messages periodically sent by the master time server and calls the **adjtime** subroutine to perform the needed corrections on the machine's clock.

When the **timed** daemon is started with the **-M** flag, the machine polls each of its local area networks to determine which networks have master time servers. The machine becomes a master time server on the networks that do not have a master time server. The machine becomes a submaster time server on the networks that already have a master time server. The **timed** daemon creates the **/var/adm/timed.masterlog** file when the **timed** daemon is started with the **-M** flag. The **/var/adm/timed.masterlog** file contains a log of the deltas between the local machine's clock and the clocks of the other machines on the networks for which the local machine is the master time server. The **/var/adm/timed.masterlog** file is updated approximately every 4 minutes and is never cleared. You may need to clear this file to conserve disk space. If the machine is only a submaster time server on its networks, the **/var/adm/timed.masterlog** file remains empty. To clear the **/var/adm/timed.masterlog** file, enter:

```
cat /dev/null > /var/adm/timed.masterlog
```

If the master time server ceases to function on a network, a new master time server is elected from the submaster time servers on that network. The **timedc** command enables you to select which submaster time server becomes the master time server.

The **timed** daemon can be controlled using the System Resource Controller (SRC), the System Management Interface Tool (SMIT), or the command line. The **timed** daemon is not started by default. Use the **rc.tcpip** file to start the **timed** daemon with each initial program load.

Manipulating the timed Daemon with the System Resource Controller

The **timed** daemon is a subsystem controlled by the **SRC**. The **timed** daemon is a member of the **SRC tcpip** system group. Use the following SRC commands to manipulate the **timed** daemon:

Item	Description
startsrc	Starts a subsystem, group of subsystems, or a subserver.
stopsrc	Stops a subsystem, group of subsystems, or a subserver.
lssrc	Gets the short status of a subsystem, group of subsystems, or a subserver. The long status option usually found in lssrc is not supported for the timed daemon.

Flags

Item	Description
-c	Specifies that the master-timed daemon should ignore the time values it gets from the other worker-timed daemons when for calculating the average network time. This flag changes the network time to be the same as the system clock on the master-timed daemon.
-i Network	Specifies a network to be excluded from clock synchronization. The <i>Network</i> variable can be either a network address or a network name. If a network name is specified for the <i>Network</i> variable, the network name must be defined in the /etc/networks file. Specify one network address or network name with each -i flag. Do not use this flag with the -n flag.
-M	Specifies the machine is a master or submaster time server on its local area networks. If a master time server is not currently available on a network, the machine becomes the master time server for that network. If a master time server already exists on a network, the machine becomes a submaster time server on that network. However, the machine can become the master time server if the current master time server becomes inoperative. The timed daemon creates the /var/adm/timed.masterlog file when the timed daemon is started with the -M flag.
-n Network	Specifies a network to include in clock synchronization. The <i>Network</i> variable can be either a network address or a network name. If a network name is specified for the <i>Network</i> variable, the network name must be defined in the /etc/networks file. Specify one network address or network name with each -n flag. Do not use this flag with the -i flag.
-t	Allows the timed daemon to trace the messages it receives and store them in the /var/adm/timed.log file. You can also use the timedc command to activate tracing.

Examples

1. To start the **timed** daemon with SRC control, enter:

```
startsrc -s timed
```

This command starts the daemon. You can use this command in the **rc.tcpip** file or on the command line. The **-s** flag specifies that the subsystem that follows is to be started.

2. To stop the **timed** daemon normally with SRC control, enter:

```
stopsrc -s timed
```

This command stops the daemon. The **-s** flag specifies that the subsystem that follows is to be stopped.

3. To get a short status report from the **timed** daemon, enter:

```
lssrc -s timed
```

This command returns the name of the daemon, the process ID of the daemon, and the state of the daemon (active or inactive).

4. To start the **timed** daemon with SRC control as the master or submaster time server and to exclude networks **net1** and **net2** from clock synchronization, enter:

```
startsrc -s timed -a "-M -i net1 -i net2"
```

This command starts the daemon. The machine becomes the master or submaster time server for its networks. Networks **net1** and **net2** are excluded from clock synchronization. The **-s** flag specifies that

the subsystem that follows is to be started. The **-a** flag specifies that the **timed** daemon should be started with the flags that follow. The flags must be enclosed in quotes.

5. To start the **timed** daemon, activate tracing, and include net1 and net2 in clock synchronization, enter:

```
timed -t -n net1 -n net2
```

This command starts the daemon. Tracing is activated and both net1 and net2 are included in clock synchronization.

Files

Item	Description
/var/adm/timed.log	Contains the messages traced for the timed daemon. This file is created when the timed daemon is started with the -t flag or when tracing is enabled with the timedc command.
/etc/rc.tcpip	Contains the SRC commands to be executed at system startup.
/var/adm/timed.masterlog	Contains a log of the deltas between the master time server clock and the clocks of the other machines on the networks. This file is created when the timed daemon is started with the -M flag. However, this file only contains information for those networks on which the machine is the master time server.

timedc Command

Purpose

Returns information about the **timed** daemon.

Syntax

timedc [*Subcommand* [*Parameter ...*]]

Description

The **timedc** command controls the operation of the **timed** daemon. The **timedc** command does the following:

- Measures the difference between clocks on various machines on a network.
- Finds the location of the master time server.
- Enables or disables tracing of messages received by the **timed** daemon.
- Debugs.

Without any variables, the **timedc** command assumes an interactive mode and prompts for subcommands from standard input. If variables are supplied, the **timedc** command interprets the first variable as a subcommand and the remaining variables as parameters to the subcommand. You can redirect standard input so the **timedc** command reads subcommands from a file.

Variables

The **timedc** command recognizes the following subcommands:

Item	Description
? [Parameter ...]	Displays a short description of each variable specified in the parameter list. The ? subcommand only works in interactive mode. If you give no variables, the ? subcommand shows a list of subcommands recognized by the timedc command.
clockdiff Host ...	Computes the differences between the clock of the host machine and the clocks of the machines given as variables.
election Host ...	Requests that the timed daemon on the specified host (s) reset its election timers and ensure that a timed master server is available. Up to 4 hosts can be specified. If a master timed server is no longer available, then the timed daemon on the specified host (s) will request to become the new timed master server. The specified host(s) must be running the timed daemon in submaster mode with the -M flag.
help [Parameter ...]	Displays a short description of each subcommand specified in the parameter list. If you give no variables, the help subcommand shows a list of subcommands recognized by the timedc command.
msite	Finds the location of the master site.
quit	Exits the timedc command.
trace { on off }	Enables or disables tracing of incoming messages to the timed daemon. The messages are held in the /var/adm/timed.log file.

You can use other commands for testing and debugging the **timed** daemon. Use the **help** command to find these commands.

These error messages may occur with the **timedc** command:

Item	Description
Ambiguous command	Abbreviation matches more than one command.
Invalid command	No match found.
Privileged command	Command can be executed only by the root user.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To display the time difference between the local host sahara and the remote host sandy, enter:

```
timedc clockdiff sandy
```

The output would be:

```
time on sandy.austin.century.com is 37904247 ms ahead of time on
sahara.austin.century.com
```

2. To display the client location of the **timed** daemon, enter:

```
timedc msite
```

The output would be:

```
client timed daemon runs on bupu.austin.century.com
```

timex Command

Purpose

Reports, in seconds, the elapsed time, user time, and system execution time for a command.

Syntax

timex [**-o**] [**-p**] [**-s**] *Command*

Description

The **timex** command reports, in seconds, the elapsed time, user time, and system execution time for a command. With specified flags, the **timex** command lists or summarizes process accounting data for a command and all of its children. *Command* is the name of any executable file on the system. It also reports total system activity during the execution interval. Output is written to standard error. The system uses the **/var/adm/pacct** file to select process records associated with the command and includes background processes with the same user ID, workstation ID, and execution time window.

Flags

Item	Description
-o	Reports the total number of blocks read or written and total characters transferred by a command and all its children.
-p	Lists process accounting records for a command and all its children. The number of blocks read or written and the number of characters transferred are reported. The -p flag takes the f , h , k , m , r , and t arguments defined in the acctcom command to modify other data items.
-f	Print the fork/ exec flag and system exit status columns in the output.
-h	Instead of mean memory size, shows the fraction of total available CPU time consumed by the process (hogfactor).
-k	Instead of memory size, shows total kcore minutes (memory measurement in kilobyte segments used per minute of run time).
-m	Shows mean main-memory size. This is the default. The -h flag or -k flag turn off the -m flag.
-r	Shows CPU factor.
-t	Shows separate system and user CPU times.
-s	Reports total system activity during the execution of the command. All the data items listed in the sar command are reported.

Note: Accounting must be turned on to use the **-o** or **-p** flags.

Examples

1. To report the total number of blocks read and total characters transferred by the **ls** command, enter:

```
timex -o ls
```

2. To list the process accounting records for the **ps** command, enter:

```
timex -p ps -fe
```

3. To report total system activity for the execution of the **ls** command, enter:

```
timex -s ls
```

Files

Item	Description
/var/adm/pacct	Used to select record associated with the command.

tip Command

Purpose

Connects to a remote system.

Syntax

tip [**-v**] [**- BaudRate**] { *SystemName* | *PhoneNumber* }

Description

The **tip** command connects to a remote system and allows you to work on the remote system as if logged in directly.

Either the *SystemName* parameter or the *PhoneNumber* parameter is required. The *SystemName* parameter specifies the name of a remote system to be contacted. The remote system must be defined in the **/etc/remote** file, or in the file specified by the **REMOTE** environment variable. The *PhoneNumber* parameter specifies the number to dial over a modem connection.

When the **tip** command is invoked with the *SystemName* parameter, it searches the **remote** file for an entry beginning with that system name. When the command is invoked with the *PhoneNumber* parameter, it searches the **remote** file for an entry of the form **tipBaudRate**, where *BaudRate* is the baud rate for the connection. If the **-BaudRate** flag is not used, the **tip** command looks for a **tip1200** entry, because 1200 is the default baud rate.

The actions of the **tip** command can be controlled using flags, escape signals and variables. The **tip** command reads the **/etc/remote** file to find out how to contact a remote system and discover the escape-send sequence to use when communicating with that system. In addition, the command may check the **/etc/phones** file to find out a phone number for the remote system.

A **tip** user can create an individual remote file in the format of the **/usr/lib/remote-file** file, and then specify the file to use with the **REMOTE** environment variable. A user can also create an individual phones file in the format of the **/usr/lib/phones-file** file, and then specify the file to use with the **PHONES** environment variable. The **tip** command does not read the **/usr/lib/remote-file** file or **/usr/lib/phones-file** file by default, however. The default files that the **tip** command uses are the **/etc/remote** file and **/etc/phones** file.

A **tip** user can create a **\$HOME/.tiprc** file to specify initial settings for the **tip** variables. In addition, settings made in the remote file, the phones file, and the **.tiprc** file can be overridden by using escape

signals while **tip** is running. Escape signals can also be used, for instance, to start and stop file transfers or interrupt a connection to remote system.

The **tip** command uses lock files in the **/etc/locks** directory to lock devices against multiple access and to prevent multiple users from logging in on the same system.

When the **tip** command prompts for a response, edit the line as you type using the standard keys. Entering ~. (tilde, period) in response to a prompt, or pressing the Interrupt key, will abort the **tip** dialog and return you to the remote system.

You can use the **tip** command to transfer files to and from the remote system. You can use **tip** command escape signals to start and stop the file transfers. Several **tip** command variables work together to control file transfers.

File transfers usually use tandem mode to control the flow of data. If the remote system does not support tandem mode, set the *echocheck* variable to on to cause the **tip** command to synchronize with the remote system after transmitting each character. When transferring files with the ~< and ~> escape signals, use the **eofread** and *eofwrite* variables to specify the end of a file when writing, and recognize the end of a file when reading.

If the *verbose* variable is set on, the **tip** command performs the following:

- Writes a running count of the number of lines transferred during a file transfer.
- Writes messages indicating its actions as it dials a phone number.

You can use scripting to record the conversations you have with the **tip** command. Use the *script* variable to start scripting.

Note:

1. Only a user with root user authority can change the *dialtimeout* variable.
2. Although any user can specify a host at the command line, only the root user can change the *host* variable setting after the **tip** command has been started. However, this does not change the system to which the **tip** command is currently connected.

Flags

Item	Description
-v	Displays the settings of variables as they are read from the .tiprc file.
-BaudRate	Overrides the default baud rate, which is 1200 baud.

Escape Signals

Using escape signals, you can instruct the **tip** command to terminate, log off from the remote system, and transfer files. The escape character at the beginning of a line indicates an escape signal. The default escape character is a ~ (tilde). The character can be changed using the *escape* variable. All other typed characters are transmitted directly to the remote system. The **tip** command recognizes the following escape signals:

Item	Description
~^D~	Terminates the connection and exits. You may still be logged in on the remote system; if so, you can issue another tip command to reconnect to that remote system.
~c [Directory]	Changes, on the local system, to the directory specified by the <i>Directory</i> variable. If you do not include the <i>Directory</i> variable, the tip command changes to your home directory.
~!	Escapes to a shell on the local system. When you exit from the shell, you return to the tip command.

Item	Description
~>	Copies a file from the local system to the remote system. The tip command prompts you for the name of the local file.
~<	Copies a file from the remote system to the local system. The tip command prompts you for the name of the remote file.

A **tip** file download will only download the file until one of the EOF characters listed in the **eofread** command variable is encountered. If one of these characters is not encountered, then the file copy will not succeed.

When downloading a file with the ~< signal, the user will be prompted for a local file name. The user may respond with any valid writeable file name. When prompted for the remote command, the user should append the EOF character to the end of the file being read.

This signal can be used as shown in the following example:

List command for remote system? echo "\04" | cat /etc/passwd

This example assumes that the character 0x4 is present in the **tip eofread** variable. The best way of ensuring that this character exists in the variable is to assign it in the user's **.tiprc** file, which should reside in the user's home directory.

To accomplish this, the following command can be issued:

```
echo"eofread=\04" >> ~/.tiprc
```

Item	Description
~p <i>Source</i> [<i>Dest</i>]	Sends (puts) the <i>Source</i> file to a remote UNIX host system, using the cat command to copy the <i>Source</i> file to the <i>Dest</i> file. If the <i>Dest</i> file name is not specified, the cat command uses the name of the <i>Source</i> file. If the <i>Dest</i> file exists on the remote host, it will be replaced by the <i>Source</i> file. This signal is a UNIX-specific version of the ~> signal.
~t <i>Source</i> [<i>Dest</i>]	Transfers (takes) the <i>Source</i> file from a remote UNIX host system to the local system, using the cat command to copy the <i>Source</i> file to the <i>Dest</i> file on the local system. If the <i>Dest</i> file name is not specified, the cat command uses the name of the <i>Source</i> file. If the <i>Dest</i> file exists on the local system, it will be replaced by the <i>Source</i> file. This signal is a UNIX-specific version of the ~< signal.
~	Pipes the output of a remote command to a local process. The command string sent to the local system is processed by the shell.

A remote pipe will only succeed if the data from the remote pipe is terminated by one of the eof characters listed in the **eofread tip** command variable. If one of these characters is not encountered, then the output pipe will not succeed.

When piping remote output with the ~| signal, the user will be prompted for a local command name. The user may respond with any valid command name. When prompted for the remote command, the user should append the EOF character to the end of the file being read.

This signal can be used as shown in the following example:

```
Local command? cat
List command for remote system? echo
"asdfasdfasdfasdf\04"
```

This example assumes that the character 0x4 is present in the **tip eofread** variable. The best way of ensuring that this character exists in the variable is to assign it in the user's **.tiprc** file, which should reside in the user's home directory.

To accomplish this, the following command can be issued:


```
echo"eofread=\04" >> ~/.tiprc
```

Item	Description
~\$	Pipes the output of a local process to the remote system. The command string sent to the remote system is processed by the shell.
~#	Sends a BREAK signal to the remote system.
~s { Variable=Value [!]BoolVariable all Variable? }	Sets or queries the tip command <u>variables</u>. To change the value of a non-Boolean variable, enter the variable name or abbreviation, followed by an = (equal sign), followed by the new value. For example, type ~s rc=^U to change the character used to turn uppercase conversion on or off (the <u>raisechar</u> variable). To change the value of a Boolean variable, enter the variable name or abbreviation. To reset the variable to its default value, type an ! (exclamation point) in front of the name. For example, type ~s !ec to reset the <u>echocheck</u> variable to its default value. To display all variables readable by the user, specify all as an argument to the ~s signal. You may also request the display of a specific variable by attaching a ? (question mark) to the variable name. For example, type the command ~s eol? to display the current end-of-line string (the <u>eol</u> variable).
~^Z	Stops the tip command. The ~^Z signal is only available with job control.
~^Y	Stops the local portion of the tip command. The remote portion, which displays the output from the remote system, continues to run. The ~^Y signal is only available with job control.
~?	Displays a list of the escape signals.

Variables

The **tip** command uses variables that control its operation. These variables may be numeric, string, character, or Boolean values. Some of these variables can be changed by any user who can run the **tip** command. However, the following variables can be changed only by a user with root user authority: the *baudrate* variable and the *dialtimeout* variable.

Variables may be initialized at run time in the **\$HOME/.tiprc** file. Additionally, you can display and set the variables while already running the **tip** command by using the ~s escape signal.

Variables may be numeric, string, character, or Boolean values. To set a non-Boolean variable, enter the variable name or abbreviation followed by an = (equal sign) and the value. For example, type either ~s host=zeus or ~s ho=zeus to change the **host** name to zeus. In the **.tiprc** file, type host=zeus or ho=zeus.

To change the value of a Boolean variable, enter the variable name or abbreviation as an argument to the ~s signal or on a line of the **.tiprc** file. To reset the variable to its default value, type an ! (exclamation point) in front of the name. For example, type ~s !echocheck to reset the *echocheck* variable to its default value while running the **tip** command.

Following are the common variables, their types, abbreviations, and default values.

Variable (Abbreviation)	Type	Description
<i>beautify (be)</i>	Boolean	Instructs the tip command to discard unprintable characters when a session is being scripted. Does not discard characters specified with the <i>exceptions</i> variable. The default setting is on.
<i>baudrate (ba)</i>	Numeric	Reflects the baud rate of the connection. Changing the value of this variable will <i>not</i> change the current baud setting of the connected tty device.
<i>dialtimeout (dial)</i>	Numeric	Specifies the time in seconds that the tip command waits for a connection when dialing a phone number. The default is 60 seconds. The dialtimout setting can be changed only by someone with root user authority.
<i>echocheck (ec)</i>	Boolean	Instructs the tip command to synchronize with the remote system during a file transfer by awaiting the echo of the last character transmitted before transmitting the next character. The default setting is off.
<i>eofread (eofr)</i>	String	Specifies the set of characters that signifies end-of-transmission during a remote-to-local (~< or ~t) file transfer.
<i>eofwrite (eofw)</i>	String	Specifies the string that is sent to indicate the end of a transmission during a local-to-remote (~> or ~p) file transfer.
<i>eol (none)</i>	String	Specifies the string that indicates the end of a line. The tip command recognizes escape signals only when they follow an end-of-line string.
<i>escape (es)</i>	Character	Specifies the character prefix for escape signals. The default is ~ (tilde).
<i>etimeout (et)</i>	Numeric	Specifies the time to wait for a response when the <i>echocheck</i> variable is set on. If the echo is not received within the designated time, the file transfer is discontinued. The default time is 28 seconds.
<i>exceptions (ex)</i>	String	Specifies the set of characters that should not be discarded even when the beautify switch is set to on. The \t\n\f\b string is the default.
<i>force (fo)</i>	Character	Specifies the character that is used to force literal data transmissions during binary transfers. The ^P character is the default. Literal data transmissions are off until the user types the character specified by the <i>force</i> variable.
<i>framesize (fr)</i>	Numeric	Specifies the number of bytes to buffer between files system writes when receiving files from the remote system.
<i>host (ho)</i>	String	Specifies the name of the remote system to which you were connected when the tip command was invoked. This variable cannot be changed.
<i>halfduplex (hdx)</i>	Boolean	Toggles Half-duplex mode. The default setting is off.
<i>localecho (le)</i>	Boolean	Toggles the Local-echo mode. The default setting is off.

Variable (Abbreviation)	Type	Description
<i>log</i> (none)	String	Defines the file used to log dial-outs with the tip command. The default file is the /var/spool/uucp/.Admin/aculog file. The log file can be changed only by someone with root authority.

Variable (Abbreviation)	Type	Description
<i>parity</i> (<i>par</i>)	String	Defines the parity for file transfers. Defaults to the following string: no parity, 8 data bits
<i>phones</i> (none)	String	Specifies the name of the user's phone file. The file can have any valid file name and must be set up in the format of the /usr/lib/phones-file file. The default is the /etc/phones file. If a file is specified with the PHONES environment variable, it is used in place of (not in addition to) the /etc/phones file.
<i>prompt</i> (<i>pr</i>)	Character	Specifies the character that indicates the end of the line on the remote host. This character is used to synchronize during data transfers. The tip command counts lines transferred during a file transfer, based on the number of times it receives the prompt character. The \n character is the default.
<i>raise</i> (<i>ra</i>)	Boolean	When set to on, instructs the tip command to convert all lowercase letters to uppercase before transmitting them to the remote system. The default setting is off.
<i>raisechar</i> (<i>rc</i>)	Character	Specifies a character that is used to toggle uppercase conversion. The ^A character is the default.
<i>rawftp</i> (<i>raw</i>)	Boolean	If the rawftp variable is set to on, data is transmitted over the connection during a file transfer with no additional processing carried out. That is, when sending files, line-feeds are not mapped to line-feed/carriage carried out.
<i>record</i> (<i>rec</i>)	String	Specifies the name of the file in which the tip command records the session script. The tip.record file is the default. The tip command places the file in the user's current directory on the local system.
<i>remote</i> (none)	String	Specifies the name of the user's remote system definition file. The file can have any valid file name and must be set up in the format of the /usr/lib/remote-file file. The default is the /etc/remote file. If a file is specified with the REMOTE environment variable, it is used in place of (not in addition to) the /etc/remote file.
<i>script</i> (<i>sc</i>)	Boolean	When the script switch is set on, the tip command records everything transmitted by the remote system in a file on the local system. The file name is specified by the <i>record</i> variable. If the beautify switch is set to on, only printable ASCII characters (those between 040 and 0177) will be recorded in the script file. The <i>exceptions</i> variable specifies unprintable characters that will be recorded even if the beautify switch is set to on. The default setting for the script switch is off.

Variable (Abbreviation)	Type	Description
<i>tabexpand (tab)</i>	Boolean	Causes the tip command to expand tab characters to eight spaces during file transfers. The default setting is off.
<i>verbose (verb)</i>	Boolean	When the verbose switch is set on, the tip command prints messages while dialing, shows the current number of lines transferred during a file transfer, and displays other status information about the connection. The default setting is on.
<i>SHELL (none)</i>	String	Specifies the type of shell to use for the ~! signal. The default value is /usr/bin/sh or is taken from the environment.
<i>HOME (none)</i>	String	Specifies the home directory to use for the ~c signal. The default value is taken from the environment.

Examples

1. To specify a baud rate when making a direct connection, type:

```
tip -300 hera
```

This instructs the **tip** command to use **baud rate** of 300 when contacting remote system hera.

2. To use a modem to connect to a remote system, type:

```
tip 9,343-2132
```

The **tip** command connects the local system to the remote system reached by the telephone number 343-2132, after dialing a 9 to reach an outside line.

3. To connect directly to a remote system and display the variables, type:

```
tip -v hera
```

The **-v** flag causes the **tip** command to display the values of the variables as it reads them from the **\$HOME/.tiprc** file. If the **.tiprc** file contains the following settings:

```
sc
be
rec=/home/jimk/callout
```

then output from the **-v** flag is as follows:

```
set script
set beautify
set record=/home/jimk/callout
```

Files

Item	Description
/usr/bin/tip	Contains the tip command.
/etc/locks/*	Contains lock files that prevent multiple uses of devices and multiple calls to systems.
/etc/remote	Contains system descriptions for the tip command. If the <i>remote</i> variable or the REMOTE environment variable is set, that file is used instead.

Item	Description
/usr/lib/remote-file	Contains sample remote file. If the <i>remote</i> variable or the RECORD environment variable is set, that file is used instead.
/etc/phones	Contains the telephone number database for the tip command. If the <i>phones</i> variable or the PHONES environment variable is set, that file is used instead.
/usr/lib/phones-file	Contains the telephone number database for the tip command. If the <i>phones</i> variable or the PHONES environment variable is set, that file is used instead.
\$HOME/.tiprc	Defines initial settings for the tip command variables.
tip.record	Contains the tip command scripts. By default, the file is stored in the current directory. The user can change the file name and directory using the <i>record</i> variable.

tnconconsole Command

Purpose

Reports and manages the trusted network connect (TNC) server, the TNC client, the TNC IP Referrer (IPRef), and Service Update Management Assistant (SUMA). It manages fileset and patch management policies regarding endpoint (server and client) integrity at or after network connection to protect the network from threats and attacks.

Note: This command is used to demonstrate **TNC** options and has limited functionality. To use the full function of this command, install PowerSC Standard Edition. In PowerSC Standard Edition, the name of the **tnconconsole** command was changed to the **psconf** command.

Syntax

TNC server operations:

```
tnconconsole mkserver [ tncport=<port> ] pmserver=<host:port> [tsserver=<host>]
[ recheck_interval=<time_in_minutes> ] d (days) : h (hours) : m (minutes) ] [dbpath = <user-defined
directory> ]

tnconconsole { rmserver | status }

tnconconsole { start | stop | restart } server

tnconconsole chserver attribute = value

tnconconsole add -F <FSPolicyname> -r <buildinfo> [apargrp= [±]<apargrp1, apargrp2.. >]
[ifixgrp=[+|-]<ifixgrp1, ifixgrp2...>]

tnconconsole add { -G <ipgroupname> ip=[±]<host1, host2...> | {-A<apargrp> [aparlist=[±]apar1, apar2... |
{-V <ifixgrp> [ifixlist=[+|-]ifix1, ifix2...]}

tnconconsole add -P <policyname> { fspolicy=[±]<f1, f2...> | ipgroup=[±]<g1, g2...> }

tnconconsole add -e emailid [-E FAIL | COMPLIANT | ALL ] [ipgroup= [± ]<g1, g2...>]

tnconconsole add -I ip= [±]<host1, host2...>

tnconconsole delete { -F <FSPolicyname> | -G <ipgroupname> | -P <policyname> | -A <apargrp> | -V
<ifixgrp>}

tnconconsole delete -H -i <host | ALL> -D <yyyy-mm-dd>

tnconconsole certadd -i <host> -t <TRUSTED | UNTRUSTED>

tnconconsole certdel -i <host>
```

tnconconsole verify **-i** <host> | **-G** <ipgroup>

tnconconsole update **[-p]** { **-i** < host > | **-G** <ipgroup> **[-r** <buildinfo> | **-a** <apar1, apar2...> | **[-u]** **-v** <ifix1, ifix2,...> }

tnconconsole log loglevel=<info | error | none>

tnconconsole import **-C -i** <host> **-f** <filename> | **-d** <import database filename>

tnconconsole { **import** **-k** <key_filename> | **export** } **-S -f** <filename>

tnconconsole list { **-S** | **-G** < ipgroupname | **ALL** > | **-F** < FSPolicyname | **ALL** > | **-P** < policynome | **ALL** > | **-r** < buildinfo | **ALL** > | **-I -i** < ip | **ALL** > | **-A** < apargrp | **ALL** > | **-V** < ifixgrp> } **[-c]** **[-q]**

tnconconsole list { **-H** | **-s** < **COMPLIANT** | **IGNORE** | **FAILED** | **ALL** > } **-i** <host | **ALL** > **[-c]** **[-q]**

tnconconsole export **-d** <path to export directory>

tnconconsole report **-v** <CVEid|ALL> **-o** <TEXT|CSV>

tnconconsole report **-A** <advisoryname>

tnconconsole report **-P** <policynome|ALL> **-o** <TEXT|CSV>

tnconconsole report **-i** <ip|ALL> **-o** <TEXT|CSV>

tnconconsole report **-B** <buildinfo|ALL> **-o** <TEXT|CSV>

TNC client operations:

tnconconsole mkclient [**tnoport**=<port>] **tnserver**=<host:port>

tnconconsole mkclient **tnoport**=<port> **-T**

tnconconsole { **rmclient** | **status** }

tnconconsole { **start** | **stop** | **restart** } **client**

tnconconsole chclient attribute = value

tnconconsole list { **-C** | **-S** }

tnconconsole export { **-C** | **-S** } **-f** <filename>

tnconconsole import { **-S** | **-C -k** <key_filename> } **-f** <filename>

TNC IPRef operations:

tnconconsole mkipref [**tnoport**=<port>] **tnserver**=<host:port>

tnconconsole { **rmipref** | **status** }

tnconconsole { **start** | **stop** | **restart** } **ipref**

tnconconsole chipref attribute = value

tnconconsole { **import** **-k** <key_filename> | **export** } **-R -f** <filename>

tnconconsole list **-R**

Description

The TNC technology is an open standard-based architecture for endpoint authentication, platform integrity measurement, and integrating security systems. The TNC architecture inspects endpoints (network clients and servers) for compliance with security policies before allowing them on the protected network. The TNC IPRef notifies the TNC server about any new IPs that are detected on the virtual I/O server (VIOs).

SUMA helps move system administrators away from the task of manually retrieving maintenance updates from the web. It offers flexible options that enable the system administrator to set up an automated interface to download fixes from a fix distribution website to their systems.

The **tnconconsole** command manages the network server and clients by adding or deleting security policies, validating clients as trusted or untrusted, generating reports, and updating the server and the client.

The following operations can be performed by using the **tnconconsole** command:

Item	Description
add	Adds a policy, a client, or the email information on the TNC server.
apargrp	Specifies the APAR group names as part of the fileset policy that are used for verification of TNC clients.
aparlist	Specifies the list of APARs that are part of the APAR group.
certadd	Marks the certificate as trusted or untrusted.
certdel	Deletes the client information.
chclient	Changes the attributes in the <code>tnccs.conf</code> file. An explicit start command is required for the changes to take effect in the TNC client. The syntax of <code>attribute=value</code> will be same as that of mkclient .
chipref	Changes the attributes in the <code>tnccs.conf</code> file. An explicit start command is required for the changes to take effect in IPRef. The syntax of <code>attribute=value</code> is the same as that of the mkipref .
chserver	Changes the attributes in the <code>tnccs.conf</code> file. An explicit start command is required for the changes to take effect in the TNC server. The syntax of <code>attribute=value</code> is same as that of mkserver . Note: The dbpath attribute cannot be changed by using the chserver command. It can be set only while running the mkserver .
dbpath	Specifies the TNC database location. The default value is <code>/var/tnc</code> .
delete	Deletes a policy or the client information.
export	Exports the client or server certificate , or database on TNC server.
fspolicy	Specifies the fileset policy of the release, technology level and service pack that are used for verification of TNC Clients.
import	Imports a certificate on client or server, or database on TNC server.
ipgroup	Specifies the Internet Protocol (IP) group that contains multiple client IP addresses or host names.
list	Displays information about the TNC server, the TNC client, or the SUMA.
log	Sets the log level for the TNC components.
mkclient	Configures the TNC client.
mkipref	Configures the TNC IPRef.
mkserver	Configures the TNC server.
pmport	Specifies the port number on which the pmserver listens to. The default value is 38240.
pmserver	Specifies the host name or IP address of the suma command that downloads the latest service packs and security fixes available in the IBM® ECC website and the IBM Fix Central website.

Item	Description
recheck_interval	Specifies the interval in minutes or d (days) : h (hours) : m (minutes) format for the TNC server to verify the TNC clients. Note: A value of recheck_interval=0 means that the scheduler does not initiate verification of the clients at regular intervals and the registered clients are automatically verified during the startup. In such cases, the client can be manually verified.
report	Generates a report that has .txt or .csv file extension.
restart	Restarts the TNC client, the TNC server, or the TNC IPRef.
rmclient	Unconfigures the TNC client.
rmipref	Unconfigures the TNC IPRef.
rmserver	Unconfigures the TNC server.
start	Starts the TNC client, the TNC server, or the TNC IPRef.
status	Shows the status of the TNC configuration.
stop	Stops the TNC client, the TNC server, or the TNC IPRef.
tncport	Specifies the port number on which the TNC server listens to. The default value is 42830.
tncserver	Specifies the TNC server that verifies or updates the TNC clients.
tssserver	Specifies the IP or host name of the TS server.
update	Installs patches on the client.
verify	Initiates a manual verification of the client.

Flags

Item	Description
-A <advisoryName>	Specifies the advisory name for the report.
-B <buildinfo>	Specifies the build information to prepare a patch report.
-i host	Specifies the IP address or host name.
-f filename	Specifies the file from which the certificate must be read in case of an import operation, or specifies the location to which the certificate must be written in case of an export operation.
-F fspolicy buildinfo	Specifies the file system policy name, followed by the build information. The build information can be provided in the following format: 6100-04-01, where 6100 represents version 6.1, 04 is the maintenance level, and 01 is the service pack.
-G ipgroupname ip=[±]ip1, ip2...	Specifies the IP group name followed by a comma-separated IP list.
-P policyname fspolicy=[±]fspolicy1, fspolicy2... ipgroup=[±]g1, g2...	Specifies the policy name followed by a comma-separated file system policy name list and an IP group name list. File system policies and IP groups can be added or removed from the file system policy name list and IP group name list by using + or - symbols, respectively.

Item	Description
-I ip =[±] <i>ip1, ip2...</i> [±] <i>host1,host2...</i>	Specifies the IP/host name that must be ignored during verification.
-e <i>emailid</i> ipgroup =[±] <i>g1, g2...</i>	Specifies the email ID followed by a comma separated IP group name list.
-E FAIL COMPLIANT ALL	Specifies the event for which the emails need to be sent to the configured email id. FAIL- Mails are sent when the verification status of the client is FAILED. COMPLIANT- Mails are sent when the verification status of the client is COMPLAINT. ALL - Mails are sent for all the statuses of the client verification.
-d <i>database file location/dir path of database</i>	Specifies the file path location for import of the database/specifies the directory path location for export of the database.
-t TRUSTED UNTRUSTED	Marks the specified client as trusted or untrusted. Note: Only system administrators can verify the server or client as trusted or untrusted.
-c	Displays the user attributes in colon-separated records as follows: <pre># name: attribute1: attribute2: ...</pre> <pre>policy: value1: value2: ...</pre>
-p	Previews the TNC client update.
-q	Suppresses the header information.
-s COMPLIANT IGNORE FAILED ALL	Displays the client by status as follows: COMPLIANT Displays the active clients. IGNORE Displays the clients that are excluded from any verification. FAILED Displays the clients that have failed verification as per the configured policy. ALL Displays all the clients irrespective of their statuses.
-u	Uninstalls an interim fix that is installed on a TNC client.
-r <i>buildinfo</i>	Generates the report based on the build information. The build information can be provided in the following format: 6100-04-01, where 6100 represents version 6.1, 04 is the maintenance level, and 01 is the service pack.
-H	Lists the history log.
-C	Specifies that the operation is for client component.
-S	Specifies that the operation is for server component.
-T	Specifies that the client can accept request from any TS server that has a valid certificate.
-v	Specifies a comma-separated interim fix list.

Item	Description
-V	Specifies the interim fix group name.
-R	Specifies that the operation is for IPRef component.
-k <i>filename</i>	Specifies the file from which the certificate key must be read in case of an import operation.
-D <i>yyyy-mm-dd</i>	Specifies the date for a particular client entry in the log history, where <i>yyyy</i> is the year, <i>mm</i> in the month, and <i>dd</i> is the day.
-P <policyName>	Specifies the policy name to prepare a client policy report.
-S <host>	Specifies the host name to prepare a client security fix report.

Exit Status

This command returns the following exit values:

Item	Description
0	The command ran successfully, and all the requested changes are made.
>0	An error occurred. The printed error message includes more details about the type of failure.

Examples

1. To start the TNC server, enter the following command:

```
tnccconsole start server
```

2. To add a file system policy named 71D_latest for the build 7100-04-02, enter the following command:

```
tnccconsole add -F 71D_latest 7100-04-02
```

3. To delete a file system policy named 71D_old, enter the following command:

```
tnccconsole delete -F 71D_old
```

4. To validate that the client that has an IP address of 11.11.11.11 is **trusted**, enter the following command:

```
tnccconsole certadd -i 11.11.11.11 -t TRUSTED
```

5. To delete the client that has an IP address of 11.11.11.11 from the server, enter the following command:

```
tnccconsole certdel -i 11.11.11.11
```

6. To verify the client information that has an IP address of 11.11.11.11, enter the following command:

```
tnccconsole verify -i 11.11.11.11
```

7. To display the client information that has an IP address of 11.11.11.11, enter the following command:

```
tnccconsole list -i 11.11.11.11
```

8. To generate the report for clients that are in **COMPLAINT** status, enter the following command:

```
tnccconsole list -s COMPLAINT -i ALL
```

9. To generate the report for the build 7100-04-02, enter the following command:

```
tnconsole list -r 7100-04-02
```

10. To display the connection history of a client that has an IP address of 11.11.11.11, enter the following command:

```
tnconsole list -H -i 11.11.11.11
```

11. To delete the entry of a client that has an IP address of 11.11.11.11 from the log history older or equal to 1 February, 2009, enter the following command:

```
tnconsole delete -H -i 11.11.11.11 -D 2009-02-01
```

12. To import the client certificate of a client that has an IP address of 11.11.11.11 from the server, enter the following command:

```
tnconsole import -C -i 11.11.11.11 -f /tmp/client.txt
```

13. To export the server certificate from a client, enter the following command:

```
tnconsole export -S -f /tmp/server.txt
```

14. To update the client that has an IP address of 11.11.11.11 to an appropriate level from the server, enter the following command:

```
tnconsole update -i 11.11.11.11
```

15. To display the client statuses, enter the following command:

```
tnconsole status
```

16. To display the client certificate, enter the following command:

```
tnconsole list -C
```

17. To start the client, enter the following command:

```
tnconsole start client
```

Security

Attention RBAC users:

This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in Security. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand

tninit Command

Purpose

Initializes the Trusted Network subsystem and maintains the Trusted Network rules database.

Syntax

```
tninit [ -v m ] init [ filename ]
```

```
tninit [ -v m ] loadfilename
```

```
tninit [ -v m ] savefilename
```

tninit [**-v m**] **disp***filename*

Description

The **tninit** command initializes the Trusted Network subsystem and maintains the Trusted Network rules database, including the **/etc/security/rules.host** and the **/etc/security/rules.int** files that are loaded upon system startup.

Flags

Item	Description
-v	Specifies verbose mode.
-m	Maintains the existing host rules when loading a new database.
init [<i>filename</i>]	Initializes the Trusted Network subsystem. This parameter loads tables into the kernel that are responsible for making the translation between a local representation of an Sensitivity Label (SL) and what is transmitted over the network. Optionally, you can specify the name of a file containing the mappings with the <i>filename</i> parameter. If you do not specify a file, a set of hard coded mappings is used. You can see an example of the mapping in the /usr/samples/tn/rfc1108.example file.
load <i>filename</i>	Loads a rules database into the kernel. Use the <i>filename</i> parameter to specify the file name. The command appends the .host and .int extensions to get the two files that comprise the database.
save <i>filename</i>	Saves the rules that are active in the kernel into the two files of the database. Uses the <i>filename</i> parameter to specify the file name. The .host and .int extensions are appended to the file name to get the two files that comprise the database.
disp <i>filename</i>	Displays the database that is specified for standard output (STDOUT). Use the <i>filename</i> parameter to specify the file name. The command appends the .host and .int extensions to get the two files that comprise the database.

Parameters

Item	Description
<i>filename</i>	Specifies the file name. Do not use init , load , save , or disp as file name.

Authorization

A user must have the **aix.mls.network.init** authorization to run the **tninit** command.

Examples

To initialize the Trusted Network subsystem, enter the following command:

```
tninit init
```

To load a rules database into the kernel, enter the following command:

```
tninit load /etc/security/rules
```

To save the rules active in the kernel into the two files of the database, enter the following command:

```
tninit save /etc/security/rules
```

To display the rules database specified into STDOUT, enter the following command:

```
tninit disp /etc/security/rules
```

tokstat Command

Purpose

Shows token-ring device driver and device statistics.

Syntax

tokstat [**-d -r -t**] *Device_Name*

Description

The **tokstat** command displays the statistics gathered by the specified Token-Ring device driver. The user can optionally specify that the device-specific statistics be displayed in addition to the device driver statistics. If no flags are specified, only the device driver statistics are displayed.

This command is also invoked when the **netstat** command is run with the **-v** flag. The **netstat** command does not issue any **tokstat** command flags.

If an invalid *Device_Name* is specified, the **tokstat** command produces an error message stating that it could not connect to the device.

Flags

Item	Description

-d	Displays all the device driver statistics, including the device-specific statistics.
-r	Resets all the statistics back to their initial values. This flag can only be issued by privileged users.
-t	Toggles debug trace in some device drivers.

Parameters

Item	Description

<i>Device_Name</i>	The name of the Token-Ring device, for example, tok0 .

Statistic Fields

Note: Some adapters may not support a specific statistic. The value of non-supported statistic fields is always 0.

The statistic fields displayed in the output of the **toktstat** command and their descriptions are:

Title Fields

Item	Description

Device Type	Displays the description of the adapter type.

Hardware Address	Displays the Token-Ring network address currently used by the device.

Item	Description
Elapsed Time	Displays the real time period which has elapsed since the last time the statistics were reset. Part of the statistics may be reset by the device driver during error recovery when a hardware error is detected. There will be another Elapsed Time displayed in the middle of the output when this situation has occurred in order to reflect the time differences between the statistics.

Transmit Statistics Fields

Item	Description
Packets	The number of packets transmitted successfully by the device.
Bytes	The number of bytes transmitted successfully by the device.
Interrupts	The number of transmit interrupts received by the driver from the adapter.
Transmit Errors	The number of output errors encountered on this device. This is a counter for unsuccessful transmissions due to hardware/network errors.
Packets Dropped	The number of packets accepted by the device driver for transmission which were not (for any reason) given to the device.
Max Packets on S/W Transmit Queue	The maximum number of outgoing packets ever queued to the software transmit queue.
S/W Transmit Queue Overflow	The number of outgoing packets which have overflowed the software transmit queue.
Current S/W+H/W Transmit Queue Length	The number of pending outgoing packets on either the software transmit queue or the hardware transmit queue.
Broadcast Packets	The number of broadcast packets transmitted without any error.
Multicast Packets	The number of multicast packets transmitted without any error.
Timeout Errors	The number of unsuccessful transmissions due to adapter reported timeout errors.
Current SW Transmit Queue Length	The number of outgoing packets currently on the software transmit queue.
Current HW Transmit Queue Length	The number of outgoing packets currently on the hardware transmit queue.

Receive Statistics Fields

Item	Description
Packets	The number of packets received successfully by the device.
Bytes	The number of bytes received successfully by the device.
Interrupts	The number of receive interrupts received by the driver from the adapter.
Receive Errors	The number of input errors encountered on this device. This is a counter for unsuccessful reception due to hardware/network errors.
Packets Dropped	The number of packets received by the device driver from this device which were not (for any reason) given to a network demuxer.
Bad Packets	The number of bad packets received (saved) by the device driver.
Broadcast Packets	The number of broadcast packets received without error.
Multicast Packets	The number of multicast packets received without error.
Receive Congestion Errors	The number of incoming packets dropped by the hardware due to a no resource error.

General Statistics Fields

Item	Description
No mbuf Errors	The number of times mbufs were not available to the device driver. This usually occurs during receive operations when the driver must obtain mbuf buffers to process inbound packets. If the mbuf pool for the requested size is empty, the packet will be discarded. The netstat -m command can be used to confirm this.
Lobe Wire Faults	The number of times the adapter detected an open or short circuit in the lobe data path (for example, the cable is unplugged).
Abort Errors	The number of times the adapter had problems transmitting.
AC Errors	The number of times the adapter received more than one AMP (Active Monitor Present) or SMP (Standby Monitor Present) frame which had the address recognized and frame copied bits set to zero. This indicates a problem with neighbor notification. Every station learns and remembers who its Nearest Active Upstream Neighbor (NAUN) is from AMP and SMP frames. When a station reports a problem, it also reports who its NAUN is. This helps to define the <i>fault domain</i> .
Burst Errors	The number of times the adapter detected that the polarity of the signal did not switch when necessary.
Frame Copy Errors	The number of times the adapter detected that a frame with its specific address has been copied by another adapter.
Frequency Errors	The number of times the adapter detected that the frequency of the incoming signal differs from the expected frequency by more than that allowed by the IEEE 802.5 standard. Check the active monitor responsible for master clocking of the ring and compensating for frequency jitter.

Item	Description
Hard Errors	The number of times the adapter either transmitted or received a beacon MAC frame.
Internal Errors	The number of times the adapter had an internal error.
Line Errors	The number of times the adapter detected an invalid character in a frame or token.
Lost Frame Errors	The number of times the adapter transmitted a frame and failed to receive it back.
Only Station	The number of times the adapter sensed that it is the only adapter on the ring.
Token Errors	The number of times the adapter, acting as an active monitor, detected that the token got lost. This may be due to ring reconfiguration. If this occurs often, check to see if other soft errors indicate a specific problem.
Remove Received	The number of times the adapter received a Remove Ring Station MAC frame request.
Ring Recovered	The number of times the ring is purged and recovered back into a normal operating state.
Signal Loss Errors	The number of times the adapter detected the absence of a receive signal.
Soft Errors	The number of times the adapter detected a soft error (recoverable by the MAC layer protocols).
Transmit Beacon Errors	The number of times the adapter transmitted a beacon frame.
Driver Flags	The device driver internal status flags currently turned on.

Device Specific Statistics Fields

This part of the display may be different for each type of adapter. It may contain adapter-specific information and some extended statistics that were not included in the generic statistics. Some adapters may not have any device-specific statistics. Some fields that may be listed in this section are:

Item	Description
ARI/FCI Errors	ARI/FCI mismatch is also referred to as receiver congestion. If an adapter gets an address match on a frame going by on the ring, Address Recognized Indication(ARI), and has no place into which to copy the frame, Frame Copied Indication(FCI), an ARI/FCI mismatch has occurred. The adapter will turn on the ARI bits but will not turn on the FCI bits in the FS byte at the end of the frame as it goes by. In other words, the adapter saw a frame that was to be received but, could not receive it because the receive buffers have been depleted. Two seconds later the adapter will send a Report Soft Error MAC frame indicating a receiver congestion error.
DMA Bus Errors	The number of times the adapter completed a DMA transfer and detected a bus error.

Item	Description
DMA Parity Errors	The number of times the adapter completed a DMA transfer and detected a parity error.
Receive Overruns	The number of times the adapter receive FIFO was full when the adapter tried to receive a frame.
Receive Underruns	The number of times the adapter transmit FIFO was empty before the end of frame symbol was detected.
Number of read log commands issued	The number of times an adapter error counter overruns (reached 255) and the device driver issues a read log command to read (and reset) the error counters.

Examples

1. To display the device driver statistics for **tok0**, enter:

```
tokstat tok0
```

This produces the following output:

```
TOKEN-RING STATISTICS (tok0) :
Device Type: Token-Ring High-Performance Adapter (8fc8)
Hardware Address: 10:00:5a:4f:26:c1
Elapsed Time: 0 days 0 hours 8 minutes 33 seconds
Transmit Statistics:          Receive Statistics:
-----
Packets: 191                  Packets: 8342
Bytes: 17081                  Bytes: 763227
Interrupts: 156               Interrupts: 8159
Transmit Errors: 0            Receive Errors: 0
Packets Dropped: 0           Packets Dropped: 0
Max Packets on S/W Transmit Queue: 17 Bad Packets: 0
S/W Transmit Queue Overflow: 0
Current S/W+H/W Transmit Queue Length: 0
```

```
Broadcast Packets: 1          Broadcast Packets: 8023
Multicast Packets: 0          Multicast Packets: 0
Timeout Errors: 0             Receive Congestion Errors: 0
Current SW Transmit Queue Length: 0
Current HW Transmit Queue Length: 0
```

```
General Statistics:
-----
No mbuf Errors: 0             Lobe Wire Faults: 0
Abort Errors: 0               AC Errors: 0
Burst Errors: 0               Frame Copy Errors: 0
Frequency Errors: 0           Hard Errors: 0
Internal Errors: 0            Line Errors: 0
Lost Frame Errors: 0          Only Station: 0
Token Errors: 0               Remove Received: 0
Ring Recovered: 0             Signal Loss Errors: 0
Soft Errors: 0                Transmit Beacon Errors: 0
Driver Flags: Up Broadcast Running
AlternateAddress ReceiveFunctionalAddr
```

2. To display the token-ring device driver statistics and the Token-Ring device-specific statistics for **tok0**, enter:

```
tokstat -d tok0
```

This produces the following output:

```
TOKEN-RING STATISTICS (tok0) :
Device Type: Token-Ring High-Performance Adapter (8fc8)
Hardware Address: 10:00:5a:4f:26:c1
Elapsed Time: 0 days 2 hours 48 minutes 38 seconds
```

```
Transmit Statistics:
-----
Packets: 389
Bytes: 42270
Interrupts: 354
Transmit Errors: 0
Packets Dropped: 0
Max Packets on S/W Transmit Queue:17
S/W Transmit Queue Overflow: 0
Current S/W+H/W Transmit Queue Length: 0

Receive Statistics:
-----
Packets: 153216
Bytes: 14583150
Interrupts: 151025
Receive Errors: 0
Packets Dropped: 0
Bad Packets: 0
```

```
Broadcast Packets: 1
Multicast Packets: 0
Timeout Errors: 0
Current SW Transmit Queue Length: 0
Current HW Transmit Queue Length: 0

Broadcast Packets: 152642
Multicast Packets: 0
Receive Congestion Errors: 0
```

```
General Statistics:
-----
No mbuf Errors: 0
Abort Errors: 0
Burst Errors: 0
Frequency Errors: 0
Internal Errors: 0
Lost Frame Errors: 0
Token Errors: 0
Ring Recovered: 0
Soft Errors: 0
Driver Flags: Up Broadcast Running
AlternateAddress ReceiveFunctionalAddr

Lobe Wire Faults: 0
AC Errors: 0
Frame Copy Errors: 0
Hard Errors: 0
Line Errors: 0
Only Station: 0
Remove Received: 0
Signal Loss Errors: 0
Transmit Beacon Errors: 0
```

```
Token-Ring High-Performance Adapter (8fc8) Specific Statistics:
-----
DMA Bus Errors: 0
ARI/FCI Errors: 0
DMA Parity Errors: 0
```

topas Command

Purpose

Reports selected local and remote system statistics.

Syntax

```
topas [ -d hotdisk ] [ -f hotfs ] [ -h ] [ -i interval ] [ -n hotni ] [ -p hotprocess ] [ -w hotwlmclass ] [ -c hotprocessor ] [ -I remotepollinterval ] [ -@ [ wparname ] ] [ -U username ] [ [ -C -D | -G | -F | -L | -P | -V | -T | -M | -t | -E | -W ] [ -m ]
```

Restriction: You cannot use the **-C**, **-L**, **-E**, **-V**, **-T**, **-t**, **-w**, **-W**, **-I**, **-@** options when you issue the command from a workload partition.

Description

The **topas** command reports selected statistics about the activity on the local system. The command uses the curses library to display its output in a format suitable for viewing on an 80x25 character-based display or in a window of at least the same size on a graphical display. The **topas** command requires the `bos.perf.tools` and **perfagent.tools** file sets to be installed on the system.

The `topas` command can also report a limited set of performance metrics from remote AIX partitions that belong to the same hardware platform. This support is described in the [Cross-Partition View](#) and [Cluster Utilization View](#) sections.

Note: For any dynamic configuration changes to the system, the tool must be restarted to reflect the new changes.

The **topas -D** command reports the disk details. This report is described in the [Disk Panel](#) section. You can run the subcommands from the Disk panel to display the following views:

Adapter Panel

Specified by pressing the **d** key. This panel provides details on the adapters and the disks that belong to the selected adapters.

Virtual Adapter Panel

Specified by pressing the **d** key and then the **v** key. This panel provides details of the virtual adapters that are related to the disks.

MPIO Panel

Specified by pressing the **m** key. This panel provides the details of the disks and the paths.

Panel Freezing

Specified by pressing the **space bar** key on the keyboard. The **space bar** key acts as a toggle for freezing the topas panel.

Scrolling

The Page Up and Page Down keys are used to scroll through the data.

Restriction: Adapter panel, Virtual Adapter panel, and MPIO panel are restricted inside WPAR.

If the **topas** command is invoked without flags, it runs as if invoked with the following command:

```
topas -d20 -i2 -n20 -p20 -w20 -c20 -f0
```

Note: The Central Electronic Complex (CEC) or cluster panel re-spawns when the migration or hibernation of the partition is complete. All other behavior for the CEC and any other panel remains the same in the event of migration or hibernation.

The program extracts statistics from the system with an interval specified by the *monitoring_interval_in_seconds* argument. The default output, as shown below, consists of two fixed parts and a variable section. The top two lines at the left of the display show the name of the system the **topas** command runs on, the date and time of the last observation, and the monitoring interval.

The second fixed part fills the rightmost 25 positions of the display. It contains the following subsections of statistics:

Item	Description
EVENTS/QUEUES	Displays the per-second frequency of selected system-global events and the average size of the thread run and wait queues: Cswitch The number of context switches per second over the monitoring interval. Syscalls The total number of system calls per second that are run over the monitoring interval. Reads The number of read system calls per second that are run over the monitoring interval. Writes The number of write system calls per second that are run over the monitoring interval. Forks The number of fork system calls per second that are run over the monitoring interval. Execs The number of exec system calls per second that are run over the monitoring interval. Runqueue The average number of threads that were ready to run but were waiting for a processor to become available. Waitqueue The average number of threads that were waiting for paging to complete.
FILE/TTY	Displays the per-second frequency of selected file and the TTY statistics. The following data is reported: Readch The amount of bytes read per second through the read system call over the monitoring interval. Writech The amount of bytes written per second through the write system call over the monitoring interval. Rawin The amount of raw bytes read per second from TTYs over the monitoring interval. Ttyout The amount of bytes written to TTYs per second over the monitoring interval. Igets The number of calls per second to the inode lookup routines over the monitoring interval. Namei The number of calls per second to the path name lookup routines over the monitoring interval. Dirblk The number of directory blocks scanned per second by the directory search routine over the monitoring interval.

Item	Description
PAGING	Displays the per-second frequency of paging statistics. The following data is reported: Faults The total number of page faults taken per second over the monitoring interval. This includes page faults that do not cause paging activity. Steals The physical memory 4 K frames stolen per second by the virtual memory manager over the monitoring interval. PgspIn The number of 4 K pages read from paging space per second over the monitoring interval. PgspOut The number of 4 K pages written to paging space per second over the monitoring interval. PageIn The number of 4 K pages read per second over the monitoring interval. This includes paging activity associated with reading from file systems. Subtract PgspIn from this value to get the number of 4K pages read from file systems per second over the monitoring interval. PageOut The number of 4 K pages written per second over the monitoring interval. This includes paging activity associated with writing to file systems. Subtract PgspOut from this value to get the number of 4K pages written to file systems per second over the monitoring interval. Sios The number of I/O requests per second issued by the virtual memory manager over the monitoring interval.
MEMORY	Displays the real memory size and the distribution of memory in use. The following data is reported: Real,MB The size of real memory in megabytes. % Comp The percentage of real memory currently allocated to computational page frames. Computational page frames are generally those that are backed by paging space. % Noncomp The percentage of real memory currently allocated to non-computational frames. Non-computational page frames are generally those that are backed by file space, either data files, executable files, or shared library files. % Client The percentage of real memory currently allocated to cache remotely mounted files.
PAGING SPACE	Displays the size and use of paging space. The following data is reported: Size,MB The sum of all paging spaces on the system, in megabytes. % Used The percentage of total paging space currently in use. % Free The percentage of total paging space currently free.

Item	Description
NFS	Displays the NFS statistics in calls per second. The following data is reported: • Server V2 calls/sec • Client V2 calls/sec • Server V3 calls/sec • Client V3 calls/sec
Total WPAR	Displays the total number of workload partitions that are defined in the system. The total amount of workload partitions can be in the following states: Defined , Active , Broken or Transition .
Active WPAR	Displays the total number of resource active workload partitions.
AME	Displays memory compression statistics in an Active Memory Expansion enabled system. The following data is reported: TMEM,MB True Memory Size, in megabytes. CMEM,MB Compressed Pool Size, in megabytes. EF[T/A] Expansion Factors: Target & Actual. CI Compressed Pool Page-Ins. CO Compressed Pool Page-Outs.

The variable part of the **topas** display can have one, two, three, four, or five subsections. If more than one subsection displays, they are always shown in the following order:

- [Processor utilization](#)
- [Network interfaces](#)
- [Physical disks](#)
- [File system](#)
- [Workload Manager classes](#)
- [workload partitions](#)
- [Processes](#)

When the **topas** command is started, it displays all subsections for which hot entities are monitored. The Workload Manager (WLM) Classes subsection is displayed only when WLM is active.

The WLM should be started to view the WLM and WPAR statistics.

Tip: When there is no WPAR specific information for a metric, the system-wide value is displayed for that metric in inverted background (that is, white text and black context).

The following table provides the details for the subsections that the **topas** command displays:

Item	Description
Processor utilization	This subsection displays one-line report summary of all the processor usage. Pressing the c key only once turns this subsection off. If more than one processor exists, a list of processors is displayed by pressing the c key twice. Pressing the c key thrice displays a bar chart showing cumulative processor usage. The following fields are displayed by both formats: User% The percentage of processor used by programs running in user mode. (Default sorted by User%) Kern% The percentage of processor used by programs running in kernel mode. Wait% The percentage of time spent in waiting for I/O. Idle% The percentage of time that the processors are idle. PhySc The number of physical processors that are consumed. Displayed only if the partition is running with shared processor. %Entc The percentage of entitled capacity that is consumed. Displayed only if the partition is running with shared processor. When this subsection displays the list of hot processors, the list is sorted by the User% field. However, the list can be sorted by the other fields by moving the cursor to the top of the desired column.
Network interfaces	This subsection shows a one-line report summary of the activity for all network interfaces. Pressing the n key once turns off this subsection. Pressing the n key twice displays a list of active network interfaces. The maximum number of interfaces displayed is the number of active interfaces being monitored, as specified by using the -n flag. A smaller number of interfaces are displayed if other subsections are also being displayed. Both reports display the following fields: BPS The total throughput in kilobytes per second over the monitoring interval. This field is the sum of kilobytes received and kilobytes sent per second. Interf The name of the network interface. I-Pack The amount of data packets received per second over the monitoring interval. KB-In The number of kilobytes received per second over the monitoring interval. KB-Out The number of kilobytes sent per second over the monitoring interval. O-Pack The amount of data packets sent per second over the monitoring interval. When this subsection displays the list of hot network interfaces, the list is sorted by the BPS field. However, the list can be sorted by the other fields by moving the cursor to the top of the desired column. Sorting is only valid for up to 16 network adapters.

Item	Description
Physical disks	This subsection shows a one-line report summary of the activity for all physical disks. Pressing the d key turns once off this subsection. Pressing the d key again displays a list of active physical disks. The maximum number of physical disks displayed is the number of active physical disks being monitored, as specified by using the -d flag. A smaller number of physical disks is displayed if other subsections are also being displayed. Both reports display the following fields: Busy% The percentage of time the physical disk is active (bandwidth use of the drive). BPS The amount of data transferred (read and written) in kilobytes per second over the monitoring interval. This field is the sum of the values of the KB-Read and KB-Writ. Disk The name of the physical disk. KB-Read The number of kilobytes read per second from the physical disk. KB-Writ The number of kilobytes written per second to the physical disk. TPS The number of transfers per second that were issued to the physical disk. A transfer is an I/O request to the physical disk. Multiple logical requests can be combined into a single I/O request to the disk. A transfer is of indeterminate size. When this subsection displays the list of hot physical disks, the list is sorted by the BPS field. However, the list can be sorted by the other fields by moving the cursor to the top of the desired column. Sorting is only valid for up to 128 physical disks.

Item	Description
File system	This subsection shows a one-line report summary of the activity for all of the file systems. Pressing the f key once turns off this section. Pressing the f key twice displays a list of active file systems. The maximum number of file systems that are displayed is the number of active file systems that are monitored when they are specified by using the -f flag. A smaller number of file systems are displayed if other subsections are also being displayed. Both reports display the following fields: BPS The amount of data transferred (read and written) in kilobytes per second over the monitoring interval. This field is the sum of the values of the KB-Read and KB-Writ fields. File System The name of the file system. KB-Read The number of kilobytes read per second from the file system. KB-Writ The number of kilobytes written per second to the file system. TPS The number of transfers per second that are issued to the file system. A transfer is an I/O request to the file system. Multiple logical requests can be combined into a single I/O request to the file system. The size of a transfer is not determinate. When this subsection displays the list of the file systems, the list is sorted by the BPS field. However, the list can be sorted by the other fields by moving the cursor to the top of the target column. Tip: If the file system name exceeds the field width in the display, then the file system name is displayed in a truncated format. The truncation contains the first and last few characters of the file system, the middle part of the name is replaced by periods (..). For example, if the file system name is <code>filesystem001234</code>, then the name is displayed as <code>files..01234</code>.

Item	Description
WLM classes	This subsection displays a list of hot Workload Manager (WLM) Classes. The maximum number of WLM classes displayed is the number of hot WLM classes being monitored as specified with the -w flag. A smaller number of classes will be displayed if other subsections are also being displayed. Pressing the w key turns off this subsection. The following fields are displayed for each class: % processor Utilization The average processor use of the WLM class over the monitoring interval. % Mem Utilization The average memory use of the WLM class over the monitoring interval. % Blk I/O The average percent of block I/O of the WLM class over the monitoring interval. When this subsection first displays the list of hot WLM classes, the list will be sorted by the CPU% field. However, the list can be sorted by the other fields by moving the cursor to the top of the desired column. Tip: If the WLM class name exceeds the field width in the display, the WLM class name is truncated. The truncation contains the first and last few characters of the WLM class, and the middle part of the name is replaced by periods (..). For example, if the WLM class name is unclassified00123, then the WLM class name is displayed as uncla. .00123.
Workload partitions	The workload partitions subsection replaces WLM subsection if invoked with the -@ flag. This subsection displays a list of hot workload partitions. The maximum number of workload partitions that are displayed is the number of hot WPAR that are monitored (when they are specified with the -w -@ flag). A smaller number of WPAR is displayed if other subsections are also being displayed. To turn off the workload partitions subsection, press the @ key. The following fields are displayed for each WPAR: WPAR The name of the workload partition (WPAR). % processor Utilization The average processor use of the WPAR over the monitoring interval. % Mem Utilization The average memory use of the WPAR over the monitoring interval. % Blk I/O The average percent of block I/O of the WPAR over the monitoring interval. When this subsection displays the list of hot WPAR, the list is sorted by the CPU% field. However, the list can be sorted by the other fields by moving the cursor to the top of the target column that you want to use to sort the list. Tip: If the WPAR name exceeds the field width in the display, the WPAR name is truncated. The truncation contains the first and last few characters of the WPAR, and the middle part of the name is replaced by periods (..). For example, if the WPAR name is neptune00123, then the WPAR is displayed as neptu. .00123.

Item	Description
Processes	This subsection displays a list of hot processes. The maximum number of processes displayed is the number of hot processes being monitored as specified with the -p flag. A smaller number of processes will be displayed if other subsections are also being displayed. Pressing the p key turns off this subsection. The processes are sorted by their processor usage over the monitoring interval. The following fields are displayed for each process: Name The name of the executable program executing in the process. The name is stripped of any pathname and argument information and truncated to 9 characters in length. Process ID The process ID of the process. % CPU Utilization The average processor use of the process over the monitoring interval. The first time a process is shown, this value is the average processor use over the lifetime of the process. Paging Space Used The size of the paging space allocated to this process. This can be considered an expression of the footprint of the process but does not include the memory used to keep the executable program and any shared libraries it may depend on. Process Owner (if the WLM section is off) The user name of the user who owns the process. Workload Manager (WLM) Class (if the WLM section is on) The WLM class to which the process belongs. WPAR (if the WPAR section is on) The WPAR name that the process belongs to. Tip: If the WLM Class/WPAR name exceeds the field width in the display, the WLM Class/WPAR name is truncated. The truncation contains the first and last few characters of the WLM Class/WPAR, and the middle part of the name is replaced by periods (...). For example, if the WLM Class/WPAR name is unclassified00123, then the WLM Class/WPAR name is displayed as uncla..00123.

Adapter Panel View

When you use the **topas -D** command, you can press the **d** key to display the Adapter panel view. In this panel, the following metrics are displayed:

Item	Description
Adapter	The name of the adapter.
KBPS	The amount of data transferred (read or written) in the adapter in kilobytes per second.
TPS	Indicates the average number of transfers per second that the adapter issues.
KB-R	The total number of kilobytes that are read from the adapter.
KB-W	The total number of kilobytes that are written to the adapter.

If you press the **f** key, the following details of the disks that belong to the adapter are displayed on the Adapter panel:

Item	Description
AQD	The average number of requests that are waiting to be sent to the virtual target device or disk.
AQW	The average time spent by a transfer request in the waiting queue. The suffix indicates the unit of time. The default unit of time is millisecond.
ART	The average time to receive a response from the hosting for the read request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
AWT	The average time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
Busy%	The percentage of time the virtual target device or disk is active (bandwidth use of the virtual target device or disk).
KBPS	The amount of data that is read and written in kilobytes per second over the monitoring interval. This field is the sum of the values of the KB-R and KB-W metrics.
KB-R	The number of kilobytes per second that are read from the virtual target device or disk.
KB-W	The number of kilobytes per second that are written to the virtual target device or disk.
MRT	The maximum time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
MWT	The maximum time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
TPS	The number of transfers per second that are issued to the virtual target device or disk. A transfer is an I/O request to the virtual target device or disk. Multiple logical requests can be combined into a single I/O request to the disk. A transfer is of medium size.
Vtargets/Disk	The name of the virtual target device or disk.

Virtual Adapter Panel View

When you run the **topas -D** command, you can press the **v** key to display the Virtual Adapter panel view. In this panel, the following metrics are displayed:

Item	Description
AQD	The average number of requests waiting to be sent to the adapter.
AQW	The average time spent by a transfer request in the waiting queue. The suffix indicates the unit of time. The default unit of time is millisecond.
ART	The average time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
AWT	The average time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
KBPS	The amount of data transferred (read or written) in kilobytes per second in the adapter.
KB-R	The number of blocks received per second from the hosting server to the adapter.
KB-W	The number of blocks sent per second from this adapter to the hosting server.

Item	Description
MRT	The maximum time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
MWT	The maximum time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
TPS	The number of transfers per second that are issued to the adapter.
vAdapter	The name of the virtual adapter.

If you press the **f** key, the following details of the disks that belong to the adapter are displayed on the Virtual Adapter panel:

Item	Description
AQD	The average number of requests that are waiting to be sent to the virtual target device or disk.
AQW	The average time spent by a transfer request in the waiting queue. The suffix indicates the unit of time. The default unit of time is millisecond.
ART	The average time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
AWT	The average time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
Busy%	The percentage of time the virtual target device or disk is active (bandwidth use of the virtual target device or disk).
KBPS	The amount of data that is read and written in kilobytes per second over the monitoring interval. This field is the sum of the values of the KB-R and KB-W metrics.
KB-R	The number of kilobytes that are read per second from the virtual target device or disk.
KB-W	The number of kilobytes that are written per second to the virtual target device or disk.
MRT	The maximum time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
MWT	The maximum time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
TPS	The number of transfers per second that are issued to the virtual target device or disk. A transfer is an I/O request to the virtual target device or disk. Multiple logical requests can be combined into a single I/O request to the disk. A transfer is of medium size.
Vtargets/Disk	The name of the virtual target device or disk.

MPIO Panel View

When you use the **topas -D** command, you can press the **m** key to display the MPIO panel view. In this panel, the top section contains the same metrics that the Disks panel displays.

The bottom section of the panel contains the following fields:

Item	Description
Busy%	The percentage of time the path is active (bandwidth use of the path).
KBPS	The amount of data that is read and written in kilobytes per second over the monitoring interval. This field is the sum of the values of the KB-R and KB-W metrics.
KB-R	The number of kilobytes that is read per second in that path.
KB-W	The number of kilobytes that is written per second in that path.
Path	The name of the path.
TPS	The number of transfers per second that are issued in that path.

Panel Freezing

The space bar key on the keyboard acts as a toggle for freezing the topas panel. If frozen, topas stops data collection and continues to display the data from the previous iteration. You can move around the panel and sort the data based on the selected column. In frozen state, if you move between panels, some panels may not display the data. In this case, press the space bar key to unfreeze the topas panel.

Scrolling

If the amount of data is more than the topas window size, then Page Up and Page Down keys are used to scroll through the data. The data is sorted based on the selected column.

Note: The above functionality is available with selected panels in topas.

I/O Memory Entitlement Pools Panel

When a Logical Partition panel (**topas -L**) is enabled in shared-memory mode, you can press the **e** key to display the I/O Memory Entitlement Pools panel.

The following metrics are displayed in the lower section of this panel:

Item	Description
iompn	The name of the I/O memory pool.
iomin	The minimum I/O memory entitlement of the pool.
iodes	The desired I/O memory entitlement of the pool.
ioinu	The current I/O memory entitlement of the pool.
iores	The reserved I/O memory entitlement of the pool.
iohwm	The maximum I/O memory entitlement in use for the pool (high water mark).
ioafl	The total number of times the allocation requests have failed for this pool.

Cross-Partition View and Recording

This panel displays metrics similar to the `lparstat` command for all the AIX partitions it can identify as belonging to the same hardware platform. Dedicated and shared partitions are displayed in separate sections with appropriate metrics. The top section represents aggregated data from the partition set to show overall partition, memory, and processor activity.

Remote enablement for this panel to collect from other partitions requires to use the latest updates to the **perfagent.tools** and **bos.perf.tools** to support this function. For earlier versions of AIX, the **topas** command also collects remote data from partitions that have the Performance Aide product (**perfagent.server**) installed. The **topas -C** command may not be able to locate partitions residing on

other sub-nets. To avoid this, create a **\$HOME/Rsi.hosts** file containing the fully qualified host names for each partition (including domains), one host per line.

Note: The **topas -C** command sends broadcast packet to all the Logical Partitions (LPARs) in the same subnet, but only processes response from the LPARs within the same CEC.

The following metrics display in the initial cross-partition panel. Additional metrics with full descriptive labels can be displayed by using the key toggles identified in the Additional cross-partition panel subcommands section:

Partition totals:

Item	Description
Shr	The number of shared partitions based on the system processor.
Ded	The number of dedicated partitions based on the system processor.

Memory (in GB):

Item	Description
Mon	The total memory of monitored partitions.
Avl	The memory available to partition set.
InUse	The memory in use on monitored partitions.

Processor:

Item	Description
Shr	The number of shared processors.
Ded	The number of dedicated processors.
PSz	The number of shared physical CPUs in the system.
APP	Indicates the available physical processors in the system (default shared processor pool). Note: Default shared processor pool contains the physical processors that are available on the managed system. The topas command retrieves the APP value from the data that is provided by the LPARs that are in the same managed system. If these LPARs do not belong to the default shared processor pool, the topas command cannot determine the APP value for the managed system. The APP value is indicated by the - (hyphen) character in this case.
Don	The total number of processors donated to the pool
Shr_PhysB	The total number of physical processors that are consumed by all shared partitions
Ded_PhysB	The total number of physical processors that are consumed by all dedicated partitions

Individual partition data:

Item	Description
Host	The host name
OS	The operating system level
Mod	The mode of the individual partitions. The mode is displayed in a set of 3 characters.

Item	Description
Character	The first character indicates the CPU in the partition. The second character indicates the memory mode of the partition. The third character indicates the energy state of the partition.
Mem	The total memory measured in gigabytes.
InU	The memory in use measured in gigabytes.
Lp	The number of logical processors.
Us	The percentage of processor used by programs executing in user mode.
Sy	The percentage of processor used by programs executing in kernel mode.
Wa	The percentage of time spent waiting for I/O.
Id	The percentage of time the processors are idle.
PhysB	The number of physical processors that are consumed by each partition.
Ent	The entitlement granted (shared-only).
%Entc	The percent entitlement consumed (shared-only).
Vcsw	The average of virtual context switches per second (shared-only).
PhI	The average of phantom interrupts per second (shared-only).
Pmem	The physical memory that is backing the partitions logical memory (if in shared-memory mode).
%idon	The percentage of physical processor that is used while explicitly donating idle cycles. This metric is applicable only for donating dedicated partitions.
%bdon	The percentage of physical processor that is used while busy cycles are being donated. This metric is applicable only for donating dedicated partitions.
%istl	The percentage of physical processor that is used while idle cycles are being stolen by the hypervisor. This metric is applicable only for dedicated partitions.
%bstl	The percentage of physical processor that is used while busy cycles are being stolen by the hypervisor. This metric is applicable only for dedicated partitions.

For shared partitions:

First Character	Description
C	SMT enabled and capped
c	SMT disabled and capped
U	SMT enabled and uncapped
u	SMT disabled and uncapped

For dedicated partitions:

First Character	Description
S	SMT enabled and not donating

First Character	Description
d	SMT disabled and donating
D	SMT enabled and donating
-	SMT disabled and not donating

Second Character	Description
M	AMS enabled and AME disabled
-	AME and AMS disabled
E	AME enabled and AMS enabled
e	AME enabled and AMS disabled

Third Character	Description
S	Static power save mode is enabled
d	Power save mode is disabled
D	Dynamic power save mode is enabled
-	Unknown / Undefined
E	Power save mode has been enabled
d	Power save mode has been disabled

The **%idon** and **%bdon** metrics are not displayed when there is no donating dedicated partition.

Requirement: At least one partition to be monitored must have Pool Utilization Authority (PUA) configured for pool information metrics to be collected.

For cross-partition monitoring/recording, some global data is not available from any partition. The **-o** option allows you to specify these fields in the command line. Optionally, you can configure a system to allow the **topas** command to query the HMC directly for this information. This requires the following steps:

1. Install OpenSSH at the partition.
2. Enable remote command support on the HMC for user **hscroot** to allow **ssh** connections to be opened from the partition.
3. Configure **ssh** on the HMC to not require a password for the HMC user **hscroot** when queried from the selected partition. This requires the **.ssh/authorized_keys2** on the HMC for user login **hscroot**.
4. Run **ssh -l hscroot hmc_address date** from the partition to confirm whether the date is displayed without requiring that a password be entered.
5. Utilize the **topas -o** options described in the usage table to specify the managed system and HMC names when running the **topas** command.

Restriction: This functionality is currently available only for HMC version 5 and above, and should only be enabled after careful consideration of any security implications.

The following displays when press the **g** key in the initial screen, which brings the cross partition view with detailed headers:

```

Topas CEC Monitor      Interval: 10      Mon Jan 22 00:08:00 2007
Partition Info  Memory (GB)  Processor  Virtual Pools : 2
Monitored : 2   Monitored : 6.2   Monitored : 2.0   Avail Pool Proc: 5
UnMonitored: -   UnMonitored: -   UnMonitored: -   Shr Physical Busy: 0.00
Shared : 0      Available : -   Available : -   Ded Physical Busy: 0.05
Uncapped : 0    UnAllocated: -   UnAllocated: -   Donated Phys. processors: 0.00
Capped : 2     Consumed : 1.9   Shared : 0      Stolen Phys. processors : 0.01
Dedicated : 2   Dedicated : 2    Hypervisor

```

```

Donating   : 0                               Donated    : 0   Virt. Context Switch: 347
Pool Size  : 0                               Phantom Interrupts : 0

Host       OS  M Mem InU Lp  Us Sy Wa Id  PhysB  Vcsw Ent  %EntC PhI
-----
ptoolsl1   A53 U 3.1 1.9 4    1 2 0 96  0.01 398 0.20 5.3 0k
Host       OS  M Mem InU Lp  Us Sy Wa Id  PhysB  Vcsw %istl %bstl %bdon %idon
-----
ptoolsl1   A54 S 3.1 0.9 2    0 0 0 99  0.00 177  0.1  0.0  0.0  0.0
ptoolsl3   A54 S 3.1 0.9 2    0 0 0 99  0.00 170  0.2  0.0  0.0  0.0

```

The following headers are in the previous screen:

Partition Info:

Item	Description
Monitored	The number of partitions that are monitored
Unmonitored	The number of partitions that are not monitored
Shared	The number of shared partitions
Uncapped	The number of uncapped shared partitions
Capped	The number of capped partitions
Dedicated	The number of dedicated partitions
Donating	The number of partitions that are currently donating

Memory:

Item	Description
Monitored	The total memory that is monitored
UnMonitored	The total memory that is not monitored
Available	The total memory that is available
UnAllocated	The total memory that is not allocated to any partition
Consumed	The total memory that is consumed by the partitions

Processor:

Item	Description
Monitored	The number of physical processors that are monitored
UnMonitored	The number of physical processors that are not monitored
Available	The number of physical processors that are available in CEC system
UnAllocated	The number of physical processors that are not allocated to any partition
Shared	The number of processors that are in shared partitions
Dedicated	The number of processors that are in dedicated partitions
Donated	The sum of the number of processors in all the partitions donating
Pool Size	The number of shared physical CPUs in the system.

Item	Description
Avail Proc Pool	Indicates the available physical processors in the system (default shared processor pool). Note: Default shared processor pool contains the physical processors that are available on the managed system. The topas command retrieves the APP value from the data that is provided by the LPARs that are in the same managed system. If these LPARs do not belong to the default shared processor pool, the topas command cannot determine the APP value for the managed system. The APP value is indicated by the - (hyphen) character in this case.
Shr Physical Busy	The sum of physical busy of all of the shared partitions
Ded Physical Busy	The sum of dedicated busy of all of the dedicated partitions
Donated Phys. processors	The sum of the donated processor cycles from all of the partitions reported as a number of processors
Stolen Phys. processors	The sum of stolen processor cycles from all of the partitions reported as a number of processors
Virtual Pools	The number of virtual pools
Virt. Context Switch	The total number of virtual context switches per second in the monitoring interval
Phantom Interrupts	The total number of phantom interrupts per second in the monitoring interval

When the **topas** command is running inside any cross partition view, press the **p** key to bring up the pool panel. The following is an example that displays:

```
pool  psize entc maxc physb app  mem  muse
0     3.0  2.0  4.0  0.1  2.0  1.0  1.5
1     4.0  3.0  5.0  0.5  1.5  1.0  0.5
2     3.0  2.5  4.0  0.2  2.0  1.0  0.5
```

You can scroll up or down in the pool ID column and press the **f** key to list only the shared partitions that belong to the **poolid** where cursor is positioned. The following headers might be displayed in the screen:

Item	Description
psize	The effective maximum capacity of the pool
entc	The entitled capacity of the pool
maxc	The maximum capacity of the pool
physb	The sum of physical busy of processors in shared partitions of a pool
app	The available physical processor in the pool
mem	The sum of monitored memory for all shared partitions in the pool
muse	The sum of memory consumed for all shared partitions in the pool

When the **topas** command is running inside any cross-partition view, press the **v** key to display the **Virtual I/O Server/Client Throughput panel**. The following metrics are displayed:

Item	Description
AQD	The average number of requests that are waiting to be sent.
AQW	The average time spent by a transfer request in the waiting queue. The suffix indicates the unit of time. The default unit of time is millisecond.

Item	Description
ART	The average time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
AWT	The average time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
Client	The name of the VIO Client.
KBPS	The amount of data that is read and written in kilobytes per second over the monitoring interval. This field is the sum of the values of the KB-R and KB-W metrics.
KB-R	The number of kilobytes that are read per second.
KB-W	The number of kilobytes that are written per second.
MRT	The maximum time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
MWT	The maximum time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default unit of time is millisecond.
Server	The name of the VIO Server.
TPS	The number of transfers that are issued per second.

When the **topas** command is running inside the Virtual I/O Server/Client Throughput panel, press the **d** key after selecting a server from the Virtual I/O Server/Client Throughput panel to toggle to **VIO Server/Client Disk Details** panel. This panel displays the server adapter details in the top section and displays the target device and client disk details in the bottom of the section. To list the target devices and client disks belong to that adapter, select the adapter and press the **f** key.

The following metrics are displayed in a Virtual I/O Server/Client Disk Details panel:

Item	Description
Adapter	The name of the server adapter.
Vtargets	The name of the virtual target device that belongs to the server adapter.
Client_disk	The name of the client disk that is mapped to the virtual target device of the server adapter.

The following details of the adapters are displayed on the top section of the panel:

Item	Description
KBPS	The amount of data transferred (read or written) in the adapter in kilobytes per second.
TPS	The number of transfers per second that are issued to the adapter.
KB-R	The total number of kilobytes read from the adapter.
KB-W	The total number of kilobytes written to the adapter.
AQD	The number of requests waiting to be sent to the adapter.
AQW	The average time spent by a transfer request in the waiting queue. The suffix indicates the unit of time. The default time unit is millisecond.
ART	The time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.
AWT	The time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.

Item	Description
MRT	The maximum time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.
MWT	The maximum time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.

The following details for the virtual target device and client disk are displayed on the panel:

Item	Description
Busy%	The percentage of time the that the virtual target device or disk is active (bandwidth use of the virtual target device or disk).
KBPS	The number of kilobytes read and written per second over the monitoring interval. This field is the sum of the values of the KB-R and KB-W metrics.
TPS	The number of transfers per second that are issued to the virtual target device or disk. A transfer is an I/O request to the virtual target device or disk. Multiple logical requests can be combined into a single I/O request to the virtual target device or disk. A transfer is of medium size.
KB-R	The number of kilobytes read per second from the virtual target device or disk.
KB-W	The number of kilobytes written per second to the virtual target device or disk.
AQD	The average number of requests waiting to be sent to virtual target device or disk.
AQW	The average time spent by a transfer request in the waiting queue. The suffix indicates the unit of time. The default time unit is millisecond.
ART	The average time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.
AWT	The average time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.
MRT	The maximum time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.
MWT	The maximum time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.

To display the **Memory Pool panel** from the CEC panel, press the **m** key. This panel displays the statistics of all of the memory pools in the system. To display the partitions corresponding to that pool in the lower section of the panel, select a particular memory pool and press the **f** key.

The following values are displayed in the header section of the panel:

Item	Description
Mshr	The number of logical partitions (LPAR) running in the shared-memory mode.
Mded	The number of LPAR running in dedicated-memory mode.
Pools	The total number of memory pools in the system.
Mpsz	The total size of physical memory of all the memory pools in gigabytes.
MPuse	The total memory used by LPAR associated with all of the pools in gigabytes.
Entl	The total I/O memory entitlement of all of the LPAR in all the pools in gigabytes.

Item	Description
Use	The total I/O memory entitlement in use of all of the LPAR in all the pools in gigabytes.
Mon	The total monitored memory of the system (sum of the values of the Mpsz metric and the Total memory of dedicated memory partitions metric).
InUse	The total memory in use of the system (sum of the MPuse metric and Total memory inuse for dedicated memory partitions metric).
Avl	The total memory available for the system (the value of the Mon metric minus the value of the InUse metric).

The following values of the pools are displayed:

Item	Description
mpid	The ID of the memory pool.
mps	The size of the total physical memory of the memory pool in gigabytes.
mpus	The total memory of the memory pool in use (this is the sum of the physical memory allocated to all of the LPAR in the pool).
mem	The size of the aggregate logical memory of all the partitions in the pool in gigabytes.
memu	The aggregate logical memory that is used for all the partitions in the pool in gigabytes.
iome	The aggregate of I/O memory entitlement that is configured for all the LPAR in the pool in gigabytes.
iomu	The aggregate of the I/O memory entitlement that is used for all the LPAR in the pool in gigabytes.
hpi	The aggregate number of hypervisor page faults that have occurred for all of the LPAR in the pool.
hpit	The aggregate of time spent in waiting for hypervisor page-ins by all of the LPAR in the pool in milliseconds.

The following values of the partitions in the pools are displayed:

Item	Description
mem	The size of logical memory of the partition in gigabytes.
memu	The logical memory that is used for the partition in gigabytes.
meml	The logical memory loaned to hypervisor by the LPAR.
pmem	The physical memory that is allocated to the partition from the memory pool in gigabytes.
iom	The amount of I/O memory entitlement that is configured for the LPAR in gigabytes.
iomu	The amount of I/O memory entitlement that is used for the LPAR in gigabytes.
hpi	The number of hypervisor page faults.
hpit	The time spent in waiting for hypervisor page-ins in milliseconds.
vcs	The virtual context switches average per second.
physb	The physical processor that is busy.
%entc	The percentage of the consumed processor entitlement.

Cluster Utilization View

A cluster is a group of related partitions or nodes. The Cluster Utilization view can either show utilization of an HA cluster or a user-defined cluster. This panel displays metrics similar to the `lparstat` command for all the AIX partitions it can identify as belonging to the same hardware platform. The dedicated and shared partitions are displayed in separate sections with appropriate metrics. The top section represents aggregated data from the partition set to show overall partition, memory, and processor activity.

The following metrics are displayed in an initial cluster utilization panel. Additional metrics with full descriptive labels can be displayed using the key toggles identified in the [Additional Cluster Utilization Panel Subcommands](#) topic.

Partition totals:

Item	Description
Shr	The number of shared partitions based on the system processor.
Ded	The number of dedicated partitions based on the system processor.

Memory (in GB):

Item	Description
Mon	The total memory of monitored partitions.
InUse	The memory in use on monitored partitions.

Processor:

Item	Description
Shr	The number of shared processors.
Ded	The number of dedicated processors.
Shr_PhysB	The total number of physical processors that are busy for all shared partitions.
Ded_PhysB	The total number of physical processors that are busy for all dedicated partitions.

Individual partition data:

Item	Description
Host	The host name.
CEC	The CEC identifier.
OS	The operating system level
Mem	The total memory measured in gigabytes.
M	The mode of the individual partitions.
InU	The memory in use measured in gigabytes.
Lp	The number of logical processors.
Us	The percentage of the processor used by programs executing in user mode.
Sy	The percentage of the processor used by programs executing in kernel mode.
Wa	The percentage of time spent waiting for I/O.
Id	The percentage of time the processors are idle.

Item	Description
PhysB	The number of physical processors that are busy.
Ent	The entitlement granted (shared-only).
%Entc	The percentage entitlement consumed (shared-only).
Vcsw	The average of virtual context switches per second (shared-only).

For shared partitions

Character	Description
C	SMT enabled and capped
c	SMT disabled and capped
U	SMT enabled and uncapped
u	SMT disabled and uncapped

For dedicated partitions

Character	Description
S	SMT enabled and not donating
d	SMT disabled and donating
D	SMT enabled and donating
-	SMT disabled and not donating

The following data is displayed when you press the **g** key on the initial screen, which generates the cluster utilization view with detailed headers:

Topas CEC Cluster Monitor ID:										Interval: 10					Thu Apr 2 16:13:18 2009				
Partitions					Memory (GB)					Processor									
Shr :2					Mon : 6.0					Shr :1.5					Shr_PhyB : 0.01				
Ded :2					InU : 3.0					Ded :2					Ded_PhyB : 0.00				
Host	CEC	OS	M	Mem	InU	Lp	Us	Sy	Wa	Id	PhysB	Vcsw	Ent	%EntC					
-----shared-----																			
clock16	19318230	A61	U	2.0	1.1	2	0	0	0	99	0.00	423	0.75	0.6					
clock15	19318230	A61	U	2.0	1.6	2	0	0	0	99	0.01	985	0.75	0.9					
Host	CEC	OS	M	Mem	InU	Lp	Us	Sy	Wa	Id	PhysB	Vcsw							
-----dedicated-----																			
ses10	19318230	A61	D	2.0	1.1	2	0	0	0	99	0.00	0							
clock10	19318230	A61	D	0.0	0.0	2	0	0	0	99	0.00	742							
The following display when press g key from the above panel, which brings the cluster utilization view with detailed headers:																			
Topas Cluster Monitor ID:										Interval: 10					Thu Apr 2 16:13:44 2009				
Partition Info					Memory (GB)					Processor					Supplier: ses10.in.ibm.com				
Monitored :4					Monitored:6.0					Monitored :3.5					Shr Physical Busy :0.01				
Shared :2					Consumed :3.0					Shared :1.5					Ded Physical Busy :0.00				
Uncapped :2										Dedicated :2									
Capped :2																			
Dedicated :2																			
Host	CEC	OS	M	Mem	InU	Lp	Us	Sy	Wa	Id	PhysB	Vcsw	Ent	%EntC					
-----shared-----																			
clock16	19318230	A61	U	2.0	1.1	2	0	0	0	99	0.00	423	0.75	0.6					
clock15	19318230	A61	U	2.0	1.6	2	0	0	0	99	0.01	985	0.75	0.9					
Host	CEC	OS	M	Mem	InU	Lp	Us	Sy	Wa	Id	PhysB	Vcsw							
-----dedicated-----																			
ses10	19318230	A61	D	2.0	1.1	2	0	0	0	99	0.00	0							
clock10	19318230	A61	D	0.0	0.0	2	0	0	0	99	0.00	742							

Implementation Specifics

Disks and network adapters added after starting **topas** or any other SPMI consumer will not be reflected in **topas**. You must stop **topas** and all clients that use SPMI and then restart after the changes to disks and network adapters are made.

Flags

Item	Description
<code>-@wparname</code>	Shows the WPAR-specific metrics. If you specify a WPAR name with the <i>wparname</i> parameter, the topas monitors that WPAR.
<code>-chotprocessor</code>	Specifies with the <i>hotprocessor</i> parameter the number of hot processors to be monitored. This is also the maximum number of processors displayed when enough room is available on the screen. If this number exceeds the number of processors available, only the installed processors will be monitored and displayed. If this argument is omitted, a default of 2 is assumed. If a value of 0 (zero) is specified, no processor information is monitored.
<code>-C</code>	Displays the Cross-partition panel. The topas command collects a set of metrics from AIX partitions running on the same hardware platform. The metrics are similar to those collected by the lparstat command. Dedicated and shared partitions are displayed, and a set of aggregated values provide an overview of the entire hardware systems partition set. Certain values only available from the HMC platform can be set through the line command if an HMC connection is not available.
<code>-G</code>	Displays the Cluster Utilization panel. The topas command collects a set of metrics from AIX partitions that are running on the same hardware platform. The metrics are similar to those collected by the lparstat command. Dedicated and shared partitions are displayed.

Item**-D****Description**

Displays the Disk Metrics display (Disk panel view). The display reports disk service times, disk queuing metrics, and disk throughput. The following metrics are reported:

Disk

The name of the physical disk.

Busy%

The percentage of time that the physical disk is active (bandwidth use for the disk).

KBPS

The number of kilobytes that are read and written per second over the monitoring interval. This field is the sum of the values of the **KB-R** and **KB-W** metrics.

TPS

The number of transfers per second that are issued to the physical disk. A transfer is an I/O request to the physical disk. Multiple logical requests can be combined into a single I/O request to the disk. A transfer is of medium size.

KB-R

The number of kilobytes read per second from the physical disk.

ART

The average time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.

MRT

The maximum time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.

KB-W

The number of kilobytes written per second to the physical disk.

AWT

The average time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.

MWT

The maximum time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.

AQW

The average time spent by a transfer request in the waiting queue. The suffix indicates the unit of time. The default time unit is millisecond.

AQD

The average number of requests that are waiting to be sent to disk.

With the **-D** flag specified, you can run the following subcommands:

- To view the **Adapter Panel**, press the **d** key.
- To display all of the virtual adapters present in the partition (**Virtual Adapter Panel**), press the **v** key.
- To display the disks that belong to the adapter or the virtual adapter, press the **f** key.
- To display the **MPIO Panel**, press the **m** key. This panel displays the disks details and the path details. To list the paths of the disks, press the **f** key.

Limitation:

The **-D** option provides Disk panel view where it reports disk service times, disk queuing metrics, and disk throughput. Whenever **-D** option is started, it resets the disk minimum and maximum service time metrics during the first interval. Because the service time metrics are reset during first interval of **-D** option, the existing instance of **-D** option or some other consumer's use of the disk service time metrics is affected.

-d hotdisk

Specifies the number of disks to be monitored. The *hotdisk* parameter specifies the number of the hot disks to be monitored. This is also the maximum number of disks displayed when enough room is available on the screen. When this number exceeds the number of disks installed, only the installed disks will be monitored and displayed. If this argument is omitted, a default of 2 is assumed. If a value of 0 (zero) is specified, no disk information is monitored.

Item	Description
-E	Displays the statistics of the shared Ethernet adapter on a Virtual I/O Server. The following metrics are displayed: KBPS The total throughput in kilobytes per second over the monitoring interval. This field is the sum of the kilobytes received and kilobytes sent per second. I-Pack The number of data packets received per second over the monitoring interval. O-Pack The number of data packets sent per second over the monitoring interval. KB-In The number of kilobytes received per second over the monitoring interval. KB-Out The number of kilobytes sent per second over the monitoring interval.
-F	Displays the file system display. When you specify the flag with the -@ flag or the @ subcommand, file system is shown in two windows. The top part of the display shows a list of active WPAR. This list can be sorted on any column. The display reports file system service times, file system queuing metrics, and file system throughput. The following metrics are reported: File System The name of file system. KBPS The amount of data transferred (read and written) per second over the monitoring interval. This field is the sum of the values of KB-Read and KB-Writ. TPS The number of transfers per second that are issued to the file system. A transfer is an I/O request to the file system. Multiple logical requests can be combined into a single I/O request to the file system. The size of a transfer is not determinate. KB-Read The amount of kilobytes read per second from the file system. KB-Writ The amount of kilobytes written per second from the file system. Open The logical number of files open. Create The logical number of files creates. Lock The number of files lock file system. Tip: If the file system name exceeds the field width in the display, then the file system name is displayed is truncated. The truncation contains the first and last few characters of the file system, and the middle part of the name is replaced by periods (...). For example, if the file system name is <code>filesystem001234</code>, then the file system name is displayed as <code>files...01234</code>.
-f <i>HotFS</i>	Specifies with the <i>HotFS</i> parameter the number of file system to be monitored. This is also the maximum number of file system displayed when enough room is available. When this number exceeds the number of file system mounted, only the mounted file system is monitored and displayed. If you do not specify the -f flag, the default value is two. If you specify a value of zero, the file system information is monitored.
-h	Displays help information in the following format: <pre>usage: topas [-d number-of-monitored-hot-disks] [-h] [-i monitoring-interval_in_seconds] [-n number-of-monitored-hot-network- interfaces] [-p number-of-monitored-hot-processes] [-w number-of-monitored-hot-WLM_classes] [-c number-of-monitored-hot-processors] [-U username_owned_processes] [-D -P -W -L] [-m]</pre>
-i <i>interval</i>	Sets the monitoring interval or the recording interval in seconds. If you specify the -i flag with the <i>interval</i> parameter, the <i>interval</i> parameter sets the monitoring intervals. The default value for the <i>interval</i> parameter is two seconds. If you specify the -i flag with the -R mode, the <i>interval</i> parameter becomes the recording interval for partition metrics. The default value for the <i>interval</i> parameter is 300 seconds. Valid values are 10, 15, 30, 60, 120, and 300 seconds.

Item	Description
-I <i>remotepollinterval</i>	For cross-partition display, sets with the <i>remotepollinterval</i> parameter the sampling interval to collect data from remote partitions. The default value for the <i>remotepollinterval</i> parameter is 10 seconds. Values of 10, 15, 30, 60 and 120 seconds are allowed.
-L	Displays the logical partition display. This display reports similar data to what is provided to mpstat and lparstat. In shared-memory mode, this panel displays information about I/O memory entitlement of the partition. The existing %lbusy, %hypv and hcalls metrics are replaced by the following metrics: IOME The I/O memory entitlement of the partition in gigabytes. iomu The I/O memory entitlement of the partition in use in gigabytes. pmem The physical memory that is backing logical memory of the partition in gigabytes. hpi The number of hypervisor page-ins. hpit The time in milliseconds waiting for hypervisor page-ins. With the -L flag specified, you can press the e key to display the I/O Memory Entitlement Pools panel. For more information about this panel, see I/O Memory Entitlement Pools Panel.

Item	Description
-M	Displays the Memory topology panel. The display reports similar data to what is provided by the lssrad command. There are two sections in this panel: - The first section gives us the memory topology from an SRAD point of view. Under every REF1 system detail level, it provides the individual SRAD IDs and the resources (memory, processors) associated with each of them. - The second section, the CPU RAD display, gives the relevant data at a processor level. The following metrics are displayed as part of this panel. REF1 The first hardware provided reference point, that identifies sets of resources that are near to each other. SRAD Scheduler Resource Allocation Domain ID. TOTALMEM The total memory in MB under the SRAD. INUSE The memory in use under the SRAD. FREE Free memory under the SRAD. FILECACHE The number of file cache bytes that are taken by the LRU daemon. HOMETHRDS The number of threads for which the SRAD is home. Threads typically run on the CPUs contained in the home SRAD, but it is not guaranteed. The system chooses a home SRAD for a thread when it is created. A thread's home SRAD may change during a thread's lifetime. CPUS The processors which are associated with this SRAD. 0 would indicate that cpu0 is associated with the corresponding SRAD id. 0 - 28 would indicate that all cpus from cpu0 to cpu28 are associated with the corresponding SRAD. If the cpu ids are not contiguous, then the values will be separated by commas. TOTALDISP Total number of threads dispatched from the corresponding processor during that interval. LOCALDISP% Percentage of threads that were dispatched locally within this SRAD, usually at the chip level. NEARDISP% Percentage of threads that were dispatched to a CPU that is not local, and that is not far. Typically, these may be resources that share the same hardware node. FARDISP% Percentage of threads that were dispatched to a processor typically outside the hardware node. Note: The hardware meanings for local, near and far vary with varying architectures.
-m	Displays in monochrome mode (no colors).
-n hotni	Specifies with the <i>hotni</i> parameter the number of hot network interfaces to be monitored. This is also the maximum number of network interfaces displayed when enough room is available on the screen. When this number exceeds the number of network interfaces installed, only the installed network interfaces will be monitored and displayed. If this argument is omitted, a default of value of 2 is assumed. If a value of 0 (zero) is specified, no network information is monitored.

Item	Description
-P	Similar to the ps command, the -P flag displays the full-screen process display. This display shows a list of the busiest processes, similar to the process subsection on the default display, only with more columns showing more metrics per process. This list can be sorted by any column. Following are the metrics displayed. USER The login name of the process owner. Truncates the user name to 8 characters. PID The process ID of the process. PPID The process ID of the parent process. PRI The priority of the process or kernel thread; higher numbers mean lower priority. NI The priority of a process specified with the nice command ; used in calculating priority for the sched other policy. DATA RES The real-memory data (resident set) size of the process (4 KB pages). TEXT RES The real-memory text (resident set) size of the process (4 KB pages). PAGE SPACE The virtual working set size used by process (4 KB pages). Note: The true paging space allocations per process are not available using the topas command. For more detailed reports, see the svmon command. TIME The total execution time for the process. CPU% The percentage of processor usage. PGFAULTS The number of I/O and other page faults. COMMAND The command name. Truncates the command name to 9 characters. When specified with -@ (topas -P -@), a new field WPAR is displayed and the PPID field is removed. All other metrics remains the same. WPAR The WPAR name that the process belongs to. Tip: If the WPAR class name exceeds 12 characters and it need to be displayed in a 12 character format, the first five characters will be followed by two periods (.), and then follows the last five characters. For example, if the WPAR class name is neptune001234, then the WPAR name is displayed as neptu . . 01234.
-photprocess	Specifies with the <i>hotprocess</i> parameter the number of hot processes to be monitored. This is also the maximum number of processes shown when enough room is available on the screen. If this argument is omitted, a default of 20 is assumed. If a value of 0 is specified, no process information will be monitored. Retrieval of process information constitutes the majority of the topas overhead. If process information is not required, always use this option to specify that you do not want process information.
-t	Toggles the tape display section on or off in the main topas display.

Item	Description
-T	Displays the full screen tape display panel. Note: Only the Atape device utilization is reported. The following metrics are displayed in this panel: Tape The name of the tape device. Busy% The bandwidth use of the tape. KBPS The amount of data transferred (read or written) to the tape in kilobytes per second. TPS The average number of transfers per second issued to the tape. KB-R The total number of kilobytes read from the tape. ART The average time to receive a response for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond. MRT The maximum time to receive a response for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond. KB-W The total number of kilobytes written to the adapter. AWT The average time to receive a response for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond. MWT The maximum time to receive a response for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.
-U <i>username</i>	With the -P flag, this flag shows the processes owned by the user specified with the <i>username</i> parameter. Only processes owned by the user that is specified will be shown in the All Process Display.
-V	Displays the Volume Group panel. The panel reports the following metrics of the volume groups in the top section of the panel, and the same metrics of the logical volumes in the bottom section of the panel. LogicalVolume/VolumeGroup The name of the logical volume or the volume group. TPS The total number of I/O requests over the interval that the metrics are displayed. KB-R The total number of kilobytes read over the interval. KB-W The total number of kilobytes written over the interval. KBPS The amount of data transferred (read or written) in kilobytes per second in the enquiring logical volume or volume group.
-W	Displays the full-screen WLM class display, which is a split display. The top part of the display shows a list of hot WLM classes, similar to the WLM classes subsection on the default display, but with enough space available to display the full class names. This list can be sorted on any column. If you specify the -@ flag, or if you press the @ subcommand, the WPAR section is displayed and the WLM section is not displayed. The WPAR section shows the list of hot WPAR. This list can be sorted on any column. The bottom part of the display shows a list of busiest processes, similar to the full screen process display, but only displays processes that belong to one WLM class or WPAR that are selected with the f key. Note: If the WLM class is not active then the default system processes will be displayed in the bottom part of the display.
-w [<i>number of monitored hot WLM classes</i>]	Specifies with the <i>hotwlmclass</i> parameter the number of hot Workload Manager (WLM) classes to be monitored. This is also the maximum number of WLM classes displayed when enough room is available on the screen. If this number exceeds the number of WLM classes installed, only the installed WLM classes will be monitored and displayed. If this argument is omitted, a default of 2 is assumed. If a value of 0 (zero) is specified, no WLM class information is monitored.

General Subcommands

While **topas** is running, it accepts 1-character subcommands. Each time the monitoring interval elapses, the program checks for one of the following subcommands and responds to the action requested.

Item	Description
a	Shows all of the variable subsections being monitored (processor, network, disk, WLM, and process). Pressing the a key always returns the topas command to the initial main display.
c	Replaces the current display of the cumulative report with the processor subsection. When you press the c key again, it displays the cumulative report. The number of busiest processors displayed will depend upon the space available on the screen.
C	Activates the Cross-Partition panel. If the panel is currently active, the C key resets the panel to display the global summary, dedicated, and shared sections. See the Additional Cross-partition Panel Subcommands section below for options specific to this panel.
d	Replaces the current display of the total disk activity with a list of the busiest disks. When you press the d key again, it displays the total disk activity. The number of busiest disks displayed will depend on the space available on the screen.
D	Replaces the current display with the Disk Metric display. This display offers additional information about disk access times and disk queuing. If the D key is pressed again, the display toggles back to the default main screen.
E	Shows the shared Ethernet adapter panel in VIO Server.
f	Press the f key while moving the cursor over a WLM class to display the list of top processes in the class at the bottom of the WLM screen. In the file system subsection of the topas command main panel, press the f key to replace the default report of total file system activity of the system with a list of busiest file system. When you press the f key again, it returns to the default display of the total file system activity. The number of busiest file system depends upon the space available on the screen. In the Volume Group panel (topas -V), you can select a volume group name and press the f key to display the list of top logical volumes that belong to the volume group at the bottom of the LVM panel.
F	Replaces the default display with the full-screen file system display. This display provides more detailed information about file systems on the system than the file system section of the main display. When the you press the F key again, it returns to the default main display.
G	Activates the Cluster Utilization panel. If the panel is currently active, the G key resets the panel to display the global summary, dedicated, and shared sections. See the Additional Cluster Utilization Panel Subcommands topic for options specific to this panel.
h	Shows the help screen.
H	Shows the help screen for the local panel, if available.
L	Replaces the current display with the logical partition display; LPAR, Micro-Partitioning, and simultaneous multithreading metrics similar to what lparstat and mpstat provide are displayed.
n	Replaces the report on the total network activity of the system with the list of the busiest interfaces. Press the n key in the network interfaces subsection. The number of busiest interfaces displayed will depend upon the space available on the screen.

Item	Description
p	Toggles the hot processes subsection on and off. The number of busiest processes displayed will depend upon the space available on the screen.
P	Replaces the default display with the full-screen process display. This display provides more detailed information about processes running on the system than the process section of the main display. When the P key is pressed again, it toggles to the default main display.
q	Quits the program.
r	Refreshes the display.
t	Toggles the tape display on or off in the main panel.
T	Shows the full-screen tape display.
V	Shows the Volume Group panel.
w	Toggles the Workload Manager (WLM) classes subsection on and off. The number of busiest WLM classes displayed will depend upon the space available on the screen.
W	Replaces the default display with the full-screen WLM class display. This display provides more detailed information about WLM classes, WPAR classes, and processes assigned to classes. When you press the @ key, the WLM class subsection is replaced by WPAR subsection. When you press the W key again, it toggles back to the default main display.
@	Toggles between the WLM class metric and WPAR metrics, that is, WPAR is monitored instead of WLM. This is the at (@) key. This key is valid for the Main panel, Process panel, File System panel, and WLM panel. If you press the @ key from any other panel, it is ignored. The @ key is restricted inside a WPAR, that is, it is ignore inside a WPAR. The @ key is valid in the following panels: Main Panel The WLM and Process subsections are replaced by the WPAR metric. Process Panel The default mode of the process panel is replaced by the WPAR mode. File System Panel The file system panel contains WPAR names if you press the f key. The per WPAR file-system metrics are displayed on the lower section of this panel. WLM Panel The WLM subsection is replaced by the WPAR subsection.
Arrow and Tab keys	Subsections from the main display such as the processor, Network, Disk, WLM Classes, and the full-screen WLM and Process displays can be sorted by different criteria. Positioning the cursor over a column activates sorting on that column. The entries are always sorted from highest to lowest value. The cursor can be moved by using the Tab key or the arrow keys. Sorting is only valid for 128 disks and 16 network adapters.
~	Shows the nmon screen. This is the tilde (~) key.

Additional Cross-Partition Panel Subcommands

When the topas Cross-partition panel is active, it accepts the following additional 1-character subcommands. Each time the monitoring interval elapses, the program checks for one of the following subcommands and responds to any requested action.

Item	Description
d	Toggles the dedicated partition section on and off.
g	Toggles the top global section of the panel between brief listing, detailed listing, and off.
r	Forces topas to search the for HMC configuration changes if a connection is available. This includes the discovery of new partitions, processors, or memory allocations.
s	Toggles the shared partition section on and off.
p	Toggles the pool panel section on or off. Inside the pool panel, user can select one pool ID and press the f key to list the shared partitions that belong to the pool.
v	Toggles the Virtual I/O Server/Client Throughput details on or off. You can select one virtual I/O server and press the f key to list the VIO clients that belong to that server.
m	Toggles the memory pool panel on or off. You can select a memory pool and press the f key to view the partitions in that pool.

Additional Cluster Utilization Panel Subcommands

When the topas Cluster Utilization panel is active, it accepts the following additional 1-character subcommands. Each time the monitoring interval elapses, the program checks for one of the following subcommands and responds to any requested action:

Item	Description
d	Toggles the dedicated partition section on and off.
g	Toggles the top global section of the panel between brief listing, detailed listing, and off.
s	Toggles the shared partition section on and off.

Additional Disk Panel (topas -D) Subcommands

When the topas Disk panel is active, it accepts the following additional 1-character subcommands. Each time the monitoring interval elapses, the program checks for one of the following subcommands and responds to any requested action:

Item	Description
d	Toggles the Adapter panel on or off.
m	Toggles the MPIO panel on or off.

Additional Adapter Panel Subcommands

When the topas Adapter panel is active, it accepts the following additional 1-character subcommand. Each time the monitoring interval elapses, the program checks for one of the following subcommands and responds to any requested action:

Item	Description
v	Toggles the Virtual Adapter panel on or off. Press this key from the Adapter panel.

Additional Logical Partition Panel (topas -L) Subcommands

When the topas Logical panel is active, it accepts the following additional 1-character subcommand. Each time the monitoring interval elapses, the program checks for one of the following subcommands and responds to any requested action:

Item	Description
e	Toggles the I/O Memory Entitlement Pools panel.

Additional Virtual I/O Server/Client Throughput Panel Subcommands

When the topas Virtual I/O Server/Client Throughput panel is active, it accepts the following additional 1-character subcommand. Each time the monitoring interval elapses, the program checks for one of the following subcommands and responds to any requested action:

Item	Description
d	Turns the Virtual I/O Server/Client Disk panel on or off for the Virtual I/O Server that is selected in the Virtual I/O Server/Client Throughput panel. You can select the server adapters and press the f key to list the disks and the clients that belong to that adapter.

Sample Full-Screen Workload Manager Classes Output

The following is an example of the display generated by the **topas -W** command:

```
Topas Monitor for host:      ptoolsl3      Interval:      2      Mon Feb 12 06:25:11 2007
WLM-Class (Active)          CPU%          Mem%          Blk-I/O%
System                      0             57             0
Shared                      0             4              0
Default                     0             0              0
Unmanaged                   0            14             0
Unclassified                0            38             0

=====
USER      PID    PPID  PRI  NI   DATA  TEXT  PAGE  TIME  CPU%  PGFAULTS
root       1      0  108  20   197     9   180   0:24  0.0    I/O  OTH  COMMAND
root    1032      0   16  41     3  3374     3   0:00  0.0    0    0  init
root    1290      0   60  41     4  3374     4   0:02  0.0    0    0  lrud
root    1548      0   36  41     4  3374     4   0:26  0.0    0    0  xmgc
root    1806      0   37  41    16  3374    16  13:25  0.0    0    0  netm
root    2064      0   16  41     4  3374     4   0:04  0.0    0    0  gil
root    2698      1  108  20    14     2    14   0:00  0.0    0    0  wlmshed
root    3144      1  108  20    40     1    36   5:19  0.0    0    0  shlap
root    3362      0  108  20     4  3374     4   0:00  0.0    0    0  syncd
root    3666      1  108  20   135    23   123   0:00  0.0    0    0  lvmbb
root    3982      0  108  20     4  3374     4   0:01  0.0    0    0  errdemon
root      0      0   00  00     0     0     0   0:00  0.0    0    0  rtcmd
```

The following is an example of the display generated by **topas -W -@** command:

```
Topas Monitor for host:      ptoolsl3      Interval:      2      Mon Feb 12 06:25:11 2007
WPAR          CPU%          Mem%          Blk-I/O%
neptune001234  0             1             0

=====
USER      PID    PPID  PRI  NI   DATA  TEXT  PAGE  TIME  CPU%  PGFAULTS
root    356372  491650  58  41   370     67   370   0:00  0.1    I/O  OTH  COMMAND
root    262246  188508  24  41   256     21   256   6:27  0.1    0    0  topas
root    192626      1   60  20   113     17   113  11:17  0.1    0    0  xmtopas
root     61470      0   16  41    17      0    17   0:31  0.0    0    0  getty
root    290818      1   58  41   284     67   284   1:54  0.0    0    1  wlmshed
root      0      0   00  00     0     0     0   0:00  0.0    0    0  topas
```

root	57372	0	37	41	30	0	30	3:39	0.0	0	0	gil
root	86248	1	60	20	47	0	47	1:04	0.0	0	0	rpc.lock
root	385224	237728	60	20	254	197	254	0:00	0.0	0	0	sendmail
root	131174	176242	60	20	175	79	175	0:03	0.0	0	0	aixmibd
root	53274	0	36	41	13	0	13	0:05	0.0	0	0	netm
root	90244	1	60	20	126	2	126	2:35	0.0	0	0	syncd
root	45078	0	60	41	14	0	14	0:58	0.0	0	0	xmgc
root	266384	176242	60	20	644	160	644	0:27	0.0	0	0	IBM.CSMA
root	250004	176242	60	20	617	157	617	0:26	0.0	0	0	rmcd
root	184410	176242	60	20	254	197	254	0:14	0.0	0	0	sendmail
root	151640	0	60	20	13	0	13	0:02	0.0	0	0	rgsr
root	40980	0	59	41	71	0	71	0:02	0.0	0	0	pilegc
root	110738	0	60	20	13	0	13	0:01	0.0	0	0	n4bg
root	180368	1	60	20	98	14	98	0:01	0.0	0	0	cron
root	1	0	60	20	158	10	158	0:01	0.0	0	0	init

Examples

1. To display up to twenty "hot" disks every five seconds and omit network interface, WLM classes, file system information and process information, enter the following command:

```
topas -i5 -n0 -p0 -w0 -f0
```

2. To display the five most active processes and up to twenty most active WLM classes (which is the default when omitting the **-w** flag) but no network , disk, or file system information, enter the following command:

```
topas -p5 -n0 -d0 -f0
```

3. To run the program with default options, enter the following command:

```
topas
```

4. To go directly to the process display, enter the following command:

```
topas -P
```

5. To go directly to the WLM classes display, enter the following command:

```
topas -W
```

6. To go directly to the logical partition display, enter the following command:

```
topas -L
```

7. To go directly to the disk metric display, enter the following command:

```
topas -D
```

8. To go directly to the file system display, enter the following command:

```
topas -F
```

9. To go directly to WPAR monitoring mode *abc*, enter the following command:

```
topas -@ abc
```

10. To go directly to the **topas** WPAR mode, enter the following command:

```
topas -@
```

11. To go directly to the LVM display, enter the following command:

```
topas -V
```

12. To go directly to the tape display, enter the following command:

```
topas -T
```

13. To go to the shared Ethernet adapter on the VIO Server panel, enter the following command:

```
topas -E
```

14. To go directly to the cluster utilization display, enter the following command:

```
topas -G
```

15. To go directly to the Memory topology panel and view SRAD statistics, enter the following command:

```
topas -M
```

16. To display the process utilization specific to the user **guest**, enter the following command:

```
topas -P -U guest
```

17. To display top two processors with high processor utilization, enter the following command:

```
topas -c2
```

Files

Item	Description
<code>/usr/bin/topas</code>	Contains the topas command.

topasout Command

Purpose

Generates reports by processing **xmwl**m, **nmon**, and **topas** recordings.

Syntax

Local reports

topasout -R *type* [**-i** *interval*] [**-b** *time*] [**-e** *time*] *topas_recording_file*

Comma-separated report

topasout -c [**-m** *type*] *topas_recording_file*

Spread-sheet report

topasout [-s] [**-m** *type*] *topas_recording_file*

Nmon analyzer report

topasout -a *topas_recording_file*

WLE Report from topasrec / nmon file

topasout -R *wle* { *nmon_recording_file* | *topas_recording_file* }

CEC reports

topasout -R *type* [**-i** *interval*] [**-b** *time*] [**-e** *time*] *topas_recording_file*

Comma-separated report

topasout [-c] *topas_recording_file*

Spread-sheet report

topasout -s *topas_recording_file*

Description

The **topasout** command is used to convert the binary recordings generated by the **xmwlm**, **xmtrend**, or **topasrec** utilities. The binary recording can be the local system recording, the central electronic complex (CEC) recording, or the cluster recording. Through SMIT, you can enable, configure, or disable a binary recording.

If there is more than one value for a metric within the user-specified interval, the **topasout** command averages out all of the values to get single value that can be printed in the report. For values that cannot be averaged out (like simultaneous multithreading, dedicated and shared modes), the **topasout** command takes the last or the first values that are recorded in the interval.

Local reports

There are several types of local reports: the Summary report, the Detailed report, the LAN report, the Disk report, the Comma-separated report, the Nmon analyzer report, the Adapter report, and the Virtual adapter report.

Summary report

A Summary report presents the consolidated view of system information.

The following column headings are in a summary report:

Item	Description
Time	Ending time of the report interval. Metric values are averaged out over this interval and printed in the report
InU	Memory that is used
Us	Percentage of processor time spent in the user mode
Sy	Percentage of processor time spent in the system mode
Wa	Percentage of processor time spent waiting for I/O
Id	Percentage of time that the processor is idle
PhysB	Percentage of physical processors that are busy
RunQ	The average number of threads that are ready to run but are waiting for a processor to become available
WtQ	The average number of threads that are waiting for paging to be completed
Cswitch	The number of context switches per second in the reporting interval
Syscall	The number of system calls executed per second in the reporting interval
PgFault	The number of I/O and other page faults
%don	Sum of %idle cycles donated and %busy cycles donated
%stl	Sum of %idle cycles stolen and %busy cycles stolen

The following sample shows the output of a local Summary report:

```
Report: System Summary --- hostname: aixfvt19          version:1.1
Start:01/24/07 04:45:50  Stop:01/24/07 04:48:07  Int: 5 Min  Range: 2 Min
Mem: 1.2 GB Dedicated SMT: ON Logical CPUs: 2
Time      InU Us Sy Wa Id PhysB RunQ WtQ CSwitch Syscall PgFault
04:48:07  1.2 3 0 0 88 3.43 1.1 0.0 168 893 23
```

Detailed report

A detailed report provides a detailed view of the system metrics.

The following column headings are in a detailed report:

Item	Description
Mode	The information about the following modes are reported: • Don represents donating dedicated partition. • Ded represents that the dedicated partition are not donating or the donation is not enabled. • Shr represents shared mode.
Lp	Number of logical processors.
SMT	Status of the SMIT. It is On when the SMT is enable. It is Off when the SMT is disabled.
Ent	Entitlement granted (shared-only).
Poolid	Pool ID. This column is applicable only if this partition belongs to a valid shared processor pool.
Kern	Percentage of processor time spent in the kernel mode.
User	Percentage of processor time spent in the user mode.
Wait	Percentage of processor time spent for waiting for I/O.
Idle	Percentage of time that the processor is idle.
PhysB	Percentage of physical processors that are busy.
Entc	Percentage of entitled capacity that is consumed. This heading is applicable for shared partition only.
Sz, GB (in Memory section)	Memory size in gigabytes.
InU (in Memory section)	Memory used in gigabytes.
%Comp	Percentage of real memory that is allocated to computational page frames. Computational page frames are backed by paging space.
%Nonc	Percentage of real memory that is allocated to non-computational page frames. Non-computational page frames are backed by file space: either data files, executable files, or shared library files.
%Clnt	Percentage of real memory that is allocated to cache, remotely mounted files.
Sz, GB (in Paging section)	Paging space in gigabytes.
InU (in Paging section)	Paging space used in gigabytes.
Flt	Total number of page faults that are taken per second in the reporting interval. This includes page faults that do not cause paging activity.
Pg-I	Number of 4 K pages that are read per second in the reporting interval.
Pg-O	Number of 4 K pages that are written per second in the reporting interval.
Bdon	Percentage of physical processor that is used while busy cycles are being donated. This metric is applicable only for donating dedicated partitions.
Idon	Percentage of physical processor that is used while explicitly donating idle cycles. This metric is applicable only for donating dedicated partitions.

Item	Description
Istl	Percentage of physical processor that is used while idle cycles are being stolen by the hypervisor. This metric is applicable only for dedicated partitions.
Bstl	Percentage of physical processor that is used while busy cycles are being stolen by the hypervisor. This metric is applicable only for dedicated partitions. The %idon and %bdon metrics are not displayed when no dedicated partition is donating.
Vcsw	Average number of virtual context switches per second in the reporting interval.
Phint	Average number of phantom interrupts per second in the reporting interval. This column is applicable only to shared partitions.
Cswth	Number of process context switches per second in the reporting interval.
Syscl	Number of system calls per second run in the reporting interval.
RunQ	Average number of threads that are ready to run but are waiting for a processor to become available.
WtQ	Average number of threads that are waiting for paging to complete.
SrvV2	Number of NFS Server V2 calls per second in the reporting interval.
ClV2	Number of NFS Client V2 calls per second in the reporting interval.
SrvV3	Number of Server V3 calls per second in the reporting interval.
ClV3	Number of Client V3 calls per second in the reporting interval.
Network	Name of the network interface.
I-Pack	Number of data packets that are received per second.
O-Pack	Number of data packets that are sent per second in the reporting interval.
KB-I	Number of kilobytes that are received per second in the reporting interval.
KB-O	Number of kilobytes that are sent per second in the reporting interval.
Disk	Name of the physical disk.
Busy%	Percentage of time that the physical disks are active (bandwidth utilization for the drive).
KBPS	Number of kilobytes that are read and written per second in the reporting interval. This column is the sum of the KB-R and KB-W metrics.
TPS	Number of transfers per second that are issued to the physical disk. A transfer is an I/O request to the physical disk. Multiple logical requests can be combined into a single I/O request to the disk. The size of a transfer is not determinate.
KB-R	Number of kilobytes that are read per second from the physical disk in the reporting interval.
KB-W	Number of kilobytes that are written per second to the physical disk in the reporting interval.

The following sample shows a local Detailed report:

```

Sample output
#Report: System Detailed --- hostname: ptoolsl1 version: 1.2
Start:12/21/05 10.00.00 Stop:12/21/05 11.00.00 Int: 5 Min Range: 60 Min
Time: 10.00.00 -----
CONFIG      CPU      MEMORY      PAGING
Mode Don    Kern    12.0      Sz,GB 16.0      Sz,GB 4.0
LP 4       User    8.0      InU    4.3      InU    2.3
SMT ON      Wait    0.0      %Comp  3.1      Flt    221
Ent 3.0     Idle    80.0     %NonC  9.0      Pg-I   87
Poolid 3     PhyB    0.7      %CInt  2.0      Pg-O   44
              EntC    8.0

PHYP      EVENTS/QUEUES  NFS
Bdon 0.1 Cswth 3213 SrvV2 32
Idon 0.5 Syscl 43831 CltV2 12
Bstl 0.5 RunQ 1 SrvV3 44
Istl 0.4 WtQ 0 CltV3 18
Vcsw 1214
Phint 120

Network KBPS I-Pack O-Pack KB-I KB-O
en0 0.6 7.5 0.5 0.3 0.3
en1 22.3 820.1 124.3 410.0 61.2
lo0 0.0 0.0 0.0 0.0 0.0

Disk Busy% KBPS TPS KB-R KB-W
hdisk0 0.0 0.0 0.0 0.0 0.0
hdisk1 0.0 0.0 0.0 0.0 0.0

topasout local report - detailed report

```

Disk reports

A Disk report provides information about the amount of data that are read or written to disks.

The following column headings are in a Disk report:

Item	Description
Mem	Total memory that are available in gigabytes at the first reporting interval.
Logical CPUs	Number of logical processors at the first reporting interval.
Time	Ending time of the reporting interval. Metric values are averaged out over this time interval and printed in the report.
InU	Total memory that are used in gigabytes.
PhysB	Percentage of physical processors that are busy.
MBPS	Number of megabytes that are read and written per second. This column is the sum of the MB-W and MB-R metrics.
TPS	Number of transfers per second that are issued to the physical disk. A transfer is an I/O request to the physical disk. Multiple logical requests can be combined into a single I/O request to the disk. The size of a transfer is not fixed.
MB-R	Data that are read in megabytes per second from the physical disk.
MB-W	Data that are written in megabytes per second to the physical disk.

The following sample shows the output of a local Disk report:

```

Sample output
Report: Total Disk I/O Summary --- hostname: aixfvt19 version:1.1
Start:01/24/07 04:45:50 Stop:01/24/07 04:48:07 Int: 5 Min Range:15 Min
Mem: 1.2 GB Dedicated SMT: ON Logical CPUs: 2
Time InU PhysB MBPS TPS MB-R MB-W
04:48:07 1.2 3.4 0.2 2.1 0.1 0.1
04:53:07 1.2 3.4 0.3 2.1 0.0 0.3
...

```

LAN reports

A LAN report provides the amount of data that are received or sent in the network interfaces.

The following column headings are in a LAN report:

Item	Description
Mem	Total memory available in gigabytes at the first reporting interval.
Logical processors	Number of logical processors at the first reporting interval.
Time	Ending time of the reporting interval. Metric values are averaged out over this time interval and printed in the report.
InU	Total memory used in gigabytes.
PhysB	Percentage of physical processors that are busy.
MBPS	Sum of the MB-I and MB-O values. It equals to the data in megabytes sent and received per second.
MB-I	Data in megabytes that are received per second in the reporting interval.
MB-O	Data in megabytes that are sent per second in the reporting interval.
Xmtdrp	Average amount of transmitted packets that are dropped per second at device driver level in the reporting interval.
Rcvdrp	Average amount of received packets that are dropped per second at device driver level in the reporting interval.

The following sample shows the output of a local LAN report:

```
#Report: System LAN Summary --- hostname: tooltime2          version:1.1
Start:03/02/07 00:38:18 Stop:03/02/07 07:08:32   Int:  5 Min   Range:  390 Min
Mem:  4.0 GB  Shared SMT: ON  Logical CPUs:  2
Time      InU    PhysB  MBPS  MB-I  MB-O  Rcvdrp  Xmtdrp
00:43:18  0.6    0.1    0.0   0.0   0.0    0        0
00:48:18  0.6    0.3    0.0   0.0   0.0    0        0
00:53:19  0.7    0.2    0.0   0.0   0.0    0        0
...
```

Nmon analyzer style output

The **topasout** command generates a Nmon analyzer report that can be viewed with the **nmon** analyzer.

The **topasout** command is used to post process the binary recordings generated by the **xmwlm** utility the **xmtrend** utility and the **topasrec** utility. The binary recording can be the Local System recording, Central Electronic Complex (CEC) recording or Cluster recording. Through SMIT you can enable, configure or disable a binary recording.

Note: The **xmwlm** and **xmtrend** utilities are obsolete and are replaced by the **topasrec** utility.

Use the **topasout** command with the **-a** flag to generate this report. You can open the generated **.csv** file with a **nmon** analyzer. For example, to generate a **xmwlm.061016.csv** file, enter the following command:

```
topasout -a /etc/perf/daily/xmwlm.061016
```

The generated **.csv** file locates in the same directory of the original file, that is, in the **/etc/perf/daily/** directory. The file name is **xmwlm.061016.csv**.

Comma-separated report

The **topasout** command generates a report that contains data that is separated with comma.

Use the **topasout** command with the **-c** flag to generate this report. The output file is written to **recordedfilename_01** file.

For example, to generate a comma-separated report for the **xmwlm.060503** file, enter the following command:

```
topasout -c /etc/perf/daily/xmwlm.060503
```

The output file is the **xmwlm.060503_01** file which locates in the same directory as the original file.

When you specify the **-m** flag, the **topasout** command writes the *min*, *max*, *mean*, *stdev*, and the *exp* values of the recorded metrics in the report.

The following sample shows the output of a local report with the data separated by commas:

```
#Monitor: xmtrend recording-- hostname: aixfvt19 ValueType: mean
Time="2007/01/24 04:45:50", CPU/gluser=0.02
Time="2007/01/24 04:45:50", CPU/glkern=0.28
Time="2007/01/24 04:45:50", CPU/glwait=0.00
Time="2007/01/24 04:45:50", CPU/glidle=99.69
Time="2007/01/24 04:45:50", NFS/Server/v3calls=0.00
Time="2007/01/24 04:45:50", NFS/Server/v2calls=0.00
...
```

Spreadsheet format report

The **topasout** command generates a report in spreadsheet format.

Use the **topasout** command with **-s** flag to generate this report. The output file is written to *recordedfilename_01* file.

For example, to generate a report in spreadsheet format for the **xmwlm.060503** file, enter the following command:

```
topasout -s /etc/perf/daily/xmwlm.060503
```

The output file is the **xmwlm.060503_01** file which locates in the same directory as the original file.

When you specify the **-m** flag, the **topasout** command writes the *min*, *max*, *mean*, *stdev*, and the *exp* values of the recorded metrics in the report.

Adapter report

An Adapter report provides information about the amount of data that is read or written to adapters.

The following metrics of the adapter are in the report:

Item	Description
Adapter	Name of the adapter
KBPS	Amount of data transferred (read or written) in the adapter in kilobytes per second
TPS	Number of transfers per second that are issued to the adapter
KB-R	Number of kilobytes read from the adapter
KB-W	Number of kilobytes written to the adapter

The following metrics of the disks are in the report:

Item	Description
Vtargets/Disk	Name of the virtual target device or disk.
Busy%	Percentage of time that the virtual target device or disk is active (bandwidth use for the drive).
KBPS	Number of kilobytes read and written per second over the monitoring interval. This field is the sum of the values of the KB-R and KB-W metrics.

Item	Description
TPS	Number of transfers per second that are issued to the virtual target device or disk. A transfer is an I/O request to the virtual target device or disk. Multiple logical requests can be combined into a single I/O request to the disk. A transfer is of medium size.
KB-R	Number of kilobytes read per second from the virtual target device or disk.
KB-W	Number of kilobytes written per second to the virtual target device or disk.
AQD	Average number of requests waiting to be sent to the virtual target device or disk.
AQW	Average queue that is waiting per request reported in millisecond. The suffix indicates the unit of time. The default time unit is millisecond.
ART	Average time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.
AWT	Average time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.
MRT	Maximum time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.
MWT	Maximum time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.

Virtual adapter report

The following metrics of the adapter are reported in the Virtual adapter report:

Item	Description
vAdapter	Name of the adapter.
KBPS	Amount of data transferred (read or written) in the adapter in kilobytes per second.
TPS	Number of transfers per second that are issued to the adapter.
KB-R	Number of blocks received per second from the hosting server to this adapter.
KB-W	Number of blocks sent per second from this adapter to the hosting server.
AQD	Number of requests waiting to be sent to adapter.
AQW	Time spent by a transfer request in the wait queue. Reported in millisecond. The suffix indicates the unit of time. The default time unit is millisecond.
ART	Time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.
AWT	Time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.
MRT	Maximum time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.

Item	Description
MWT	Maximum time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.

The following metrics of the disks are in the report:

Item	Description
Vtargets/Disk	Name of the virtual target device or disk.
Busy%	Percentage of time that the virtual target device or disk is active (bandwidth use for the drive).
KBPS	Number of kilobytes read and written per second over the monitoring interval. This field is the sum of the values of the KB-R and KB-W metrics.
TPS	Number of transfers per second that are issued to the virtual target device or disk. A transfer is an I/O request to the virtual target device or disk. Multiple logical requests can be combined into a single I/O request to the disk. A transfer is of medium size.
KB-R	Number of kilobytes read per second from the virtual target device or disk.
KB-W	Number of kilobytes written per second to the virtual target device or disk.
AQD	Average number of requests waiting to be sent to the virtual target device or disk.
AQW	Average queue that is waiting per request reported in milliseconds. The suffix indicates the unit of time. The default time unit is millisecond.
ART	Average time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.
AWT	Average time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.
MRT	Maximum time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.
MWT	Maximum time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.

On Demand WLE input from topasrec / nmon recordings

In addition to weekly peak inputs for WLE through SMIT, the user can invoke an On Demand WLE input file to study a particular workload and use that data to size the systems and generate reports. The **topasout** command has the capability to study a particular **topas** or **nmon** recording and generate WLE readable reports in xml format using this option.

Use topasout -R wle -Oifile=<filename> option to generate the WLE report. For example, to generate a report from file, use the following command.

```
topasout -R wle -Oifile=/etc/perf/daily/xmwl1m_130504.topas
```

If it is **nmon** recording, specify the **-Otype** option along with the **-Oifile** option as shown below:

```
topasout -R wle -Oifile=/etc/perf/daily/xmwl1m_130504.nmon -Otype=nmon
```

The *wle* option is different from the other types of **-R** in a way that both **topas** and **nmon** recordings can be given as an input to this option while only **topas** recordings (recordings generated through **xmwl** and **topasrec**) can be given as an input file for the other options.

CEC reports

There are five types of CEC reports: the Summary report, the Detailed report, the Shared processor pool report, the Comma-separated report, and the Spread-sheet report.

Summary report

This report provides a summary of the CEC system. The reporting is based on the partitions that actually responded to the **topas** command. If the partitions in the CEC do not have the **xmtopas** or **xmservd** configured, the partitions cannot be monitored.

A CEC summary report contains the following column headings:

Header (partition details):

Item	Description
Mon	Number of the partitions that are monitored in the first reporting time interval
UnM	Number of the partitions that are not monitored in the first reporting time interval
Shr	Number of the shared partitions in the first reporting time interval
Ded	Number of the dedicated partitions in the first reporting time interval
Cap	Number of the capped partitions in the first reporting time interval
UnC	Number of the uncapped partitions in the first reporting time interval

CEC:

Item	Description
ShrB	Shared physical processor busy. (Sum of physical busy of processors in the shared partitions.)
DedB	Dedicated physical processor busy. (Sum of physical busy of processors in the dedicated partitions.)
Don	Total number of the processors that are donated to the physical pool.

Processors:

Item	Description
Mon	Number of the physical processors that are monitored
UnMon	Number of the physical processors that are not monitored.
Shr	Number of the processors in shared partitions
Ded	Number of the processors in dedicated partitions
PSz	Number of the active shared processors in the physical pool
APP	Available physical processors in the pool

Memory (GB):

Item	Description
Mon	Total memory of the monitored partitions
UnM	Total memory of the partitions that are not monitored
Avl	Memory available to partitions

Item	Description
InUse	Memory in used in the monitored partitions
UnA	Memory that is not available for partitions

The following sample shows the output of a CEC Summary report:

```

Sample Output
#Report: CEC Summary --- hostname: ptools13 version:1.2
Start:02/22/07 00:44:06 Stop:02/22/07 23:59:06 Int: 5 Min Range:1395 Min
Partition Mon: 3 UnM: 0 Shr: 1 Ded: 2 Cap: 2 UnC: 1
-CEC----- -Processors----- -Memory (GB)-----
Time ShrB DedB Don Stl Mon UnM Shr Ded PSz APP Mon UnM Avl UnA InU
00:49 0.00 0.00 - - 2.2 0.0 0.2 2 2.0 2.0 9.4 0.0 8.0 0.0 1.0
00:54 0.00 0.00 - - 2.2 0.0 0.2 2 2.0 2.0 9.4 0.0 8.0 0.0 1.0
00:59 0.00 0.00 - - 2.2 0.0 0.2 2 2.0 2.0 9.4 0.0 8.0 0.0 1.0

```

Detailed report

A CEC Detailed report gives a detailed view of all the partitions that the **topas** command is able to record data from.

The followings column headings are in a CEC Detailed report:

Partition Info:

Item	Description
Monitored	Number of partitions that are monitored
Unmonitored	Number of partitions that are not monitored
Shared	Number of shared partitions
Uncapped	Number of uncapped shared partitions
Capped	Number of capped shared partitions
Dedicated	Number of dedicated partitions
Donating	Number of partitions that are donating

Memory:

Item	Description
Monitored	Total memory that is monitored
UnMonitored	Total memory that is not monitored
Available	Total memory that is available
UnAllocated	Total memory that is not allocated to any partition
Consumed	Total memory that is consumed by the partitions

Processor:

Item	Description
Monitored	Number of physical processors that are monitored
UnMonitored	Number of physical processors that are not monitored
Available	Number of physical processors that are available in the CEC system
UnAllocated	Number of physical processors that are not allocated to any partition
Shared	Number of processors in shared partitions

Item	Description
Dedicated	Number of processors in dedicated partitions
Donated	Sum of the number of processors in all of the partitions that are currently donating
Pool Size	Number of active shared processors in the physical pool
Avail Proc Pool	Available physical processors in pool. This is the idle cycles in the pool reported as a number of processors
Shr Physical Busy	Sum of the busy physical processors of all of the shared partitions
Ded Physical CPUs	Sum of the busy physical processors of all of the dedicated partitions
Donated Phys. CPUs	Sum of the donated processor cycles (reported as a number of processors) from all partitions
Stolen Phys. CPUs	Sum of the stolen processor cycles (reported as a number of processors) from all partitions
Virtual Pools	Number of the virtual pools
Virt. Context Switch	Total number of the virtual context switches per second in the monitoring interval
Phantom Interrupts	Total number of the phantom interrupts per second in the monitoring interval

Individual partition data:

Item	Description
Host	Host name
OS	Operating system level
M	The M column heading represents the mode. In shared partitions, it displays the following attributes: • C- SMT is enabled and capped • c- SMT is disabled and capped • U- SMT is enabled and uncapped • u- SMT is disabled and uncapped In dedicated partitions, it displays the following attributes: • S- SMT is enabled and is not donating • d- SMT is disabled and donating • D- SMT is enabled and donating
Mem	Total memory in gigabytes
InU	Memory in used in gigabytes
Lp	Number of logical processors
Us	Percentage of processor that is used by programs executing in the user mode
Sy	Percentage of processor that is used by programs executing in kernel mode
Wa	Percentage of time that is spent waiting for I/O
Id	Percentage of time that the processor is idle

Item	Description
PhysB	Number of physical processors that are busy
Ent	Entitlement granted (shared only)
%Entc	Percentage of entitlement consumed (shared only)
Vcsw	Virtual context switches average per second (shared only)
PhI	Phantom interrupts average per second (shared only)
%idon	Percentage of physical processor that is used while explicitly donating idle cycles. This metric is applicable only for donating dedicated partitions.
%bdon	Percentage of physical processor that is used while busy cycles are being donated. This metric is applicable only for donating dedicated partitions
%istl	Percentage of physical processor that is used while busy cycles are being stolen by the hypervisor. This metric is applicable only for dedicated partitions

The following sample shows the output of a CEC Detailed report:

```
#Report: CEC Detailed --- hostname: ptoolsl3 version:1.2
Start:03/06/07 07:19:39 Stop:03/06/07 07:28:39 Int: 5 Min Range: 9 Min

Time: 07:24:38 -----
Partition Info  Memory (GB)  Processors  Avail Pool : 2.0
Monitored : 3 Monitored : 9.4 Monitored : 2.2 Shr Physcl Busy: 0.01
UnMonitored: 0 UnMonitored: 0.0 UnMonitored: 0.0 Ded Physcl Busy: 0.01
Shared : 1 Available : 0.0 Available : 0.0 Donated Phys. CPUs: 0.00
UnCapped : 1 UnAllocated: 0.0 Unallocated: 0.0 Stolen Phys. CPUs : 0.00
Capped : 2 Consumed : 0.0 Shared : 0.2 Hypervisor
Dedicated : 2 Dedicated : 2.0 Virt Cntxt Swtch: 545
Donating : 0 Donated : 0 Phantom Intrpt : 0
Pool Size : 2.0

Host OS M Mem InU Lp Us Sy Wa Id PhysB Vcsw Ent %EntC PhI
-----shared-----
ptoolsl1 A53 U 3.1 1.9 4 0 1 0 98 0.01 317 0.2 2.55 0

Host OS M Mem InU Lp Us Sy Wa Id PhysB Vcsw %istl %bstl
-----dedicated-----
ptoolsl3 A54 3.1 0.9 2 0 0 0 99 0.00 228 - -
ptoolsl1 A52 3.1 2.7 1 0 1 0 99 0.01 0 - -

Time: 07:28:39 -----
```

Shared-processor-pool report

The CEC Shared-processor-pool report contains information about the shared processor pools.

The following column headings are included in a Shared-processor-pool report:

Item	Description
psize	Effective maximum capacity of the pool.
entc	Entitled capacity of the pool.
maxc	Maximum capacity of the pool.
physb	Sum of the physical busy of processors in the shared partitions of a pool. (The "physical busy" refers to fraction of physical processors that are busy.)
app	Available physical processors in the pool.
mem	Sum of the monitored memory for all of the shared partitions in the pool.

Item	Description
muse	Sum of the memory consumed by all of the shared partitions in the pool.

The following sample shows the output of a CEC Shared-processor-pool report:

Sample Output

```
#Report: Topas CEC Pool Detailed --- hostname: ptoolsl1          version: 1.0
pool  psize entc  maxc physb app  mem  muse
0      3.0  2.0   3.0 0.1  1.0  2.0  1.0
1      4.0  3.0   5.0 0.5  1.5  1.0  0.5
2      3.0  2.5   4.0 0.2  2.0  1.0  0.5

Host      Pi OS  M Mem InU Lp  Us Sy Wa Id  PhysB Vcsw Ent  %EntC PhI
-----shared-----
ptools1    0 53 U  11   9  2  11 13  0 75  0.10  121 0.25  0.3  3
ptools5    1 53 U  12  10  2  12  3  0 85  0.20  121 0.25  0.3  3
ptools3    1 53 C  5.0 2.6  2  10  1  0 89  0.15   52 0.25  0.3  2
ptools7    2 53 c  2.0 0.4  1   0  1  0 99  0.05   2 0.10  0.3  2

Host      OS  M Mem InU Lp  Us Sy Wa Id  PhysB Vcsw %istl %bstl %bdon %idon
-----dedicated-----
ptools6    52  1.1 0.1  1  11  7  0 82  0.50   50  10    5  10  0
ptools8    52  1.1 0.1  1  11  7  0 82  0.50   60   0    1  -  -
ptools2    52  1.1 0.1  1  11  7  0 82  0.50  200   0   15 25 10
```

Memory pool report

The **topasout** command generates the Memory pool report that contains information about the memory pools in the CEC and the partitions that belong to the memory pools. The following values are displayed in the header section:

Item	Description
Mshr	Number of LPAR that are running in the shared-memory mode
Mded	Number of LPAR that are running in the dedicated-memory mode
Pools	Total number of memory pools in the system
Mpsz	Total size of physical memory of all the memory pools in gigabytes
MPuse	Total memory used by LPAR associated with all the pools in gigabytes
Entl	Total I/O memory entitlement of all of the LPAR in all of the pools in gigabytes
Use	Total I/O memory entitlement in use of all of the LPAR in all of the pools in gigabytes
Mon	Total monitored memory of the system in gigabytes
InUse	Total memory in use of the system in gigabytes
Avl	Total free memory available in the system in gigabytes

The following values are displayed in the memory pools section:

Item	Description
mpid	ID of the memory pool
mpsz	Size of the total physical memory of the memory pool in gigabytes
mpus	Total memory of the memory pool in use (this value is the sum of the physical memory allocated to all of the LPAR in the pool)
mem	Aggregate logical memory size of all of the partitions in the pool in gigabytes

Item	Description
memu	Aggregate logical memory used for all of the partitions in the pool in gigabytes
iome	Aggregate of I/O memory entitlement that is configured for all of the LPAR in the pool in gigabytes
iomu	Aggregate of the I/O memory entitlement that is used for all of the LPAR in the pool in gigabytes
hpi	Aggregate number of hypervisor page faults that have occurred for all of the LPAR in the pool
hpit	Aggregate amount of time spent waiting for hypervisor page-ins by all of the LPAR in the pool in milliseconds

The following values are displayed in the partitions section:

Item	Description
mem	Logical memory size of the partition in gigabytes
memu	Logical memory that is used for the partition in gigabytes
meml	Logical memory that is loaned to the hypervisor by the LPAR
pmem	Physical memory allocated to the partition from the memory pool in gigabytes
iom	Amount of I/O memory entitlement that is configured for the LPAR in gigabytes
iomu	Amount of I/O memory entitlement that is used for the LPAR in gigabytes
hpi	Number of hypervisor page faults
hpit	Time spent waiting for hypervisor page-ins in milliseconds
vcswh	Virtual context switches as an average per second
physb	Physical processor busy
%entc	Percentage of the processor entitlement that is consumed

Comma-separated reports

The **topasout** command generates a CEC report that contains data that are separated with comma.

Use the **topasout** command with the **-c** flag to generate this report. The output file is written to *recordedfilename_01* file.

For example, to generate a report in spreadsheet format for the **topas_CEC.070221** file in the **/etc/perf/** directory, enter the following command:

```
topasout -c /etc/perf/topas_CEC.070221
```

The output file is the **topas_CEC.070221_01** file, which locates in the same directory as the original file.

The **topas** recordings support only the **-m mean** option.

The following sample shows the output of a **topas_CEC** report:

```
#Monitor: topas_CEC recording-- hostname: ptoolsl3 ValueType: mean
Time="2007/03/06 07:19:39", CEC/Lpars/monitored=3.00
Time="2007/03/06 07:19:39", CEC/Lpars/unmonitored=0.00
Time="2007/03/06 07:19:39", CEC/Lpars/shared=1.00
Time="2007/03/06 07:19:39", CEC/Lpars/dedicated=2.00
Time="2007/03/06 07:19:39", ptoolsl1/LPAR/Sys/osver=5.30
Time="2007/03/06 07:19:39", ptoolsl1/LPAR/Sys/shared=1.00
Time="2007/03/06 07:19:39", ptoolsl1/LPAR/Sys/capped=0.00
```

```
Time="2007/03/06 07:19:39", ptools11/LPAR/Sys/smt=1.00
...
```

Spreadsheet format reports

The **topasout** command generates a CEC report in spreadsheet format.

Use the **topasout** command with the **-s** flag to generate this report. The output file is written to *recordedfilename_01* file.

For example, to generate a report in spreadsheet format for the **topas_CEC.070221** file in the **/etc/perf/** directory, enter the following command:

```
topasout -s /etc/perf/topas_CEC.070221
```

The output file is the **topas_CEC.070221_01** file, which locates in the same directory as the original file.

The **topas** recordings can use only the **-m mean** option.

VIOS report

The VIOS report contains information about Virtual I/O Server/Client throughput. The following column headings are included in a Virtual I/O Server/Client throughput report:

Item	Description
Server	Name of the VIO Server.
Client	Name of the VIO Client.
KBPS	Number of kilobytes read and written per second over the monitoring interval. This field is the sum of the values of the KB-R and KB-W metrics.
TPS	Number of transfers that are issued per second.
KB-R	Number of kilobytes read per second.
KB-W	Number of kilobytes written per second.
AQD	Average number of requests waiting to be sent.
AQW	Average queue that is waiting per request reported in millisecond. The suffix indicates the unit of time. The default time unit is millisecond.
ART	Average time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.
AWT	Average time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.
MRT	Maximum time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.
MWT	Maximum time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.

VIOS adapter report

The VIOS adapter report contains information on virtual I/O server or client (VIOS) adapter and disk details. The following details on the disks are reported:

Item	Description
Adapter	Name of the server adapter.

Item	Description
Vtargets	Name of the virtual target device belonging to the server adapter.
Client_disk	Name of the client disk that is mapped to the virtual target device of the server adapter.

The following details of the adapters are displayed:

Item	Description
KBPS	Amount of data transferred (read or written) in the adapter in kilobytes per second.
TPS	Number of transfers per second issued to the adapter.
KB-R	Total number of kilobytes read from the adapter.
KB-W	Total number of kilobytes written to the adapter.
AQD	Average number of requests waiting to be sent to the virtual target device or disk.
AQW	Average queue waiting per request reported in millisecond. The suffix indicates the unit of time. The default time unit is millisecond.
ART	Average time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.
AWT	Average time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.
MRT	Maximum time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is millisecond.
MWT	Maximum time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is millisecond.

The following details of the virtual target device and the client disk are reported:

Item	Description
Busy%	Percentage of time that the virtual target device or disk is active.
KBPS	Number of kilobytes read and written per second over the monitoring interval. This field is the sum of the value of the KB-R and KB-W metrics.
TPS	Number of transfers per second that are issued to the virtual target device or disk. A transfer is an I/O request to the virtual target device or disk. Multiple logical requests can be combined into a single I/O request to the disk. A transfer is of medium size.
KB-R	Number of kilobytes read per second from the virtual target device or disk.
KB-W	Number of kilobytes written per second to the virtual target device or disk.
AQD	Average number of requests waiting to be sent to the virtual target device or disk.
AQW	Average queue waiting per request that is reported in milliseconds. The suffix indicates the unit of time. The default time unit is milliseconds.

Item	Description
ART	Average time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is milliseconds.
AWT	Average time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is milliseconds.
MRT	Maximum time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is milliseconds.
MWT	Maximum time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is milliseconds.

Flags

Item	Description
-a	The -a flag is used only for nmon analyzer report.
-b time	The time in the recorded file that the topasout command begins to generate reports from. The time can either be in the YYMMDDHHMM format or the HHMM format. You must use the same time format for end time if it is specified. YYMMDD represents year, month, and day. HHMM represents hour and minute. In HHMM format, the value must range from 0000 through 2359. The default value for begin time is 0000. The report is generated for the first day of the recording within the given time range. In YYMMDDHHMM format, the default value is the time of the first recorded data in the recording file. The command generates report for the data between the begin and end time range.
-c	Specifies that the topasout command should format the output files as comma-separated ASCII. Each line in the output files contains one time stamp and one observation.
-e time	The time in the recorded file that the topasout command stop generating reports from. The time can be in the YYMMDDHHMM format or the HHMM format. You must use the same time format for the begin time if it is specified. YYMMDD represents year, month, and day. HHMM represents hour and minute. In YYMMDDHHMM format, the default value is the time of the last recorded data in the recording file. The report is generated for the data between the begin and end date and time range. In HHMM format, the default value for end time is 2359. The report is generated for the first day of the recording within the given time range.
-i interval	The -i flag defines the interval in minute that the topasout command need to average the values. The valid values of the -i flag are 5, 10, 15, 30, or 60. The default value is 5 minutes.

Item	Description
-m <i>type</i>	By default, the topasout only outputs the mean values. Other recorded values and the full set for local recordings are available through other options including the <i>min</i> , <i>max</i> , <i>mean</i> , <i>stdev</i> , <i>set</i> , and <i>exp</i> options.
-O	The -O flag can have the following values: fullhostname=[on off]. When the -O flag is set to on, it displays the full hostname in a new column. The default value is off for the fullhostname.
-R <i>type</i>	Use the -R flag to specify the type of a report for xmwlm recordings or topasout recordings. The <i>type</i> parameter has the following variables: summary Generates Summary report. detailed Generates Detailed report. lan Generates LAN report. disk Generates Disk report. poolinfo Generate Shared-processor-pool report. mempool Generates memory pool report. If there is no memory pool, the header will be displayed without any values. adapter Generates adapter report. vadapter Generates virtual adapter report. vios Generates Virtual I/O Server/Client throughput report. vios_adapter Generates Virtual I/O Server/Client adapter and disk detailed report. The reports generated with the -R flag are printed to the console.
-s	Specifies that topasout should format the output files in a format suitable for input to spreadsheet programs.

Parameters

Item	Description
<i>xmwlm_recording_file</i>	Specifies that the input file is a recording created using the topasrec/xmwlm command.
<i>topas_recording_file</i>	Specifies that the input file is a recording created using the topasrec/topas command.
<i>nmon_recording_file</i>	Specifies that the input file is a recording created using the nmon command.

Examples

1. To generate a Detailed report from an **xmwl**m recording file from 10:00 a.m. to 11:00 p.m., enter the following command:

```
topasout -R detailed -i 15 -b 1000 -e 2300 /etc/perf/daily/xmwl.m.070226
```

2. To generate a Summary report from an **xmwl**m recording file, enter the following command:

```
topasout -R summary /etc/perf/daily/xmwl.m.070226
```

3. To generate a Disk report from an **xmwl**m recording file, enter the following command:

```
topasout -R disk /etc/perf/daily/xmwl.m.070226
```

4. To generate a LAN report from an **xmwl**m recording file, enter the following command:

```
topasout -R lan /etc/perf/daily/xmwl.m.070226
```

5. To generate an adapter report from an **xmwl**m recording file, enter the following command:

```
topasout -R adapter /etc/perf/daily/xmwl.m.070226
```

6. To generate a virtual adapter report from an **xmwl**m recording file, enter the following command:

```
topasout -R vadapter /etc/perf/daily/xmwl.m.070226
```

7. To generate a **nmon** analyzer report from an **xmwl**m recording file named **xmwl.m.070226** in the **/etc/perf/daily/** directory, enter the following command:

```
topasout -a /etc/perf/daily/xmwl.m.070226
```

The output is written to **/etc/perf/daily/xmwl.m.070226.csv**

8. To generate a Shared-processor-pool report from **topas CEC** recording, enter the following command:

```
topasout -R poolinfo /etc/perf/topas_CEC.070302
```

9. To generate a Summary report from **topas CEC** recording from 2:00 p.m. to 4:00 p.m. on the first day of recorded data, enter the following command:

```
topasout -R summary -b 1400 -e 1600 /etc/perf/topas_CEC.070302
```

10. To generate a VIOS report from a **topas CEC** recording, enter the following command:

```
topasout -R vios /etc/perf/topas_CEC.070302
```

11. To generate a VIOS adapter report from a **topas CEC** recording, enter the following command:

```
topasout -R vios_adapter /etc/perf/topas_CEC.070302
```

12. To generate a memory pool report from a **topas CEC** recording, enter the following command:

```
topasout -R mempool /etc/perf/topas_CEC.070302
```

13. To generate a summary report from a **topas CEC** recording from 2:00 p.m., March 10, 2008 to 4:00 p.m., March 12, 2008, enter the following command:

```
topasout -R summary -b 0803101400 -e 0803121600 /etc/perf/ptools11_cec_080310.topas
```

14. To generate a detailed report from a **topas Cluster** recording from 2:00 p.m., March 10, 2008 to 4:00 p.m., March 12, 2008, enter the following command:

```
topasout -R summary -b 0803101400 -e 0803121600 /etc/perf/ptools11_cluster_080310.topas
```


15. To generate a **nmon** analyzer report from an **CEC Recording** file named ptoolsl1_cec_080310.topas in the **/etc/perf/** directory enter the following command:

```
topasout -a /etc/perf/ptoolsl1_cec_080310.topas
```

16. To generate a **nmon** analyzer report from an **Cluster Recording** file named ptoolsl1_cluster_080310.topas in the **/etc/perf/** directory, enter the following command:

```
topasout -a /etc/perf/ptoolsl1_cluster_080310.topas
```

17. To report the full hostname in the detailed reported, enter the following command:

```
topasout -R detailed -O fullhostname=on
```

Location

/usr/bin/topasout

Files

Item	Description
/usr/bin/topas	Contains the topas command.
/usr/bin/xmwlm	Contains the xmwlm command.
/usr/bin/topasout	Contains the topasout command. The topasout command is included in the perfagent.tools fileset.

topasrec Command

Purpose

The **topasrec** command generates binary recording of the local system metrics, CEC (Central Electronic Complex) metrics, and Cluster metrics.

Note: The **xmwlm** and **xmtrend** utilities are obsolete and are replaced by **topasrec** command.

Syntax

Local binary recording:

```
topasrec -L [ -c sample_count ] [ -o < output_filename > ] [ -s seconds ] [ -t trace level ]
```

Local Azizo recording:

```
topasrec -L -O type=azizo
```

CEC recording:

```
topasrec -C [ -c sample_count ] [ -o < output_filename > ] [ -s seconds ] [ -O xmtopas=<hostname> ]
```

Cluster recording:

```
topasrec -G [ -c sample_count ] [ -o < output_filename > ] [ -s seconds ] [ -O xmtopas=<hostname> ]
```

List running recording:

```
topasrec -l
```

Description

Note:

1. You cannot run the **topasrec** command inside a workload partition (WPAR).

2. The CEC or cluster recording re-spawns after the partition migration or hibernation is complete. The active recording file is renamed to **<current_file_name>.mig.<HH>.<MM>.<SS>** after migration of the partition, and **<current_file_name>.hib.<HH>.<MM>.<SS>** after hibernation of the partition.

The **topasrec** command records the local system data, the cross-partition data (CEC statistics), and the cluster data in binary format.

When you run the **topasrec** command for a CEC recording, the **topasrec** command collects a set of metrics from the AIX partitions running on the same CEC. The **topasrec** command collects dedicated and shared partition data, and a set of aggregated values to provide an overview of the partition set on the same CEC.

The **topasrec** command finds metrics to be recorded from the **/usr/lpp/perfagent/daily.cf** file, and you should not alter the **daily.cf** file. Altering the **daily.cf** file affects the following recording files:

1. Persistent/nonpersistent local recordings
2. WLE Recording
3. Performance management service data collection
4. Performance PMR (**perfpmr**) data collected for performance problem analysis

The nmon, CEC, and cluster recordings are not affected by altering the **daily.cf** file. If you want to have a reduced subset of metrics for recording, you can back up the existing **daily.cf** file, and alter it to remove the metrics that you do not want to record. Removing these metrics affects all the recording files previously listed. For example, if you do not want **Disk/*/busy** metrics to be recorded by using the **topasrec** command, you can remove this line from the **/usr/lpp/perfagent/daily.cf** file.

Note: For any dynamic configuration changes to the system, the tool has to be restarted to reflect the new changes.

Flags

Item	Description
-C	Records CEC statistics in binary format. The -C flag specifies that the cross-partition statistics are to be recorded.
-c sample_count	Records the specified number of records and then stops. If the -c flag is not specified, or if the value of the sample_count parameter is zero, the recording is continuous and the topasrec command writes to the recording file until it is stopped.
-L	Records local statistics in binary format.
-l	Lists the recordings that are running.
-s seconds	Specifies the recording interval in seconds. The value of the seconds parameter should be a multiple of 60. For continuous recordings (topasrec -c 0) of CEC and local statistics, the default value of recording interval is 900 seconds. For a sample count that is greater than zero, the default value of the recording interval is 300 seconds.
-O xmtopas=<hostname>	Specifies the name of the host that aggregates the data and provides it to topasrec . If this is not specified, topasrec will get data from one of the known aggregators. Note: You cannot use the override option with persistent recording.

Item

-o < *output_filename* >

Description

Specifies the name of the output file. The value of the *output_filename* parameter can be a directory with an optional file prefix. You can specify one of the following types of file names to the *output_filename* parameter:

- A directory. The directory should always end with /. For example, the **/etc/perf/** directory.
- A directory with a file name. For example, the **/home/tester/perf_load** file.
- A file name. For example, the **perf_load** file.

The default output file is the current directory (./).

In CEC recording, Cluster Recording and local recording, the default prefix of the file name is the host name.

If you provide a file name that contains a directory and a file name prefix in the **-o output_filename** flag, the name of the recorded file is in the following format:

- For CEC metrics, the output is in the following format:
 <filename>_cec_YYMMDD_HHMM.topas
- For Cluster metrics, the output is in the following format:
 <filename>_cluster_YYMMDD_HHMM.topas
- For local metrics, the output is in the following format:
 <filename>_YYMMDD_HHMM.topas

If you provide a file name that contains only the directory prefix, the name of the recorded file is in the following format:

- For CEC metrics, the output is in the following format:
 <filename/hostname>_cec_YYMMDD_HHMM.topas
- For Cluster metrics, the output is in the following format:
 <filename/hostname>_cluster_YYMMDD_HHMM.topas
- For local metrics, the output is in the following format:
 <filename/hostname>_YYMMDD_HHMM.topas

In these formats, year (YY), month (MM), day (DD), hour (HH), and minute (MM) correspond to the time when the recording file is created.

Note: For CEC/Cluster Recording, if **xmtopas** override option is used then filename will be the value specified for **xmtopas=<value>**.

Example:

```
< value>_cec_YYMMDD_HHMM.topas
```

```
< value>_cluster_YYMMDD_HHMM.topas
```

-r *retention*

Specifies the number of days for which the file must be retained. The minimum value is 1. For example, **-r 5** specifies that the file is retained for five days.

Item	Description
-R <i>max_days_per_file</i>	Specifies the number of days for which the performance data needs to be written to a file. The minimum value is 1 and maximum value is 366. For example, if we start a persistent recording with option -R 2 on day 1, the performance data of day 1 and day 2 are written to the same file. On day 3, a new file is created that contains the performance data of day 3 and day 4.
-t <i>trace level</i>	Specifies the trace level. The trace level can be set from 1 to 9.

Parameters

Item	Description
<i>sample_count</i>	Specifies the number of records to generate.
<i>output_filename</i>	Specifies the name of the output file.
<i>seconds</i>	Specifies the recording interval in seconds.

Examples

1. To start a local binary recording that runs for 5 minutes and contains system metrics every 1 minute, enter the following command:

```
topasrec -L -c 5 -s 60
```

If the file is created at 23:14, Mar 10, 2008, and the host name is ses15, then the output file name is ./ses15_080310_2314.topas.

2. To start a continuous local binary recording with a /home/test/sample file name, enter the following command:

```
topasrec -L -o /home/test/sample
```

If the file is created at 12:05, Mar 10, 2008, and the host name is ses15, then the output file name is /home/test/sample_080310_1205.topas.

3. To start a CEC recording that runs for 20 minutes with metrics recorded at 120-second intervals, and generate an output file named sample, enter the following command:

```
topasrec -C -o sample -s 120 -c 10
```

If the file is created at 08:07, Feb 1, 2008, and the host name is ses15, then the output file name is ./sample_cec_080201_0807.topas.

4. To start a continuous local binary recording with a /home/test/sample_bin file name, enter the following command:

```
topasrec -C -o /home/test/sample_bin
```

If the file is created at 04:20, Feb 1, 2008, and the host name is ses15, then the output file name is /home/test/sample_bin_080201_0420.topas.

5. To list the details of the running recordings, enter the following command:

```
topasrec -l
```

6. To enable trace, enter the following command:

```
topasrec -L -t 1
```

7. To start a Cluster recording that runs for 20 minutes with metrics recorded at 120-second intervals, and generate an output file named sample, enter the following command:

```
topasrec -G -o sample -s 120 -c 10
```

If the file is created at 08:07, Feb 1, 2008 and the host name is ses15 then the output file name is /sample_cluster_080201_0807.topas..

8. To start a continuous local Cluster recording with a /home/test/sample_bin file name, enter the following command:

```
topasrec -G -o /home/test/sample_bin
```

9. To manually start a local azizo recording, enter the following command:

```
topasrec -L -O type=azizo
```

If a valid /etc/perf/xmtopas.cf file is present, the azizo recording is automatically started by the **xmtopas** command. After the recording is started, it generates the azizo.<yymmdd> file in the /etc/perf/ directory and runs only if the **xmtopas** command is running

Files

Item	Description
/usr/bin/topasrec	Contains the topasrec command.

topsvcs Command

Purpose

Starts or restarts topology services on a cluster node.

Syntax

topsvcs

Description

Use topsvcs script to start the operation of topology services for a cluster.

The topsvcs script is not normally executed from the command line. It is normally called by the topsvcsctl control script, which is in turn called by the HACMP/ES startup process.

The topsvcs script issues these commands:

```
no -o nonlocsrcroute=1
no -o ipsrcroutesend=1
no -o ipsrcrouterrecv =1
no -o ipsrcrouteforward=1
```

These commands enable IP source routing. Do not change this setting, because the topology services subsystem requires this setting to work properly. If you change the setting, the topology services subsystem and a number of other subsystems that depend on it will no longer operate properly.

Flags

- s Instructs the topology services daemon to reject messages that are apparently delayed.

-d

Instructs the topology services daemon not to reject messages that are apparently delayed (this is the default).

Security

You must have root privilege to run this command.

Exit Status

0

Indicates the successful completion of the command.

1

Indicates the command was unsuccessful.

Environment Variables

HB_SERVER_SOCKET

This environment variable should be set before this command can be executed. It must be set to the location of the UNIX-domain socket used by topology services clients to connect to the topology services daemon. This environment variable must be set to `/var/ha/soc/hats/server_socket.partition name`.

HA_SYSPAR_NAME

If HB_SERVER_SOCKET is not set, then HA_SYSPAR_NAME must be set to the partition name.

Restrictions

This command is valid in an HACMP environment only.

Use this command *only* under the direction of the IBMSupport Center.

Standard Output

When the `-h` flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

This command writes error messages (as necessary) to standard error.

Examples

To instruct the topology services daemon on the local node to start discarding apparently delayed messages, enter:

```
export HA_SYSPAR_NAME=partition1
/opt/rsct/bin/hatsoptions -s
```

Location

/opt/rsct/bin/topsvcs

Contains the topsvcs script

Files

`/var/ha/soc/hats/server_socket.partition name`

topsvcsctrl Command

Purpose

Starts the topology services subsystem.

Syntax

```
topsvcsctrl { -a | -s | -k | -d | -c | -u | -t | -o | -r | -h }
```

Description

The `topsvcsctrl` control script controls the operation of the topology services subsystem. The subsystem is under the control of the system resource controller (SRC) and belongs to a subsystem group called `topsvcs`. This script is normally started by the HACMP/ES startup process.

An instance of the topology services subsystem runs on every node of a cluster.

From an operational point of view, the topology services subsystem group is organized as follows:

Subsystem

topology services

Subsystem group

topsvcs

SRC subsystem

topsvcs

The `topsvcs` subsystem is associated with the `hatsd` daemon and the `topsvcs` script. The `topsvcs` script configures and starts the `hatsd` daemon. The subsystem name on the nodes is `topsvcs`. There is one of each subsystem per node and it is associated with the cluster to which the node belongs.

Daemons

`hatsd`

Provides the topology services. The `topsvcs` script configures and starts the `hatsd` daemon.

The `topsvcsctrl` script is not normally executed from the command line. It is normally called by the HACMP/ES startup command.

The `topsvcsctrl` script provides a variety of controls for operating the topology services subsystems:

- Adding, starting, stopping, and deleting the subsystems
- Cleaning up the subsystems, that is, deleting them from all system partitions
- Turning tracing on and off
- Refreshing the subsystem

Before performing any of these functions, the script obtains the current cluster name (using the `cllscldr` command) and the node number (using the `clhandle` command). If the node number is 0, the control script is running on the control workstation.

Except for the `clean` and `unconfigure` functions, all functions are performed within the scope of the current system partition.

Adding the subsystem: When the `-a` flag is specified, the control script uses the `mkssys` command to add the topology services subsystem to the SRC. The control script operates as follows:

1. It makes sure the `topsvcs` subsystem is stopped.
2. It removes the `topsvcs` subsystem from the SRC (in case it is still there).
3. It adds the `topsvcs` subsystem to the SRC.

Starting the subsystem: When the `-s` flag is specified, the control script uses the `startsrc` command to start the topology services subsystem, `topsvcs`.

Stopping the subsystem: When the `-k` flag is specified, the control script uses the `stopsrc` command to stop the topology services subsystem, `topsvcs`.

Deleting the subsystem: When the `-d` flag is specified, the control script uses the `rmssys` command to remove the topology services subsystem from the SRC. The control script operates as follows:

1. It makes sure that the `topsvcs` subsystem is stopped.
2. It removes the `topsvcs` subsystem from the SRC using the `rmssys` command.
3. It removes the port number from the `/etc/services` file.

Cleaning up the subsystems: When the `-c` flag is specified, the control script stops and removes the topology services subsystems for all clusters partitions from the SRC. The control script operates as follows:

1. It stops all instances of subsystems in the clusters, using the `stopsrc -g topsvcs` command.
2. It removes all entries for the `topsvcs` subsystem from the `/etc/services` file.

Turning tracing on: When the `-t` flag is specified, the control script turns tracing on for the `hatsd` daemon, using the `traceson` command.

Turning tracing off: When the `-o` flag is specified, the control script turns tracing off (returns it to its default level) for the `hatsd` daemon, using the `tracesoff` command.

Refreshing the subsystem: When the `-r` flag is specified, the control script refreshes the subsystem, using the `topsvcs refresh` command and the `refresh` command. It rebuilds the information about the node and adapter configuration in the global object data manager (ODM) and signals the daemon to read the rebuilt information.

Logging: While it is running, the topology services daemon (`hatsd`) provides information about its operation and errors by writing entries in a log file called `/var/ha/log/topsvcs.cluster_name`.

Flags

Item	Description
-a	Adds the subsystem.
-s	Starts the subsystem.
-k	Stops the subsystem.
-d	Deletes the subsystem.
-c	Cleans the subsystems.
-u	Removes the topology services subsystem from all partitions.
-t	Turns tracing on for the subsystem.
-o	Turns tracing off for the subsystem.
-r	Refreshes the subsystem.
-h	Writes the script's usage statement to standard output.

Security

You must be running with an effective user ID of `root` to use this script.

Exit Status

- 0**
- Indicates that the script completed successfully.

1

Indicates that an error occurred.

Environment Variables

HB_SERVER_SOCKET

This environment variable should be set before this command can be executed. It must be set to the location of the UNIX-domain socket used by topology services clients to connect to the topology services daemon. This environment variable must be set to `/var/ha/soc/hats/server_socket.partition name`.

HA_SYSPAR_NAME

If HB_SERVER_SOCKET is not set, then HA_SYSPAR_NAME must be set to the partition name.

Restrictions

This command is valid in an HACMP environment only.

Use this command *only* under the direction of the IBM Support Center.

Standard Output

When the `-h` flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

This script writes error messages (as necessary) to standard error.

Examples

1. To add the topology services subsystem to the SRC, enter:

```
topsvcsctrl -a
```

2. To start the topology services subsystem, enter:

```
topsvcsctrl -s
```

3. To stop the topology services subsystem, enter:

```
topsvcsctrl -k
```

4. To delete the topology services subsystem from the SRC, enter:

```
topsvcsctrl -d
```

5. To clean up the topology services subsystem, enter:

```
topsvcsctrl -c
```

6. To turn tracing on for the topology services daemon, enter:

```
topsvcsctrl -t
```

7. To turn tracing off for the topology services daemon, enter:

```
topsvcsctrl -o
```

Location

/opt/rsct/bin/topsvcsctrl

Contains the topsvcctrl script

Files

/var/ha/log/topsvcs.cluster_name

Contains the log of the hatsd daemon on the cluster named cluster_name

touch Command

Purpose

Updates the access and modification times of a file.

Syntax

touch [**-a**] [**-c**] [**-m**] [**-f**] [**-r** *RefFile*] [*Time* | **-t** *Time* | **-d** *date_time*] { *File* ... | *Directory* ... }

Note: The preceding syntax is applicable only when the UNIX03 mode is not enabled in the AIX operating system.

touch [**-a c m f**] [**-r** *ref_file* | **-t** *time* | **-d** *date_time*] *file...* | *Directory*

Note: The preceding syntax is applicable only when the UNIX03 mode is enabled. To enable the UNIX03 mode in the AIX operating system, you must set the value of the **XPG_SUS_ENV** environment variable to **ON**.

Description

The **touch** command updates the access and modification times of each file specified by the *File* parameter of each directory specified by the *Directory* parameter. If you do not specify a value for the *Time* variable, the **touch** command uses the current time. If you specify a file that does not exist, the **touch** command creates the file unless you specify the **-c** flag.

The return code from the **touch** command is the number of files for which the times could not be successfully modified (including files that did not exist and were not created).

The **-a** and **-m** flags are active even when you do not specify them in the **touch** command.

Flags

Item	Description
-a	Changes the access time of the file specified by the <i>File</i> variable. Does not change the modification time unless -m is also specified.
-c	Does not create the file if it does not already exist. No diagnostic messages are written concerning this condition.

Item	Description
-d <i>Date_Time</i>	Uses the specified date and time instead of the current time. The <i>date_time</i> variable is specified in the decimal format, YYYY-MM-DDThh:mm:ss[.frac][tz] or YYYY-MM-DDThh:mm:ss[.frac][tz], where: YYYY Specifies the four digits of the year (0000 to 9999). MM Specifies the month of the year (01 to 12). DD Specifies the day of the month (01 to 31). hh Specifies the hour of the day (00 to 23). mm Specifies the minute of the hour (00 to 59). ss Specifies the second of the minute (00 to 59). T Indicates a time designator and can be replaced with a single space. [.frac] Specifies a fractional second. It can either be blank or a period (.) followed by one or more decimal digits. [,frac] Specifies a fractional second. It is a comma (,) followed by one or more decimal digits. [tz] If the value of the <i>[tz]</i> parameter is blank, the local time zone is used for the resulting time. If the value of the <i>[tz]</i> parameter is a character Z, the Coordinate Time Zone (UTC) is used for the resulting time. If the value of the <i>[tz]</i> parameter is blank, the value of the TimezoneInfo (TZ) environment variable is used to identify the resulting time. The value of the <i>[tz]</i> parameter is implementation-specific if the resulting time is earlier than the Epoch time. Also, if the resulting time cannot be represented as a timestamp of the file that is specified by the <i>File</i> parameter, the touch command exits with an error status.
-f	Attempts to force the touch in spite of read and write permissions on a file.
-m	Changes the modification time of <i>File</i> . Does not change the access time unless -a is also specified.
-r <i>RefFile</i>	Uses the corresponding time of the file specified by the <i>RefFile</i> variable instead of the current time.

Item	Description
<i>Time</i>	Specifies the date and time of the new timestamp in the format <i>MMDDhhmm</i>[<i>YY</i>], where: MM Specifies the month of the year (01 to 12). DD Specifies the day of the month (01 to 31). hh Specifies the hour of the day (00 to 23). mm Specifies the minute of the hour (00 to 59). YY Specifies the last two digits of the year. If the <i>YY</i> variable is not specified, the default value is the current year (70 to 99 or 00 to 37). If the value of the <i>YY</i> digits is between 70 and 99, the century is assumed to be 19. If the value of the <i>YY</i> digits is between 00 and 37, the century is assumed to be 20.
-t <i>Time</i>	Uses the specified time instead of the current time. The <i>Time</i> variable is specified in the decimal form <i>[[CC]YY]MMDDhhmm[.SS]</i> where: CC Specifies the first two digits of the year (19 to 21). YY Specifies the last two digits of the year (00 to 99). If the value of the <i>YY</i> digits is between 70 and 99, the value of the <i>CC</i> digits is assumed to be 19. If the value of the <i>YY</i> digits is between 00 and 37, the value of the <i>CC</i> digits is assumed to be 20. For years after 2038, specify the year in the <i>yyyy</i> format. MM Specifies the month of the year (01 to 12). DD Specifies the day of the month (01 to 31). hh Specifies the hour of the day (00 to 23). mm Specifies the minute of the hour (00 to 59). SS Specifies the second of the minute (00 to 59).

Note:

1. The **touch** command calls the **utimenstat()** subroutine to change the modification and access times of the file touched. This may cause the **touch** command to fail when flags are used if you do not actually own the file, even though you may have write permission to the file.
2. Do not specify the full path name **/usr/bin/touch** if you receive an error message when using the **touch** command.

Exit Status

This command returns the following exit values:

Item	Description

0	The command executed successfully. All requested changes were made.
>0	An error occurred.

Security

hm

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To update the access and modification times of a file, enter:

```
touch program.c
```

This sets the last access and modification times of the `program.c` file to the current date and time. If the `program.c` file does not exist, the **touch** command creates an empty file with that name.

2. To avoid creating a new file, enter:

```
touch -c program.c
```

3. To update only the modification time, enter:

```
touch -m *.o
```

This updates the last modification times (not the access times) of the files that end with a `.o` extension in the current directory. The **touch** command is often used in this way to alter the results of the **make** command.

4. To explicitly set the access and modification times, enter:

```
touch -c -t 02171425 program.c
```

This sets the access and modification dates to 14:25 (2:25 p.m.) February 17 of the current year.

5. To use the time stamp of another file instead of the current time, enter:

```
touch -r file1 program.c
```

This gives the `program.c` file the same time stamp as the `file1` file.

6. To touch a file using a specified time other than the current time, enter:

```
touch -t 198503030303.55 program.c
```

This gives the `program.c` file a time stamp of 3:03:55 a.m. on March 3, 1985.

Files

Item	Description
<code>/usr/bin/touch</code>	Contains the touch command.

tpm_activate Command

Purpose

Changes the Trusted Platform Module (TPM) active states.

Syntax

```
tpm_activate [ -a ] [ -h ] [ -i ] [ -l [ none | error | info | debug ] ] [ -s ] [ -t ] [ -v ]
```

Description

The **tpm_activate** command reports the status of the TPM flags regarding the active state of the TPM. This is the default behavior, and it is also accessible through the **-s** (or **--status**) option. It prompts for the owner password when it reports the TPM status.

The **-a** (or **--active**) option changes the TPM to the active state (through the **TPM_PhysicalSetDeactivated** API). This operation is persistent. It requires physical presence for authorization, and a system reboot operation to take effect.

The **-i** (or **--inactive**) option (through the **TPM_PhysicalSetDeactivated** API) changes the TPM to the inactive state. This operation is persistent. It requires physical presence for authorization, and a system reboot operation to take effect. Although an inactive TPM can be considered to be off, it still allows the **tpm_takeownership** command to run.

The **-t** (or **--temp**) option causes immediate TPM deactivation (through the **TPM_SetTempDeactivated** API) to occur but persists only for the current boot cycle.

The **-s** (or **--status**), **-a** (or **--active**), **-i** (or **--inactive**), and **-t** (or **--temp**) options are mutually exclusive and the last option on the command line is carried out.

Flags

Item	Description
-a (or --active)	Makes the TPM active. This operation is persistent. The operation requires physical presence for authorization, and a system reboot operation to take effect.
-h (or --help)	Displays the command usage information.
-i (or --inactive)	Makes the TPM inactive. This operation is persistent. The operation requires physical presence for authorization, and a system reboot operation to take effect.
-l (or --log) [none error info debug]	Sets the logging level to none, error, info, or debug as specified.
-s (or --status)	Reports the status of flags regarding the TPM active states.
-t (or --temp)	Makes the TPM inactive for the current boot cycle only.
-v (or --version)	Displays the command version information.

tpm_changeauth Command

Purpose

Changes the authorization data that is associated with the owner or storage root key.

Syntax

tpm_changeauth [**-g**] [**-h**] [**-l** [**none** | **error** | **info** | **debug**]] [**-n**] [**-o**] [**-r**] [**-s**] [**-u**] [**-v**] [**-z**]

Description

The **tpm_changeauth** command is used to change the authorization data for the Trusted Platform Module (TPM) owner or the TPM storage root key (through the **TPM_ChangeAuthOwner** API). This operation prompts for the current password, prompts for the new password, and prompts for a confirmation of the new password. The **-o** (or **--owner**) option changes the TPM owner password and the **-s** (or **--srk**) option changes the TPM storage root key (SRK) password.

Flags

Item	Description
-g (or --original_password_unicode)	Uses the Trusted Computing Group Software Stack (TSS) UNICODE encoding for the original password to comply with the applications that are using the TSS popup boxes.
-h (or --help)	Displays the command usage information.
-l (or --log) [none error info debug]	Sets the logging level to none, error, info, or debug as specified.
-o (or --owner)	Changes the authorization data for the TPM owner.
-n (or --new_password_unicode)	Uses the TSS UNICODE encoding for the new password to comply with the applications that are using the TSS popup boxes.
-r (or --set-well-known)	Changes the password to a secret of all zeros (20 bytes of zeros). It must be specified which password (owner, SRK or both) needs to be changed.
-s (or --srk)	Changes the authorization data for the TPM storage root key.

Item	Description
-u (or --unicode)	Use the TSS UNICODE encoding for the passwords to comply with the applications that are using the TSS popup boxes.
-v (or --version)	Displays the command version information.
-z (or --well-known)	Changes the password to a new one when the current owner password is a secret of all zeros (20 bytes of zeros). It must be specified which password (owner, SRK, or both) needs to be changed.

tpm_clear Command

Purpose

Returns the Trusted Platform Module (TPM) to the default state (unowned, disabled, and inactive).

Syntax

```
tpm_clear [ -f ] [ -h ] [ -l [ none | error | info | debug ] ] [ -u ] [ -v ] [ -z ]
```

Description

The **tpm_clear** command requests the system TPM to perform a clear operation (through the **TPM_OwnerClear** API), which clears all the ownership information. Consequently, it invalidates all keys and the data that is tied to the TPM and disables and deactivates the TPM. This operation prompts for the owner password. The **-f** (or **--force**) option relies on the physical presence to authorize the command (through the **TPM_ForceClear** API) by skipping the owner password prompt.

Note: The **TPM_OwnerClear** API can be disabled until the current owner is cleared by using the **-f** (or **--force**) option with the **tpm_setclearable** command. The **TPM_ForceClear** API can be disabled for the current boot cycle with the **tpm_setclearable** command. This command requires you to reboot the system to complete the operation.

Flags

Item	Description
-f (or --force)	Lets the TPM rely on the physical presence for authorization, thus, skipping the owner password prompt.
-h (or --help)	Displays the command usage information.
-l (or --log) [none error info debug]	Sets the logging level to none, error, info, or debug as specified.

Item	Description
-u (or --unicode)	Uses the Trusted Computing Group Software Stack (TSS) UNICODE encoding for the passwords to comply with the applications that are using the TSS popup boxes.
-v (or --version)	Displays the command version information.
-z (or --well-known)	Changes the password to a new one when the current owner password is a secret of all zeros (20 bytes of zeros). It must be specified which password (owner, storage root key, or both) needs to be changed.

tpm_clearable Command

Purpose

Disables the Trusted Platform Module (TPM) clear operations.

Syntax

tpm_clearable [**-f**] [**-h**] [**-l** [**none** | **error** | **info** | **debug**]] [**-o**] [**-s**] [**-u**] [**-v**] [**-z**]

Description

The **tpm_clearable** command reports the status of TPM flags regarding how the TPM can be cleared. This behavior is the default behavior, and it is also accessible through the **-s** (or **--status**) option. For requesting the TPM status report, it prompts for the owner password.

The **-o** (or **--owner**) option requests the TPM to disable the clear operations (through the **TPM_DisableOwnerClear** API) thus, disabling the owner from clearing out the ownership information. This operation prompts for the owner password. This operation remains in effect until the current owner is cleared.

The **-f** (or **--force**) option (through the **TPM_DisableForceClear** API) disables TPM clear operations by using physical presence to authorize a clear operation. This operation does not require authorization and skips the owner password prompt. This operation remains in effect only until a system reboot operation.

Flags

Item	Description
-f (or --force)	Disables the use of physical presence for authorizing a clear operation until a system reboot operation occurs.
-h (or --help)	Displays the command usage information.
-l (or --log) [none error info debug]	Sets the logging level to none, error, info, or debug as specified.
-o (or --owner)	Disables the use of owner authorization for authorizing a clear operation until a new owner exists.
-s (or --status)	Report the status of flags regarding how the TPM can be cleared.
-u (or --unicode)	Uses the Trusted Computing Group Software Stack (TSS) UNICODE encoding for the passwords to comply with the applications that are using the TSS popup boxes.
-v (or --version)	Displays the command version information.
-z (or --well-known)	Changes the password to a new one when the current owner password is a secret of all zeros (20 bytes of zeros). It must be specified which password (owner, storage root key, or both) needs to be changed.

tpm_createek Command

Purpose

Creates an endorsement key pair on the Trusted Platform Module (TPM).

Syntax

```
tpm_createek [ -h ] [ -l [ none | error | info | debug ] ] [ -v ]
```

Description

The **tpm_createek** command creates an endorsement key pair on the TPM (through the **TPM_CreateEndorsementKeyPair** API). The endorsement key pair is not often required because it is normally installed as a part of manufacturing. However, you might need to run this command if commands such as **tpm_getpubek** are returning error code from the TPM layer.

Flags

Item	Description
-h (or --help)	Displays the command usage information.
-l (or --log) [none error info debug]	Sets the logging level to none, error, info, or debug as specified.
-v (or --version)	Displays the command version information.

tpm_enable Command

Purpose

Changes the Trusted Platform Module (TPM) enabled states.

Syntax

```
tpm_enable [ -e ] [ -d ] [ -h ] [ -l [ none | error | info | debug ] ] [ -o ] [ -s ] [ -u ] [ -v ] [ -z ]
```

Description

The **tpm_enable** command reports the status of the TPM flags regarding the enabled state of the TPM. This is the default behavior, and it is also accessible through the **-s** (or **--status**) option. For requesting the TPM status report, it prompts for the owner password.

The **-e** (or **--enable**) option changes the system TPM to the enabled state (through the **TPM_OwnerSetDisable** API). This operation is persistent, and it prompts for the owner password.

The **-d** (or **--disable**) option (through the **TPM_OwnerSetDisable** API) changes the system TPM to the disabled state. This operation is persistent, and it prompts for the owner password. A disabled TPM can be considered to be off, and it does not allow the **tpm_takeownership** command to run.

The **-f** (or **--force**) option overrides the owner password prompt, and it relies on physical presence for the operation authorization (through the **TPM_PhysicalEnable** and **TPM_PhysicalDisable** APIs).

The **--enable**, **--disable**, and **--status** options are mutually exclusive, and the last option on the command line is carried out.

Flags

Item	Description
-e (or --enable)	Enables the TPM. This operation is persistent, and it prompts for owner authorization.
-d (or --disable)	Disables the TPM. This operation is persistent, and it prompts for owner authorization.
-h (or --help)	Displays the command usage information.
-l (or --log) [none error info debug]	Sets the logging level to none, error, info, or debug as specified.
-o (or --owner)	Overrides the prompt for owner authorization and uses physical presence to authorize the action.
-s (or --status)	Reports the status of flags regarding the TPM-enabled states.
-u (or --unicode)	Uses the Trusted Computing Group Software Stack (TSS) UNICODE encoding for the passwords to comply with the applications that are using the TSS popup boxes.
-v (or --version)	Displays the command version information.
-z (or --well-known)	Changes the password to a new one when the current owner password is a secret of all zeros (20 bytes of zeros). It must be specified which password (owner, storage root key, or both) needs to be changed.

tpm_getpubek Command

Purpose

Displays the public part of the Trusted Platform Module (TPM) endorsement key.

Syntax

tpm_createek [**-h**] [**-l** [**none** | **error** | **info** | **debug**]] [**-u**] [**-v**] [**-z**]

Description

The **tpm_getpubek** command requests the TPM's public part of the endorsement key (through the **TPM_ReadPubek** API). This operation can be restricted to require owner authorization. In that case, the command prompts for the owner password and requests the data (through the **TPM_OwnerReadPubek** API). The public key information is displayed on a successful call.

Flags

Item	Description
-h (or --help)	Displays the command usage information.
-l (or --log) [none error info debug]	Sets the logging level to none, error, info, or debug as specified.
-u (or --unicode)	Uses the Trusted Computing Group Software Stack (TSS) UNICODE encoding for the passwords to comply with the applications that are using the TSS popup boxes.
-v (or --version)	Displays the command version information.

Item	Description
-z (or --well-known)	Changes the password to a new one when the current owner password is a secret of all zeros (20 bytes of zeros). It must be specified which password (owner, storage root key, or both) needs to be changed.

tpm_ownable Command

Purpose

Verifies whether the Trusted Platform Module (TPM) allows the **tpm_takeownership** command to run.

Syntax

tpm_ownable [**-a**] [**-h**] [**-l** [**none** | **error** | **info** | **debug**]] [**-p**] [**-s**] [**-u**] [**-v**] [**-z**]

Description

The **tpm_ownable** command reports the status of the TPM flags regarding whether the TPM can be owned. This is the default behavior, and it is also accessible through the **-s** (or **--status**) option. Requesting a report of this status prompts for the owner password. The **-a** (or **--allow**) option sets the system TPM to allow **tpm_takeownership** operations (through the **TPM_SetOwnerInstall** API). This operation requires physical presence.

The **-p** (or **--prevent**) option (through the **TPM_SetOwnerInstall** API) prevents the TPM from accepting the **tpm_takeownership** command. This operation requires physical presence. These operations are persistent, and the **tpm_takeownership** command requires the TPM be enabled.

Flags

Item	Description
-a (or --allow)	Allows the tpm_takeownership command to run.
-h (or --help)	Displays the command usage information.
-l (or --log) [none error info debug]	Sets the logging level to none, error, info, or debug as specified.
-p (or --prevent)	Prevents the tpm_takeownership command to run.
-s (or --status)	Reports the status of flags regarding whether the TPM can be owned.
-u (or --unicode)	Uses the Trusted Computing Group Software Stack (TSS) UNICODE encoding for the passwords to comply with the applications that are using the TSS popup boxes.
-v (or --version)	Displays the command version information.
-z (or --well-known)	Changes the password to a new one when the current owner password is a secret of all zeros (20 bytes of zeros). It must be specified which password (owner, storage root key, or both) needs to be changed.

tpm_present Command

Purpose

Changes the physical presence states and settings of the Trusted Platform Module (TPM).

Syntax

```
tpm_present [ -a ] [ -c ] [ --disable-cmd ] [ --disable-hw ] [ --enable-cmd ] [ --enable-hw ] [ -h ] [ -l  
[ none | error | info | debug ] ] [ --lock ] [ --set-lifetime-lock ] [ -u ] [ -v ] [ -z ] [ -y ]
```

Description

The **tpm_present** command reports the status of the TPM flags regarding TPM physical presence. This behavior is the default behavior, and it is also accessible through the **--status** option. It prompts for the owner password when it reports the TPM status. All changes are made with the **TSC_Physical Presence** API.

Flags

Item	Description
-a (or --assert)	Asserts that an administrator is physically present at the system.
-c (or --clear)	Removes the assertion that an administrator is physically present at the system.
--disable-cmd	Disallows the use of commands to signal that an administrator is physically present.
--disable-hw	Disallows the use of hardware signals to signal that an administrator is physically present.
--enable-cmd	Allows the use of commands to signal that an administrator is physically present.
--enable-hw	Allows the use of hardware signals to signal that an administrator is physically present.
-h (or --help)	Displays the command usage information.
-l (or --log) [none error info debug]	Sets the logging level to none, error, info, or debug as specified.
--lock	Locks the assertions of physical presence in the current states until a system reboot operation.
--set-lifetime-lock	Allows no further changes to the flags controlling how physical presence can be signaled permanently. This option can never be undone.
-u (or --unicode)	Uses the Trusted Computing Group Software Stack (TSS) UNICODE encoding for the passwords to comply with the applications that are using the TSS popup boxes.
-v (or --version)	Displays the command version information.
-z (or --well-known)	Changes the password to a new one when the current owner password is a secret of all zeros (20 bytes of zeros). It must be specified which password (owner, storage root key, or both) needs to be changed.
-y (or --yes)	Answers yes to all questions. This flag is applicable only with the --set-lifetime-lock flag.

tpm_restrictpubek Command

Purpose

Restricts the ability to display the public part of the endorsement key to the owner.

Syntax

```
tpm_restrictpubek [ -h ] [ -l [ none | error | info | debug ] ] [ -r ] [ -s ] [ -v ]
```

Description

The **tpm_restrictpubek** command reports the status of who can display the public part of the endorsement key. This is the default behavior, and it is also available with the **-s** (or **--status**) option. This operation remains in effect until the owner is cleared and it prompts for the owner password. With the **-r** (or **--restrict**) option, the ability to display the public part of the endorsement key is restricted to the owner (through the **TPM_DisablePubekRead** API). The command prompts for the owner password to complete the operation. The **--status** and **--restrict** options are mutually exclusive, and the last option on the command line is carried out.

Flags

Item	Description
-h (or --help)	Displays the command usage information.
-l (or --log) [none error info debug]	Sets the logging level to none, error, info, or debug as specified.
-r (or --restrict)	Restricts the owner to see the public part of the endorsement key.
-s (or --status)	Displays the status of who can see the public part of the endorsement key to the owner.
-u (or --unicode)	Uses the Trusted Computing Group Software Stack (TSS) UNICODE encoding for the passwords to comply with the applications that are using the TSS popup boxes.
-v (or --version)	Displays the command version information.
-z (or --well-known)	Changes the password to a new one when the current owner password is a secret of all zeros (20 bytes of zeros). It must be specified which password (owner, storage root key, or both) needs to be changed.

tpm_selftest Command

Purpose

Requests that the Trusted Platform Module (TPM) perform a self-test and report the results.

Syntax

tpm_selftest [**-h**] [**-l** [**none** | **error** | **info** | **debug**]] [**-r**] [**-v**]

Description

The **tpm_selftest** command requests that the system TPM performs a self-test (through the **TPM_SelfTestFull** API) and report the results. The **-r** (or **--results**) option reports the outcome of the last self-test operation without requesting another test to be run. If the TPM fails the self-test, it enters the failure mode where no commands are accepted. The results are reported in a manufacturer-specific format. The TPM self-test always runs automatically at every boot operation.

Flags

Item	Description
-h (or --help)	Displays the command usage information.
-l (or --log) [none error info debug]	Sets the logging level to none, error, info, or debug as specified.

Item	Description
-r (or --results)	Reports results only.
-v (or --version)	Displays the command version information.

tpm_takeownership Command

Purpose

Sets up an owner on the Trusted Platform Module (TPM).

Syntax

tpm_takeownership [**-h**] [**-l** [**none** | **error** | **info** | **debug**]] [**-u**] [**-v**] [**-z**]

Description

The **tpm_takeownership** command sets up an owner on the system TPM (through the **TPM_TakeOwnership** API). This operation requires that the TPM be enabled and restricted by the **tpm_setownable** command. The command prompts for owner and security root key passwords and confirmations. This command can take a while to process.

Flags

Item	Description
-h (or --help)	Displays the command usage information.
-l (or --log) [none error info debug]	Sets the logging level to none, error, info, or debug as specified.
-u (or --unicode)	Uses the Trusted Computing Group Software Stack (TSS) UNICODE encoding for the passwords to comply with the applications that are using the TSS popup boxes.
-v (or --version)	Displays the command version information.
-y (or --owner-well-known)	Sets the owner secret to all zeros (20 bytes of zeros).
-z (or --well-known)	Changes the password to a new one when the current owner password is a secret of all zeros (20 bytes of zeros). It must be specified which password (owner, storage root key, or both) needs to be changed.

tpm_version Command

Purpose

Reports the Trusted Platform Module (TPM) version and manufacturer information.

Syntax

tpm_version [**-h**] [**-l** [**none** | **error** | **info** | **debug**]] [**-v**]

Description

The **tpm_version** command reports the system TPM version and manufacturer information. The information reported is specific to the manufacturer.

Flags

Item	Description
-h (or --help)	Displays the command usage information.
-l (or --log) [none error info debug]	Sets the logging level to none, error, info, or debug as specified.
-v (or --version)	Displays the command version information.

tprof Command

Purpose

Reports processor usage.

Syntax

```
tprof { [ -c ] [ -C { all | cpulist } ] [ -d ] [ -D ] [ -e ] [ -@ { ALL | wparlist } ] [ { [ -E [ mode [ -b ] [ -B ] ] ]  
[ -f frequency ] ] [ -F ] [ -I ] [ -j ] [ -k ] [ -l ] [ -L objectlist ] [ -m objectslist ] [ -M sourcepathlist ] [ -N ]  
[ -p processlist ] [ -P { all | pidslist } ] [ -s ] [ -S searchpathlist ] [ -t ] [ -T bufferize ] [ -u ] [ -v ] [ -V  
verbosefilename ] [ -g ] [ -G "start=mmddhhmmssyy,end=mmddhhmmssyy" ] [ -O options ] [ -z ] [ -Z ] | -R } { { -r  
rootstring } } { [ -A { all | cpulist } ] [ -n ] ] [ -r rootstring ] [ -X [ timedata [ , buckets=N ] ] ] } { -x program | -y  
program } } } { -a [ -A [ all ] ] [ -f frequency ] [ -F ] [ -v ] [ -z ] [ -V verbosefilename ] [ -T bufferize ] } { { -r  
rootstring ] -y program } } { { -r rootstring } } }
```

Note:

- All the list type inputs are separated by a comma except for pathlist, which is separated by a colon.
- Multi-cpu profiling mode is automatically disabled while running in real-time mode.
- Microprofiling is automatically disabled if per-processor profiling is turned on.
- Log Buffer size that was specified will be omitted if the **tprof** command runs in realtime mode.
- If the **-x** flag is specified without the **-A** flag, **tprof** runs in realtime mode.
- If the **-x** flag is specified with the **-A** flag, **tprof** runs in automated offline mode.
- If the **-x** flag is omitted **tprof** runs in post-processing mode or manual offline mode, depending on the presence of cooked files and the **-F** flag.
- The **-@** flag is automatically disabled if the **tprof** command runs in a workload partition in real-time or automated-offline modes.
- The **-y** flag can be used only with the **-E** flag or the **-a** flag.
- The **-O showaddrbytes=on** option cannot be used with the **-z** option.
- The **-O wrapfname=on** option should be used with the **-l** option.
- The **-G** option can be used only in post-processing mode.
- The **-O pdetails=on** option can be used only with the **-p** option.
- When manually collecting traces with the **-A** option for the **tprof** post-processing mode, it is mandatory to specify the **-pP** and **I** options of the **trace** command.

Description

The **tprof** command reports processor usage for individual programs and the system as a whole. This command is a useful tool for anyone with a Java™, C, C++, or FORTRAN program that might be processor-bound and who wants to know which sections of the program are most heavily using the processor.

The **tprof** command can charge processor time to object files, processes, threads, subroutines (user mode, kernel mode and shared library) and even to source lines of programs or individual instructions. Charging processor time to subroutines is called profiling and charging processor time to source program lines is called micro-profiling.

For subroutine-level profiling, the **tprof** command can be run without modifying executable programs, that is no recompilation with special compiler flags is necessary. This is still true if the executables have been stripped, unless the traceback tables have also been removed. However, recompilation is required to get a micro-profile, unless a listing file is already available. To perform micro-profiling on a program, either the program should be compiled with the **-g** flag and the source files should be accessible to the **tprof** command or the program should be compiled with the **-qlist** flag and either both the object listing files and the source files or just the object listing files should be accessible to the **tprof** command. To take full advantage of **tprof** micro-profiling capabilities, it is best to provide both the `.lst` and the source file.

The **tprof** command can run in the following modes:

- Realtime or online
- Manual offline
- Automated offline
- Post-processing

If you specify the **-x** flag without the **-A** flag, the **tprof** command runs in realtime mode. In realtime mode, the **tprof** command starts the AIX **trace** utility in the background, and processes the trace data as it gets generated. When the program being profiled ends, **tprof** collects symbolic name information, and generates the **tprof** reports.

Note: This mode does not allow per-processor profiling.

If you specify the **-x** flag with the **-A** flag, the **tprof** command runs in automated offline mode. In this mode, the **tprof** command starts the AIX **trace** utility and logs the trace data into a file. Once the trace data collection is done, it collects symbolic name information, and the **tprof** command opens the trace log file and processes the data to generate reports. In this mode, the **tprof** command generates the following files in addition to the **tprof** report files:

- *rootstring.syms*
- *rootstring.trc* [-cpuid]

All of the input and report files used by the **tprof** command are named *rootstring.suffix*, where *rootstring* is either specified with the **-r** flag, or is the program name specified with the **-x** flag.

In realtime mode and automated offline mode, the *ulimit* value of the data area for the program that is being profiled is set to **unlimited**.

In automated offline mode, you can specify the **-N** flag to collect source line information into the generated **RootString.syms** file. And you can specify the **-I** flag to collect binary instructions into the generated **RootString.syms** file.

The **tprof** command can re-process these files any time to generate profiling reports. This is called manual offline mode. The *rootstring.syms* file contains symbolic name information similar to the output of the **gensyms** command. The *rootstring.trc*[-cpuid] files are trace log files. The **-cpuid** is added to the names when per-processor tracing is on. In that case, each file contains trace data from one processor only.

If you specify the **-c** flag with the **-A** flag, the *rootstring.syms* and *rootstring.trc*[-cpuid] files are not generated. Instead, the following two files are created:

- *rootstring.csyms*
- *rootstring.ctrac* [-cpuid]

Those files are *cooked*, that is they are a pre-processed version of the normal trace and name files. **tprof** post-processes cooked file much faster.

If you specify neither the **-A** flag nor the **-x** flag, the **tprof** command runs either in manual offline or in post-processing mode. To run the **tprof** command in post-processing mode, the following files must be available:

- *rootstring.csyms*
- *rootstring.ctr*[**-cpuid**]

These files are generated when the **tprof** command runs (in any mode except post-processing mode) with the **-c** flag.

To run the **tprof** command in manual offline mode, the following files must be available:

- *rootstring.syms*
- *rootstring.trc* [**-cpuid**]

To generate these files, you need to manually run the **gensyms** command and AIX trace facility, or run the **tprof** command in automated offline mode without the **-c** flag.

The **tprof** command always first looks for *rootstring.csyms* and *rootstring.ctr*[**-cpuid**] files. Only if these files are not available, does it look for the *rootstring.syms* and *rootstring.trc*[**-cpuid**] files. To prevent the **tprof** command from looking for the *rootstring.csyms* and *rootstring.ctr*[**-cpuid**] files, that is, force the manual offline mode, use the **-F** flag.

If the input symbols file contains demangled names, you cannot use the **-Z** flag.

The **tprof** command generates a **tprof** report file named *rootstring.prof*, which holds the process, thread, object file and subroutine level profiling report. The file can contain the following sections and subsections:

- Summary report section:
 - Processor usage summary by process name
 - Processor usage summary by threads (tid)
- Global (pertains to the execution of all processes on system) profile section:
 - Processor usage of user mode routines
 - Processor usage of kernel routines, including milicode routines called in kernel mode
 - Processor usage summary for kernel extensions
 - Processor usage of each kernel extension's subroutines
 - Processor usage summary for privately loaded, global, and named shared libraries, and milicode routines called in user mode
 - Processor usage of each shared library's subroutines
 - Processor usage of each Java class
 - Processor usage of each Java methods of each Java class
- Process and thread level profile sections (one section for each process or thread) :
 - Processor usage of user mode routines for this process/thread
 - Processor usage of kernel routines for this process/thread, including milicode routines called in kernel mode
 - Processor usage summary for kernel extensions for this process/thread
 - Processor usage of each kernel extension's subroutines for this process/thread
 - Processor usage summary for privately loaded, global, and named shared libraries for this process/thread, and milicode routines called in user mode
 - Processor usage of each shared library's subroutines for this process/thread
 - Processor usage of each Java class for this process/thread
 - Processor usage of Java methods of each Java class for this process/thread

The summary report section is always present in the *rootstring.prof* report file. You can turn on or turn off various subsections of the global profile section using the following profiling flags:

- **-u** turns on subsections a
- **-k** turns on subsection b
- **-e** turns on subsections c and d
- **-s** turns on subsections e and f
- **-j** turns on subsections g and h

If you specify the **-p**, **-P** and **-t** flags, the process and thread level profile sections are created for processes and threads. The subsections present within each of the per-process or per-thread sections are identical to the subsections present in the global section, they are selected using the profiling flags (**-u,-s,-k,-e,-j**).

Optionally, if you run the **tprof** command with the **-C** flag, the command also generates per-processor profiling reports, which contains one profiling report per processor. The generated **tprof** reports have the same structure and are named using the convention: *rootstring.prof[-cpuid]*.

If you specify the **-m** flag, the **tprof** command generates micro-profiling reports. The reports use the following naming convention: *rootstring.source.mprof*, where *source* is the base name of a source file. If more than one source file has the same base name, a number to uniquely identify them is appended to the report file names. For example, *rootstring.Filename.c.mprof-1*. The micro-profiling report has the following information:

- The full path name of the annotated source file.
- A hot line profile section which has all the line numbers from that source file hit by profiling samples, sorted by processor usage. For each source line, one line reports the percentage of time spent on behalf of all processes, followed by additional lines with the breakdown by individual process.
- A source line profile section for each of the functions in that source file, which have processor usage. This section contains the source line number, processor usage and source code. If a **.lst** file for that source file is accessible to **tprof**, then it interlaces the instruction lines from the **.lst** file with the source lines from the source file and charges processor usage appropriately. This provides breakdown by instruction for each source file.

If a source file is not present, but a **.lst** file is present, **tprof** only shows the processor usage based on the source lines and the instructions from the **.lst** file.

If neither the **.lst** file nor the source file is present, but the source file is compiled with the **-g** flag, the **tprof** command retrieves the source line numbers and generates a similar report, with the source code column missing.

Note: If per-processor profiling is requested, micro-profiling is automatically disabled. The **tprof** command cannot report correct source line information if a **.c** file is included in another **.c** file. The **tprof** command cannot micro-profile Java classes or methods.

If you specify the **-m** flag, the **-N** flag is automatically specified to gather the source line info into a symbols file in automated offline mode.

If you specify the **-Z** flag with the **-m** flag, one report file is generated per subroutine.

The following naming convention is used: *RootString.source.routine.mprof*, where *routine* is the name of one of the subroutines listed in the source file. In addition, a file named *RootString.source.HOT_LINES.mprof* containing the hot line profiling information described above is also created.

If you specify the **-L** flag, the **tprof** command generates annotated listing files. The files use the following naming convention: *RootString.source.alst*, where *source* is the base name of a source file. If more than one source file has the same base name, a number to uniquely identify them is appended to the report file name. For example, *RootString.Filename.c.alst-1*. If you specify the **-Z** flag with the **-L** flag, one report file is generated per subroutine. The following naming convention is then used: *RootString.source.routine.alst*, where *routine* is the name of one of the subroutines listed in the source file.

If you specify the **-N** flag or **-I** flag when profiling a Java program using JPA (**-x java -Xrunjpa** or **-x java -agentlib:jpa**), the JIT source line number and instructions can be collected if the corresponding parameter is added to the **-Xrunjpa** flag or the **-agentlib:jpa** flag:

- **source=1** turns on JIT source line collecting (requires IBM JRE 1.5.0 or later version).
- **instructions=1** turns on JIT instructions collecting.

The following restrictions apply for non-root users running the **tprof** command:

- The **tprof** will not be able to verify that the running kernel is the same as the **/unix** file. This means that even if a warning message is displayed, in most cases the running kernel and **/unix** are the same, so the data should be accurate.
- When the **gensyms** command is run by a non-root user, the same warning as in restriction #1 (above) is given and the **gensyms** file is marked. If **tprof** is run in the offline mode, the file created with the **gensyms** command will flag **tprof** as to kernel that is not verified.
- The **tprof** will not be able to open and read symbols on files which do not have the read permission set. Some private, shared libraries do not have read permission, and some kernel extensions are not readable.

Time-Based versus Event-Based Profiling

By default, **tprof** is time-based and is driven by the decremter interrupt. Another mode of profiling is event-based profiling, in which the interrupt is driven by either software-based events or by Performance Monitor events. With event-based profiling, both the sampling frequency and the profiling event can be varied on the command line.

The **-E** flag enables event-based profiling. The **-E** flag is one of the four software-based events (EMULATION, ALIGNMENT, ISLBMISS, DSLBMISS) or a Performance Monitor event (PM_*). By default, the profiling event is processor cycles. All Performance Monitor events are prefixed with **PM_**, such as **PM_CYC** for processor cycles or **PM_INST_CMPL** for instructions completed. The **pm1ist** lists all Performance Monitor events that are supported on a processor. The chosen Performance Monitor event must be taken in a group where we can also find the **PM_RUN_INST_CMPL** Performance Monitor event. On POWER4 and later processors, profiling on marked events results in better accuracy. Marked events have the **PM_MRK_** prefix.

If you specify the **-y** flag, only the specified program and its descendents are profiled. Use the **-y** flag only with the **-E** or **-a** flag.

The **-f** flag varies the sampling frequency for event-based profiling. For software-based events and processor cycles, supported frequencies range from 1 to 500 milliseconds, with a default of 10 milliseconds. For all other Performance Monitor events, the range is from 10000 to MAXINT occurrences of the event, with a default of 10000 events. If you specify the **-f** flag with the **-y** flag, the sampling frequency can range from 1 through the MAXINT occurrences for other Performance Monitor events, with a default of 10000 events.

Additional information is added to the **.prof** file to reflect the processor name, profiling event, and sampling frequency.

Java Applications Profiling

To profile Java applications, you must specify the **-j** flag, and start the applications with the **-Xrunjpa** API (for running on Java 5 and earlier JVMs) or the **-agentlib:jpa** (for running on Java 6 JVM) of the **java** command line option. When you specify this option, the JVM will automatically calls the **jpa** library whenever new classes and methods are loaded into memory. The library will in turn collect address to name mapping information for methods and classes in files named **/tmp/JavaPID.syms**, where **PID** is the process ID of a process running a Java Virtual Machine. The **tprof** command will automatically look in that directory for such files.

When running in automated offline mode, or selecting the cooking flags, the **tprof** command will copy the information contained in **JavaPID.syms** files into the **RootString.syms** or **RootString.csyms** file. The corresponding files in **/tmp** can then be deleted. The directory content should be kept up to date by **tprof** command users. Whenever the JVM corresponding to a particular **JavaPID.syms** is stopped, the file should be deleted.

Profile Accuracy

The degree to which processor activity can be resolved is determined by the number of samples captured and the degree to which *hot spots* dominate. While a program with a few hot spots can be profiled with relatively few samples, less-frequently executed sections of the program are not visible in the profiling reports unless more samples are captured. In cases where user programs run less than a minute, there may be insufficient resolution to have a high degree of confidence in the estimates.

A simple solution is to repeatedly execute the user program or script until you achieve the degree of resolution you need. The longer a program is run, the finer the degree of resolution of the profile. If you doubt the accuracy of a profile, run the **tprof** command several times and compare the resulting profiles.

Information

The **-@** flag controls the addition of WPAR information to a **tprof** report. Sub-options specify what information is included to some of the report sections; these sub-options is in one of the following forms:

- The **-@** flag alone (that is, with no suboption) adds a summary of the processor usage WPAR name. Also, the WPAR name is shown for each process listed in the sections summarizing processor usage by process and by thread.
- The **ALL** suboption causes the **tprof** report to contain a process, thread, object file and subroutine-level profiling report for the overall system and for each running WPAR.
- A comma-separated list of WPAR names results in a process, thread, object file and subroutine-level profile section for each named WPAR in the **tprof** report.

Note: When a WPAR is used as a checkpoint and is restarted, some shared library areas might be local to the WPAR. In this case, the name of the WPAR is printed after the name of the area *myarea@mywpar*. In all other cases, the area is system-wide; thus the WPAR name is omitted.

XML Report Generating

The **-X** flag generates an XML report file named **RootString.etm**. This file can be shown in Visual Performance Analyzer. The XML report file contains four sections:

- Profile general information
- Symbol data
- Profile hierarchy
- Temporal data

The **-X** is used in automated offline mode to generate XML report directly.

The **-X** is also used in manual offline mode to generate XML report from the **RootString.syms** and **RootString.trc** files.

If the **-X timedata** is specified, the generated XML report will include the time data information. By default, the time data generating function is turned off.

To specify the bucket number for the time data, use the *buckets=N* argument. The default bucket number is 1800.

Large Page Analysis

The **tprof -a** command collects the profile trace from a representative application run, and produces performance projections. The projections map different portions of the data space of an application to different page sizes. The large page analysis uses the information in the trace to project translation buffer performance when the command maps any of the following application memory regions to a different page size:

- Static application data (data that is initialized or not initialized)
- Application heap (data that is dynamically allocated)
- Stack
- Application text

Performance projections are provided for each of the page sizes that the operating system supports. The first performance projection is a baseline projection that maps all of the memory regions to a default page size of 4 KB. Subsequent projections map one region at a time to a different page size. The following statistics are reported for each projection:

- Page size
- Number of pages needed to back all of the regions
- Translation miss score
- Cold translation miss score

The summary section lists the processes that are profiled and the statistics that are reported. It includes the following information:

- Number or percentage of memory reference
- Modeled memory reference
- Malloc calls
- Free calls

Data Profiling

The **tprof -b** command turns on basic data profiling and collects data access information. The summary section reports access information across the kernel data, library data, user global data, and the stack heap sections for each process.

If you specify the **-b** flag with the **-s**, **-u**, **-k**, and **-e** flags, the **tprof** command data profiling reports most used data structures (exported data symbols) in shared library, binary, kernel and kernel extensions. The **-b** flag also reports the functions that use those data structures.

Comparison of tprof Versus prof and gprof

The most significant differences between these three commands is that **tprof** collects data with no impact on the execution time of the programs being profiled, and works on optimized and stripped binaries without any need for recompilation, except to generate micro-profiling reports. Neither **gprof** nor **prof** have micro-profiling capabilities or work on optimized binaries, while they do require special compilation flags, and induce a slowdown in the execution time that can be significant. **prof** does not work on stripped binaries.

The **prof** and **gprof** tools are standard, supported profiling tools on many UNIX systems, including this operating system. Both **prof** and **gprof** provide subprogram profiling and exact counts of the number of times every subprogram is called. The **gprof** command also provides a very useful *call graph* showing the number of times each subprogram was called by a specific parent and the number of times each subprogram called a child. The **tprof** command provides neither subprogram call counts nor call graph information.

Like the **tprof** command, both the **prof** and **gprof** commands obtain their processor consumption estimates for each subprogram by sampling the program counter of the user program.

tprof collects processor usage information for the whole system, while **prof** and **gprof** collect only profiling information for a single program and only for the time spent in user mode. **tprof** also provides summary for all processes active during the execution of the profiled user program and fully support libraries and kernel mode profiling.

tprof support the profiling of Java applications, which **prof** and **gprof** do not.

Flags

Item	Description
-@ { ALL wparlist }	Includes the WPAR information in the generated reports. The ALL option includes summaries for all of the WPAR. When this option is set, the report contains a 'SYSTEM' report and a report per WPAR traced. The <i>wparlist</i> option specifies a comma-separated list of WPAR. When the <i>wparlist</i> option is set, the tprof command produces a report for each WPAR specified.
-a	Turns on the large page analysis.
-A { all cpulist }	Turns on automatic offline mode. No argument turns off per-processor tracing. all enables tracing of all processors. <i>cpulist</i> is a comma separated list of processor-ids to be traced.
-b	Turns on basic data profiling.
-B	Turns on basic data profiling with the information about the instruction address mapped function.
-c	Turns on generation of cooked files.
-C all cpulist	Turns on the per-processor profiling. Specify all to generate profile reports for all processors. Processor numbers should be separated with a comma if you give a <i>cpulist</i> (for example, 0,1,2). Note: per-processor profiling is possible only if per-processor trace is either on (in automated offline mode), or has been used (in manual offline mode). It is not possible at all in online mode. This option is not supported if the number of CPUs traced is greater than 128.
-d	Turns on deferred tracing mode, that is defers data collection until trcon is called.
-D	Turns on detailed profiling which displays processor usage by instruction offset under each subroutine.
-e	Turns on kernel extension profiling.
-E [mode]	Enables event-based profiling. The possible modes are: PM_event Specifies the hardware event to profile. If no mode is specified for the -E flag, the default event is processor cycles (PM_CYC). EMULATION Enables the emulation profiling mode. ALIGNMENT Enables the alignment profiling mode. ISLBMISS Enables the Instruction Segment Lookaside Buffer miss profiling mode. DSLBMIS Enables the Data Segment Lookaside Buffer miss profiling mode.

Item	Description
-f <i>frequency</i>	Specifies the sampling frequency. The sampling frequency can be from 1 to 500 milliseconds for processor cycles and EMULATION , ALIGNMENT , ISLBMISS , and DSLBMIS events, and from 10000 to MAXINT event occurrences for other Performance Monitor events. If you specify the -f flag with the -y flag, the value of the sampling frequency ranges from 1 through the value of the MAXINT occurrences for other Performance Monitor events, with the default value of 10000 events.
-F	Overwrites cooked files if they exists. If used without the -x flag, this forces the manual offline mode.
-g	Does not translate symbol names into human-readable names.
-G	Sets trace processing start date and end date. The parameters are specified in the following format: <pre>"start=mmddhhmmssyy,end=mmddhhmmssyy"</pre> where mmddhhmmssyy is the month, day, hour, minute, second, and year respectively. This option can have the following values: start When set, trace processing starts from the specified start date string. end When set, trace processing stops at the specified end date string.
-I	Turns on binary instructions collecting. Note: The -I flag activates to gather binary instructions when generating symbol files or cooked symbol files in automated offline mode. However, in manual offline mode, the -I flag does not affect the report files.
-j	Turns on Java classes and methods profiling.
-k	Enables kernel profiling.
-l	Enables long names reporting. By default tprof truncates the subroutine, program and source file names if they do not fit into the available space in the profiling report. This flag disables truncation.
-L <i>objectlist</i>	Enables listing annotation for objects specified by the comma separated list, <i>objectlist</i> . Executables and shared libraries can have their listing files annotated. Specify the archive name for libraries. Note: 1. To enable listing annotation of programs, user mode profiling (-u) must be turned on. 2. To enable listing annotation of shared libraries, shared library profiling (-s) must be turned on. 3. To annotate a listing generated with IPA compilations, specify a .lst as the <i>objectlist</i>.

Item	Description
-m <i>objectlist</i>	Enables micro-profiling of objects specified by the comma separated list, <i>objectlist</i>. Executables, shared libraries, and kernel extensions can be micro-profiled. Specify the archive name for libraries and kernel extensions. Note: 1. To enable micro-profiling of programs, user mode profiling (-u) must be turned on. 2. To enable micro-profiling of shared libraries, shared library profiling (-s) must be turned on. 3. To enable micro-profiling of kernel extensions, kernel extension profiling (-e) must be turned on.
-M <i>PathList</i>	Specifies the source path list. The <i>PathList</i> is a colon separated list of paths that are searched for source files and .lst files that are required for micro-profiling and listing annotation. By default the source path list is the object search path list.
-n	Turns off postprocessing. If the -n flag is specified, the -u, -s, -k, -e, and -j flags are ignored. The data is collected, the .trc file and the gensyms files are generated, but the .prof file is not generated. This helps avoid overloading the system during a benchmark, for example. The -A flag must be used if the -n option is used.
-N	Turns on source line number info collecting. The -N flag activates to gather source line information when generating symbol files or cooked symbol files in automated offline mode. However, in manual offline mode, the -N flag does not affect the report files.

Item**Description****-O**

This option can have the following values:

showaddrbytes=[on|off]

Turns on the Address and Bytes columns in subroutine reports. The default value is off.

wrapfname=[on|off]

Turns on the line wrap of the long function name. To wrap the function names on a line, set value as -l. The default value is off.

pdetails=[on|off]

Turns on the data consolidation process for the report. The report consolidates data for the specified processlist in the **kernel** and **sharedlib** segment of the Process Summary section in the report.

>|

threshold_cmp_val=value

Sets a threshold-compare floating point value.

The Power processor has a threshold counter facility. The threshold counter is different from the primary performance monitor counters (PMCs) because the threshold counter can count the number of events that occur between a separate set of designated start events and end events from the core, cache, and memory subsystems.

This option specifies a threshold value that you can use to compare against the number of events between the designated start events and end events.

threshold_event_sel=name

Specifies the events that must be considered for threshold counting. Valid values for the **threshold_event_sel** option are as follows:

THRESHOLD_COUNT_CYCLES

Counts the number of events that occur or cycles on which the control register (CTRL) run latch has been set. The AIX operating system uses the control register (CTRL) bit to indicate the idle or run state. The performance monitoring unit (PMU) also use this bit to avoid counting events during idle periods. This bit is commonly called the *run latch*. This option does not depend on freeze conditions. This means that the events are counted even when the PMU has frozen the contents of event counters.

THRESHOLD_COUNT_NUM_COMP_INSTR

Counts the number of completed instructions when the CTRL run latch is set.

This option does not depend on freeze conditions. This means that the events are counted even when the PMU has frozen the contents of event counters.

THRESHOLD_COUNT_PMC1_EVENTS

Counts the PMC1 events.

PMC1 - PMC4 are 32-bit registers that are called programmable counters because the events that can be counted are specified by the program. This option depends on freeze conditions. This means that the events are not counted when the PMU has frozen the contents of event counters.

|<

Item**Description****-O** (continued)**THRESHOLD_COUNT_PMC2_EVENTS**

Counts the PMC2 events. This option depends on freeze conditions. This means that the events are not counted when the PMU has frozen the contents of event counters.

THRESHOLD_COUNT_PMC3_EVENTS

Counts the PMC3 events. This option depends on freeze conditions. This means that the events are not counted when the PMU has frozen the contents of event counters.

THRESHOLD_COUNT_PMC4_EVENTS

Counts the PMC4 events. This option depends on freeze conditions. This means that the events are not counted when the PMU has frozen the contents of event counters.

threshold_eve_start_sel=name

Specifies the event to start the threshold counting. For valid values of this option, see [threshold_eve_stop_sel](#).

>**-O** (continued)<**threshold_eve_stop_sel=name**

Specifies the event to stop the threshold counting. You can set the following values for the **threshold_eve_start_sel** and **threshold_eve_stop_sel** options:

PM_MRK_INST_DECODED

Sampled instructions are decoded.

PM_MRK_INST_DISP

Sampled instructions are dispatched.

PM_MRK_INST_ISSUED

Sampled instructions are issued.

PM_MRK_INST_FIN

Sampled instructions are finished.

PM_MRK_INST_CMPL

Sampled instructions are completed.

PM_MRK_LD_MISS_L1

Sampled instruction level 1 (L1) load cache is missed.

PM_MRK_L1_RELOAD_VALID

Sampled instruction level 1 (L1) reload is valid.

EVE_SEL_PMC1

Event that is selected in monitor mode control register (MMCR) 1 for PMC1 occurred.

EVE_SEL_PMC2

Event that is selected in MMCR1 for PMC2 occurred.

EVE_SEL_PMC3

Event that is selected in MMCR1 for PMC3 occurred.

EVE_SEL_PMC4

Event that is selected in MMCR1 for PMC4 occurred.

PM_MRK_NTF_FIN

Sampled instruction is in the next to finish state.

PM_MRK_L2_RC_DISP

RC machine dispatched for sampled instruction.

PM_MRK_ST_DONE_L2

RC machine done for sampled instruction.

Item

>|**-o** (continued)|<

Description**random_samp_ele_crit=name**

Specifies the random criteria for selecting the instructions for sampling. Valid values for this option are as follows:

ALL_INSTR

All instructions are eligible. This value is the default setting.

LOAD_STORE

The operation is routed to the Load Store Unit (LSU); for example, load, store.

PROB_NOP

Sample only special no-operation instructions, which are called Probe NOP events.

LARX_STCX

Sample only load and reserve indexed (LARX) instructions and store conditional indexed (STCX) instructions.

LOAD_SAMPLING

Sample only load instructions.

LONG_LATENCY_OP

Sample only long latency instructions (div/sqrt/mul).

STORE_SAMPLING

Sample only store instructions.

LOAD_MISSES

Sample only load-miss instructions.

LOAD_HIT_STORE

Sample only load-hit-store instructions. This value is applicable only for cases where a load operation can forward data from the store queue to a finish state. This value is not applicable to cases where a load-hit-store instruction is in the store queue but the load operation partially overlaps with the store.

BRANCH_MISPREDICTS

Sample only branch-mispredict instructions.

BRANCH_MISPREDICTS_DIR_CR_CTR

Sample only branch-mispredict events (Direction, CR, or CTR).

BRANCH_MISPREDICTS_TA

Sample only branch-mispredict events (TA).

BRANCHES_RESOLVED_TAKEN

Sample branches with resolution.

NON_REPEATING_BRANCHES

Sample only non-repeating branches.

ALL_BRANCHES_REQ_PRED

Sample all branches that require prediction.

group_num=event_group

Specifies a group number that must be used for the **tprof** command. This option is an optional attribute. If you specify this value, the group must have a threshold event and the same event must be used with the **-E** flag.

Item	Description
-p <i>processlist</i>	Enables process level profiling of the process names specified in the <i>processlist</i>. <i>processlist</i> is a comma separated list of process names Process level profiling is enabled only if at least one of the profiling modes (-u, -s, -k, -e, or -j) is turned on.
-P { all <i>PIDList</i> }	Enables process level profiling of all processes encountered or for processes specified with <i>PIDList</i>. The <i>PIDList</i> is a comma separated list of process-IDs. Process level profiling is enabled only if at least one of the profiling modes (-u, -s, -k, -e, or -j) is turned on.
-r <i>rootstring</i>	Specifies the <i>rootstring</i>. tprof input and report files all have names in the form of <i>rootstring.suffix</i>. If you do not specify the -r flag, the <i>rootstring</i> parameter uses the default program name that the -x flag specifies.
-R	Specifies that the tprof command should use samples weighted by PURR increment values to calculate percentages. This is the preferred mode when running in either simultaneous multithreading or Micro-Partitioning environments. The -R flag cannot be used with either the -z flag or the -Z flag.
-s	Enables shared library profiling.
-S <i>PathList</i>	Specifies the object search <i>PathList</i>. The <i>PathList</i> is a colon separated list of paths that are searched for executables, shared libraries and kernel extensions. The default object search <i>PathList</i> is the environment path list (\$PATH).
-t	Enables thread level profiling. If -p or -P are not specified with the -t flag, -t is equivalent to -P all -t. Otherwise, it enables thread level reporting for the selected processes. Thread level profiling is enabled only if at least one of the profiling modes (-u, -s, -k, -e, -j) is enabled.
-T <i>buffersize</i>	Specifies the trace <i>buffersize</i>. This flag has meaning only in real time or automated offline modes.
-u	Enables user mode profiling.
-v	Enables verbose mode.
-V <i>File</i>	Stores the verbose output in the specified <i>File</i> .
-x <i>program</i>	Specifies the program to be executed by tprof. Data collection stops when <i>program</i> completes or trace is manually stopped with either trcoff or trcstop The -x flag must be the last flag in the list of flags specified in tprof.
-X	Specifies the tprof command to call XML Generator when the tprof profiling is finished, and to generate the XML report directly from the tprof trace and symlib data. The -X option needs Java. Install the Java first, and make sure Java is in PATH.

Item	Description
-y	Turns on the event-based profiling for only the specified command and its descendents.
-z	Turns on ticks report. Enables compatibility mode with the previous version of tprof. By default processor usage is only reported in percentages. When -z is used, tprof also reports ticks. This flag also adds the Address and Bytes columns in subroutine reports. If you specify the -z flag with the -a flag, the process summary section in the report displays numbers rather than percentages.
-Z	Switches reports to use ticks instead of percentages (same as the -z flag), and splits annotated listing (when used with the -L flag) and annotated source files (when used with the -m flag) into multiple files, one per subroutine. This option turns on the -g flag.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in *Security*. For a list of privileges and the authorizations associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. The following example shows the basic global program and thread-level summary:

```
$tprof -x sleep 10
```

An output that is similar to the following is displayed:

```
Mon May 21 00:39:26 2012 System: AIX 6.1 Node: dreaming Machine: 000671894C00
Starting Command sleep 10
stopping trace collection.
Generating sleep.prof
```

The **sleep.prof** file that is generated only contains the summary report section.

2. The following example shows the global profiling with all options:

```
$tprof -skeuj -x sleep 10
```

An output that is similar to the following is displayed:

```
Mon May 21 00:39:26 2012
System: AIX 6.1 Node: dreaming Machine: 000671894C00
Starting Command sleep 10
stopping trace collection.
Generating sleep.prof
```

The **sleep.prof** file that is generated contains the summary report and global profile sections.

3. The following example shows the single process level profiling:

```
$tprof -u -p workload -x workload
```

An output that is similar to the following is displayed:

```
Mon May 21 00:39:26 2012
System: AIX 6.1 Node: dreaming Machine: 000671894C00
```

```
ming Machine: 000671894C00
Starting Command workload stopping trace collection.
Generating workload.prof
```

The **workload.prof** file that is generated contains the summary report, the global user mode profile sections, and one process level profile section for the process 'workload' that contains only a user mode profile subsection.

4. The following example shows the multiple process level profiling:

```
$tprof -se -p send,receive -x startall
```

An output that is similar to the following is displayed:

```
Mon May 21 00:39:26 2012
System: AIX 6.1 Node: drea
ming Machine: 000671894C00
Starting Command startall
stopping trace collection.
Generating startall.prof
```

The **startall.prof** file that is generated contains the summary report, the global shared library mode profile, the global kernel extension profile sections, and two process level profile sections: one for the process 'send', and one for the process 'receive'. The process level sections each contain two subsections: one with shared library profiling information and one with kernel extensions profiling information.

5. The following example shows the micro-profiling and listing annotation:

```
$tprof -m ./tcalc -L ./tcalc -u -x ./tcalc
```

An output that is similar to the following is displayed:

```
Mon May 21 00:39:26 2012
System: AIX 6.1 Node: drea
ming Machine: 000671894C00
Starting Command ./tcalc
stopping trace collection.
Generating tcalc.prof
Generating tcalc.tcalc.c.mprof
Generating tcalc.tcalc.c.alst
```

The **tcalc.prof** file that is generated contains the summary report and the global user mode profile sections. The resulting **tcalc.tcalc.c.mprof** and **tcalc.tcalc.c.alst** files contain the micro-profiling report and the annotated listing.

6. For event-based profiling on processor cycles, sampling once every 100 milliseconds, enter the following command:

```
$tprof -E -f 100 -Askex sleep 10
```

The output is similar to the following display:

```
Starting Command sleep 10
stopping trace collection.
Tue Apr 26 14:44:02 2005
System: AIX 5.3 Node: bigdomino Machine: 00C0046A4C00
Generating sleep.trc
Generating sleep.prof
Generating sleep.syms
```

7. For event-based profiling on completed instructions, sampling once every 20,000 completed instructions, enter the following command:

```
$tprof -E PM_INST_CMPL -f 20000 -Askex sleep 10
```

The output is similar to the following display:

```
Starting Command sleep 10
stopping trace collection.
Tue Apr 26 14:42:44 2005
System: AIX 5.3 Node: bigdomino Machine: 00C0046A4C00
Generating sleep.trc
Generating sleep.profp
Generating sleep.syms
```

8. For event-based profiling on emulation interrupts, sampling once every 10000 events, enter the following command:

```
$tprof -E EMULATION -Askex sleep 10
```

The output is similar to the following display:

```
Starting Command sleep 10
stopping trace collection.
Tue Apr 26 14:41:44 2005
System: AIX 5.3 Node: bigdomino Machine: 00C0046A4C00
Generating sleep.trc
Generating sleep.profp
Generating sleep.syms
```

9. The following example shows the automated offline mode:

```
$tprof -c -A all -x sleep 10
```

The output is similar to the following display:

```
Starting Command sleep 10
stopping trace collection.
Mon May 21 00:39:26 2012
System: AIX 6.1 Node: drea
ming Machine: 000671894C00
Generating sleep.ctrc
Generating sleep.csyms
Generating sleep.profp
```

The **sleep.profp** file that is generated only has a summary report section, while the two cooked files are ready to be re-postprocessed.

10. The following example shows the automated offline mode that is enabling source line collecting:

```
$tprof -A -N -x sleep 10
```

The output is similar to the following display:

```
Starting Command sleep 10
stopping trace collection.
Wed Feb 8 15:12:41 2006
System: AIX 5.3 Node: aixperformance Machine: 000F9F3D4C00
Generating sleep.trc
Generating sleep.profp
Generating sleep.syms
```

The **sleep.profp** file that is generated only contains the summary report section, while **sleep.syms** contains the source line information.

11. The following example shows the automated offline mode that is enabling source line and instruction collecting:

```
$tprof -A -N -I -r RootString -x sleep 10
```

The output is similar to the following display:

```
Starting Command sleep 10
stopping trace collection.
Wed Feb 8 15:16:37 2006
System: AIX 5.3 Node: aixperformance Machine: 000F9F3D4C00
```



```
Generating RootString.trc
Generating RootString.prof
Generating RootString.syms
```

The **rootstring.prof** file is generated. The **rootstring.syms** file contains the source line information and binary instructions.

12. To enable Java source line and instructions collecting for the application HelloAIX that is running on Java 5 JVM in realtime mode, enter the following command:

```
$tprof -N -I -x java -Xrunjpa:source=1,instructions=1 Hello AIX
```

The output is similar to the following display:

```
Thu Feb  9 13:30:38 2006
System: AIX 5.3 Node: perfdev Machine: 00CEBB4A4C00
Starting Command java -Xrunvnp_jpa:source=1,instructions=1 Hello AIX
Hello AIX!
stopping trace collection.
Generating java.prof
```

The **java.prof** file is generated. It contains the JIT source line information and the JIT instructions.

13. The following example shows the processor usage for the **vloop_lib_32** program without any shared library, thread-level profiling, per-processor tracing, or post processing:

```
$tprof -A -n -s -t -r test -x vloop_lib_32 5
```

The output is similar to the following display:

```
Starting Command vloop_lib_32 5
stopping trace collection.
Generating test.trc
Generating test.syms
```

14. The following is an example of the automated offline mode for XML report:

```
$tprof -A -X -r RootString -x sleep 10
Starting Command sleep 10
stopping trace collection.
Tue Apr 17 22:00:24 2007
System: AIX 5.3 Node: test105 Machine: 00CEBB4A4C00
Generating sleep.trc
Generating sleep.syms
Calling tprof2xml to generate XML report.
tprof2xml TraceReader Version 1.2.0
Tue Apr 17 22:00:24 2007
System: AIX 6.1 Node: test105 Machine: 00CEBB4A4C00
-----0-----
Record 0
Post-processing counters
Retrieving Disassembly
writing the XML
Writing symbol list
.
Writing process hierarchy
Finished writing sleep.etm
```

15. The following is an example of the automated offline mode enabling source line and instruction collecting:

```
$tprof -A -N -I -X -x sleep 10
Starting Command sleep 10
stopping trace collection.
Tue Apr 17 22:00:24 2007
System: AIX 5.3 Node: test105 Machine: 00CEBB4A4C00
Generating sleep.trc
Generating sleep.syms
Calling tprof2xml to generate XML report.
tprof2xml TraceReader Version 1.2.0
Tue Apr 17 22:00:24 2007
System: AIX 6.1 Node: test105 Machine: 00CEBB4A4C00
-----0-----
Record 0
```

```

Post-processing counters
Retrieving Disassembly
writing the XML
Writing symbol list
.
Writing process hierarchy
Finished writing sleep.etm
The symbol data elements in the xml report will have both bytes and
LineNumberList child elements.

```

16. The following is an example of the automated offline mode for XML report enabling timedata:

```

$tprof -A -X timedata,buckets=100 -x sleep 10
Starting Command sleep 10
stopping trace collection.
Tue Apr 17 22:18:06 2007
System: AIX 5.3 Node: test105 Machine: 00CEBB4A4C00
Generating RootString.trc
Generating RootString.syms
Calling tprof2xml to generate XML report.
tprof2xml TraceReader Version 1.2.0
Tue Apr 17 22:18:06 2007
System: AIX 5.3 Node: test105 Machine: 00CEBB4A4C00
Tue Apr 17 22:18:06 2007
System: AIX 5.3 Node: test105 Machine: 00CEBB4A4C00
-----0-----
Record 0
Post-processing counters
Retrieving Disassembly
writing the XML
Writing symbol list
.
Writing process hierarchy
Finished writing RootString.etm
The RootString.etm will have bucket elements in each object of the profile
hierachy.

```

17. The following is an example of the manual offline mode for XML report:

```

$tprof -A -x sleep 10
Starting Command sleep 10
stopping trace collection.
Tue Apr 17 22:28:01 2007
System: AIX 5.3 Node: test105 Machine: 00CEBB4A4C00
Generating sleep.trc
Generating sleep.prof
Generating sleep.syms

```

To run the **tprof** to use the **sleep.trc** and **sleep.syms** to generate XML report, enter the following to specify the **-r sleep** to generate XML report:

```

$tprof -X -r sleep
Calling tprof2xml to generate XML report.
tprof2xml TraceReader Version 1.2.0
Tue Apr 17 22:28:01 2007
System: AIX 6.1 Node: test105 Machine: 00CEBB4A4C00
-----0-----
Record 0
Post-processing counters
Retrieving Disassembly
writing the XML
Writing symbol list
.
Writing process hierarchy
Finished writing sleep.etm

```

18. For large page analysis of the workload and its descendants, enter the following command:

```

$tprof -a -y workload

```

The output is similar to the following display:

```

Starting Command workload
stopping trace collection.
Tue Apr 26 14:42:44 2005
System: AIX 5.3 Node: bigdomino Machine: 00C0046A4C00

```

```
Generating workload.trc
Generating workload.prof
Generating workload.syms
```

19. To profile only the specified program workload and its descendents, enter the following command:

```
$tprof -E PM_MRK_LSU_FIN -f 20000 -Aske -y workload
```

The output is similar to the following display:

```
Starting Command workload
stopping trace collection.
Tue Apr 26 16:42:44 2005
System: AIX 5.3 Node: bigdomino Machine: 00C0046A4C00
Generating workload.trc
Generating workload.prof
Generating workload.syms
```

20. To enable Java source line and instructions collecting for the application HelloAIX that is running on Java 6 JVM in real time mode, enter the following command:

```
$ tprof -N -I -x java -agentlib:jpa=source=1,instructions=1 Hello AIX
```

Note: When a 64-bit JDK is used, enter the **-agentlib:jpa64** command instead of **-agentlib:jpa** in the following format:

```
$ tprof -N -I -x java -agentlib:jpa64=source=1,instructions=1 Hello AIX
```

The output is similar to the following display:

```
Fri May 30 04:16:27 2008
System: AIX 6.1 Node: toolbox2 Machine: 00CBA6FE4C00
Starting Command java -agentlib:jpa=source=1,instructions=1 Hello AIX
Hello AIX!
stopping trace collection.
Generating java.prof
```

The **java.prof** file is generated. It contains the JIT source line information and JIT instructions.

21. To displays the address bytes information in the report by using the **-O showaddrbytes=on** flag, enter the following command:

```
$ tprof -O showaddrbytes=on -x sleep 5
```

A report similar to the following example is displayed:

Subroutine	%	Source	Address	Bytes
=====	=====	=====	=====	=====
h_cede_end_point	98.47	hcalls.s	111bfc	14

Sample report without -O showaddrbytes=on option

Subroutine	%	Source
=====	=====	=====
h_cede_end_point	98.47	hcalls.s

22. To display the process for trace data between 02/18/2016 02:30:30 and 02/18/2016 02:35:30 by using the **-G** option, enter the following command:

```
$tprof -G "start=021802303016,end=021802353016" -r sleep
```

To process trace data starting from 02/18/2016 02:30:30 till the end, enter the following command:

```
$tprof -G "start= 021802303016" -r sleep
```

To process trace data from start and until 02/18/2016 02:35:30, enter the following command:

```
$tprof -G "end=021802303517" -r sleep
```

23. In the following example, the function name is

```
Test::abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ123456789
```

. To display how to line wrap long function names by using the **-O wrapfname=on** option, enter the following command:

```
$tprof -ukesl -O wrapfname=on -x sleep 5

The following is a sample report:
.Test::abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVW
XYZ123456789                                215  19.40 test. C
```

The following is a sample report without using the **-O wrapfname=on** option:

```
Test::abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789_0abcdefghijklmnopqrstuvwxyz
```

```
lmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890(int,int) 215  19.40 test. C
```

24. [▶](#) To count the number of threshold count events elapsed between a distinct pair of threshold start events and end events and to specify the threshold value for comparison, enter the following command:

```
$tprof -E PM_THRESH_MET -O threshold_cmp_val=10000
-O threshold_event_sel=THRESHOLD_COUNT_CYCLES -O
threshold_eve_start_sel=PM_MRK_L1_RELOAD_VALID -O threshold_eve_stop_sel=PM_MRK_ST_DONE_L2
-O random_samp_ele_crit=ALL_INSTR -O group_num=607 -x sleep 5
```

[|<](#)

Messages

If your system displays the following message:

```
/dev/systrace: device busy or trcon: TRCON:no such device
```

This means the **trace** facility is already in use. Stop your program and try again after typing **trcstop**, stops the trace.

tput Command

Purpose

Queries the **terminfo** database for terminal-dependent information.

Syntax

For Outputting Terminal Information

```
tput [ -T Type ] [ CapabilityName {clear, init, longname, reset} [ Parameters... ] ]
```

For Using stdin to Process Multiple Capabilities

```
tput [ -S ]
```

Description

The **tput** command uses the **terminfo** database to make terminal-dependent information available to the shell. The **tput** command outputs a string if the attribute *CapabilityName* is of type *string*. The output

string is an integer if the attribute is of type *integer*. If the attribute is of type *Boolean*, the **tput** command sets the exit value (0 for TRUE, 1 for FALSE), and produces no other output.

XTERM DESCRIPTION LIMITATION

The xterm terminal description in the DEC.TI file on AIX Version 4 provides underline mode by using the SGR attribute. The SMUL and RMUL attributes are not currently defined in the XTERM terminal description on AIX Version 4. Use the more generic capability named SGR.

```
tput sgr x y
```

Where *x* is either a 1 or a 0 to turn standout mode on or off respectively, and *y* is either a 1 or a 0 to turn underline mode on or off respectively. See the article "**terminfo** file format" for more details on the SGR capability.

```
tput sgr 0 1      turn off standout; turn on underline
tput sgr 0 0      turn off standout; turn off underline
tput sgr 1 1      turn on standout; turn on underline
tput sgr 1 0      turn on standout; turn off underline
```

Flags

In addition to the capability names, the following strings are supported as arguments to the **tput** subroutine.

Item	Description
clear	Displays the clear screen sequence (this is also a capability name).
init	Displays the sequence that initializes the user's terminal in an implementation-dependent manner.
reset	Displays the sequence that will reset the user's terminal in an implementation-dependent manner.
longname	Displays the long name and the specified terminal (or current terminal if none specified).
-S	Uses stdin. This allow the tput to process multiple capabilities. When using the -S option, the capabilities cannot be entered on the command line. Enter ^D token finished.
-TType	Indicates the type of terminal. If -T is not specified, the TERM environment variable is used for the terminal.

Exit Status

This command returns the following exit values:

Item	Description
0	The requested string was written successfully.
1	Unspecified.
2	Usage error.
3	No information is available about the specified terminal type.
4	The specified operand is invalid.
>4	An error occurred.

Examples

1. To clear the screen for the current terminal, enter:

```
tput clear
```

2. To display the number of columns for the current terminals, enter:

```
tput cols
```

3. To display the number of columns for the aixterm terminal, enter:

```
tput -T aixterm cols
```

4. To set the shell variable **bold** to the begin standout mode sequence and the shell variable **offbold** to the end standout mode sequence, enter:

```
bold=`tput smso`  
offbold=`tput rmso`
```

Entering these commands might be followed by the following prompt:

```
echo "${bold}Name: ${offbold} \c"
```

5. To set the exit value to indicate if the current terminal is a hardcopy terminal, enter:

```
tput hc
```

6. To initialize the current terminal, enter:

```
tput init
```

Files

Item	Description
/usr/share/lib/terminfo/?/*	Contains the terminal descriptor files.
/usr/include/term.h	Contains the definition files.

tr Command

Purpose

Transforms characters or range of characters.

Syntax

To transform characters or sequence of characters:

```
tr [ -c | -cds | -cs | -C | -Cds | -Cs | -ds | -s ] [ -A ] String1 String2
```

To delete characters or sequence of characters:

```
tr { -cd | -cs | -Cd | -Cs | -d | -s } [ -A ] String1
```

> | **To define a range of characters for a specific locale:** | <

> | [LANG=*ll_RR*] **tr** -L {*c1-cn C1-Cn*} {*c1c2c3c4c5c6...cn C1C2C3C4C5C6...Cn*} | <

> | **To delete a range of characters that is specified by the user:** | <

> | [LANG=*ll_RR*] **tr** -L {*c1-cn C1-Cn*} | <

Description

The **tr** command deletes or substitutes characters from standard input and writes the result to standard output. The **tr** command also defines a range of characters for a specific locale. The **tr** command performs the following types of operations depending on the strings specified by the *String1* and *String2* variable and depending on the flags specified by the user.

Transforming Characters

If *String1* and *String2* are both specified and the **-d** flag is not specified, the **tr** command replaces each character contained in *String1* from the standard input with the character in the same position in *String2*.

Deleting Characters Using the -d Flag

If the **-d** flag is specified, the **tr** command deletes each character contained in *String1* from standard input.

Removing Sequences Using the -s Flag

If the **-s** flag is specified, the **tr** command removes all but the first character in any sequence of a character string represented in *String1* or *String2*. For each character represented in *String1*, the **tr** command removes all but the first occurrence of the character from standard output. For each character represented in *String2*, the **tr** command removes all but the first occurrence in a sequence of occurrences of that character in the standard output.

>|Defining a character range in the current locale environment|<

>|You can specify a character range for specific locales, code sets, and users by using the **-L** flag of the **tr** command. The **tr** command retains and recognizes the character range that you specified. The **tr** command uses this information to transform the characters to the mapped characters in the range. The character range *c1-cn* that is specified with the **-L** flag is mapped to characters *c1c2c3c4c5c6 . . . cn*, and the specified character range *C1-Cn* is mapped to characters *C1C2C3C4C5C6 . . . Cn*.|<

Special Sequences for Expressing Strings

The strings contained in the *String1* and *String2* variables can be expressed using the following conventions:

Item	Description
<i>C1-C2</i>	Specifies the string of characters that collate between the character specified by <i>C1</i> and the character specified by <i>C2</i> , inclusive. The character specified by <i>C1</i> must collate before the character specified by <i>C2</i> . Note: The current locale has a significant effect on results when specifying subranges using this method. If the command is required to give consistent results irrespective of locale, the use of subranges should be avoided.
<i>[C*Number]</i>	<i>Number</i> is an integer that specifies the number of repetitions of the character specified by <i>C</i> . <i>Number</i> is considered a decimal integer unless the first digit is a 0; then it is considered an octal integer.
<i>[C*]</i>	Fills out the string with the character specified by <i>C</i> . This option, used only at the end of the string contained within <i>String2</i> , forces the string within <i>String2</i> to have the same number of characters as the string specified by the <i>String1</i> variable. Any characters specified after the * (asterisk) are ignored.

Item**Description****[:ClassName:]**

Specifies all of the characters in the character class named by *ClassName* in the current locale. The class name can be any of the following names:

alnum	lower
alpha	print
blank	punct
cntrl	space
digit	upper
graph	xdigit

Except for **[:lower:]** and **[:upper:]** conversion character classes, the characters specified by other character classes are placed in an array in an unspecified order. Because the order of the characters specified by character classes is undefined, the characters should be used only if the intent is to map several characters into one. An exception to this is the case of conversion character classes.

For more information on character classes, see the **ctype** subroutines.

[=C=]

Specifies all of the characters with the same equivalence class as the character specified by *C*.

\Octal

Specifies the character whose encoding is represented by the octal value specified by *Octal*. An Octal value can be a one digit, two digit, or three digit octal integer. The NULL character can be expressed by using the '\0' expression, and is processed like any other character.

\ControlCharacter

Specifies the control character that corresponds to the value specified by *ControlCharacter*. The following values can be represented:

\a	Alert
\b	Backspace
\f	Form-feed
\n	New line
\r	Carriage return
\t	Tab
\v	Vertical tab

Specifies the \ (backslash) as itself, without any special meaning as an escape character.

\[

Specifies the [(left bracket) as itself, without any special meaning as the beginning of a special string sequence.

\-

Specifies the - (minus sign) as itself, without any special meaning as a range separator.

If a character is specified more than once in *String1*, the character is transformed into the character in *String2* that corresponds to the last occurrence of the character in *String1*.

If the strings specified by *String1* and *String2* are not the same length, the **tr** command ignores the extra characters in the longer string.

Flags

Item	Description
-A	Performs all operations on a byte-by-byte basis using the ASCII collation order for ranges and character classes, instead of the collation order for the current locale.
-C	Specifies that the value of <i>String1</i> be replaced by the <i>complement</i> of the string specified by <i>String1</i>. The complement of <i>String1</i> is all of the characters in the character set of the current locale, <i>except</i> the characters specified by <i>String1</i>. If the -A and -c flags are both specified, characters are complemented with respect to the set of all 8-bit character codes. If the -c and -s flags are both specified, the -s flag applies to characters in the complement of <i>String1</i>. If the -d option is not specified, the complements of the characters specified by <i>String1</i> will be placed in the array in ascending collation sequence as defined by the current setting of LC_COLLATE.
-c	Specifies that the value of <i>String1</i> be replaced by the <i>complement</i> of the string specified by <i>String1</i>. The complement of <i>String1</i> is all of the characters in the character set of the current locale, <i>except</i> the characters specified by <i>String1</i>. If the -A and -c flags are both specified, characters are complemented with respect to the set of all 8-bit character codes. If the -c and -s flags are both specified, the -s flag applies to characters in the complement of <i>String1</i>. If the -d option is not specified, the complement of the values specified by <i>String1</i> will be placed in the array in ascending order by binary value.
-d	Deletes each character from standard input that is contained in the string specified by <i>String1</i>. Note: 1. When the -C option is specified with the -d option, all characters except those specified by <i>String1</i> will be deleted. The contents of <i>String2</i> are ignored unless the -s option is also specified. 2. When the -c option is specified with the -d option, all values except those specified by <i>String1</i> will be deleted. The contents of <i>String2</i> are ignored unless the -s option is also specified.
> > -L	Adds a user-defined character range in the current locale environment to the <code>\$HOME/.trregexc/\$CODESET</code> file. The character range <code>c1-cn</code> is mapped to <code>c1c2c3c4c5c6...cn</code>, and the character range <code>C1-Cn</code> is mapped to <code>C1C2C3C4C5C6...Cn</code>. The -L flag is user-specific (depends on the <code>\$HOME</code> variable), code-set specific, and locale-specific (depends on the <code>\$LANG</code> variable). It means that you must define the character range for specific users, code sets, and locales. If the <code>\$HOME/.trregexc/\$CODESET</code> file does not exist for a specific user or locale, the file is automatically generated when you specify the -L flag. < <
-s	Removes all but the first character in a sequence of repeated characters. Character sequences specified by <i>String1</i> are removed from standard input before translation, and character sequences specified by <i>String2</i> are removed from standard output.
<i>String1</i>	Specifies a string of characters.
<i>String2</i>	Specifies a string of characters.

Exit Status

This command returns the following exit values:

0

All input was processed successfully.

>0

An error occurred.

Examples

1. To transform braces into parentheses, enter the following command:

```
tr '{}' '()' < textfile > newfile
```

This transforms each { (left brace) to ((left parenthesis) and each } (right brace) to) (right parenthesis). All other characters remain unchanged.

2. To transform braces into brackets, enter the following command:

```
tr '{}' '\[]' < textfile > newfile
```

This transforms each { (left brace) to [(left bracket) and each } (right brace) to] (right bracket). The left bracket must be entered with a \ (backslash) escape character.

3. To transform lowercase characters to uppercase, enter the following command:

```
tr 'a-z' 'A-Z' < textfile > newfile
```

4. To create a list of words in a file, enter the following command:

```
tr -cs '[:lower:][:upper:]' '[\n*]' < textfile > newfile
```

This transforms each sequence of characters other than lowercase letters and uppercase letters into a single newline character. The * (asterisk) causes the **tr** command to repeat the new line character enough times to make the second string as long as the first string.

5. To delete all NULL characters from a file, enter the following command:

```
tr -d '\0' < textfile > newfile
```

6. To replace every sequence of one or more new lines with a single new line, enter the following command:

```
tr -s '\n' < textfile > newfile
```

OR

```
tr -s '\012' < textfile > newfile
```

7. To replace every nonprinting character, other than valid control characters, with a ? (question mark), enter the following command:

```
tr -c '[:print:][:cntrl:]' '[?*' < textfile > newfile
```

This scans a file created in a different locale to find characters that are not printable characters in the current locale.

8. To replace every sequence of characters in the <space> character class with a single # character, enter the following command:

```
tr -s '[:space:]' '[#*]'
```

9. **>|** To define a character range for a specific locale, enter the following command:

```
LANG=ES_ES tr -L a-z A-Z abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

This command defines the a-z and A-Z character ranges for the ES_ES (Spanish_Spain) locale. |<

10. >| To delete a character range for a specific locale, enter the following command:

```
LANG=ES_ES tr -L a-z A-Z
```

|<

trace Daemon

Purpose

Records selected system events.

Syntax

```
trace [ -a [ -g ] ] [ -f | -l ] [ -c ] [ -C [ CPUList | all ] ] [ -d ] [ -e string-cmd ] [ -h ] [ -j EventList ]  
[ -k EventgroupList ] [ -J EventgroupList ] [ -K EventgroupList ] [ -m Message ] [ -M ] [ -N ] [ -n ]  
[ -o Name ] [ -o- ] [ -p ] [ -r reglist ] [ -s ] [ -A ProcessIDList ] [ -t ThreadIDList ] [ -x program-specification  
| -X program-specification ] [ -I ] [ -P trace-propagation ] [ -L Size ] [ -T Size ] [ -W ] [ -@ WparList ]
```

Description

The **trace** daemon configures a trace session and starts the collection of system events. The data collected by the trace function is recorded in the trace log. A report from the trace log can be generated with the **trcrpt** command.

When invoked with the **-a**, **-x**, or **-X** flags, the trace daemon is run asynchronously (for example, as a background task). Otherwise, it is run interactively and prompts you for subcommands.

To put the WPARconfigured ID (CID) in the trace hooks, use the **-W** flag.

To trace specific WPAR, use the **-@** flag with a list of WPAR names that you want to trace.

You can use the System Management Interface Tool (SMIT) to run the **trace** daemon. To use SMIT, enter:

```
smit trace
```

The following are modes of trace data collection:

Item	Description
Alternate (the default)	All trace events are captured in the trace log file.
Circular (-l)	The trace events wrap within the in-memory buffers and are not captured in the trace log file until the trace data collection is stopped.
Single (-f)	The collection of trace events stops when the in-memory trace buffer fills up and the contents of the buffer are captured in the trace log file.

Item	Description
Buffer Allocation	Trace buffers are allocated from either the kernel heap, or are put into separate segments. By default, buffers are allocated from the kernel heap unless the buffer size requested is too large for buffers to fit in the kernel heap, in which case they are allocated in separate segments. Allocating buffers from separate segments hinders trace performance somewhat. However, buffers in separate segments will not take up paging space, just pinned memory.

You can elect to trace only selected processes or threads. You can also trace a single program. You can specify whether the trace is to be propagated or extended to newly created processes or threads. When channel 0 is active, you can optionally include interrupt events in such traces.

Note:

1. Unless the trace is started before the process that is being traced, the process startup events are not captured. If the trace is started before the process that is being traced, some events from processes other than the process being traced will be captured as well.
2. When trace uses memory from the kernel heap, this memory remains part of kernel memory until the next reboot of the system. Thus, care should be taken when using large buffers.

Flags

Item	Description
-@ <i>WparList</i>	Traces the workload partitions that you specify in the <i>WparList</i> parameter. Multiple WPAR names can either be separated by commas or enclosed in quotation marks and separated by spaces. To include the current Global system in the trace, specify Global. You can only specify the -@ flag in the Global system in a workload partition environment.
-a	Runs the trace daemon asynchronously (i.e. as a background task). Once trace has been started this way, you can use the trcon , trcoff , and trcstop commands to respectively start tracing, stop tracing, or exit the trace session. These commands are implemented as links to trace .
-A <i>ProcessIDList</i>	Traces only the processes and, optionally, their children specified with the <i>ProcessIDList</i> . A process ID is a decimal number. Multiple process IDs can either be separated by commas or enclosed in quotation marks and separated by spaces. The -A flag is only valid for trace channel 0; the -A and -g flags are incompatible. All threads existing for the specified processes when tracing is started are traced. By default, if after the trace starts, the processes being traced create additional threads or processes, these are not traced unless the -P flag is specified.
-c	Saves the trace log file, adding .old to its name.

Item

-C [*CPUList* | **all**]

Description

Traces using one set of buffers per processor in the *CPUList*. The processors can be separated by commas, or enclosed in double quotation marks and separated by commas or blanks. To trace all processors, specify **all**. Since this flag uses one set of buffers per processor, and produces one file per processor, it can consume large amounts of memory and file space, and should be used with care. The files produced are named **trcfile**, **trcfile-0**, **trcfile-1**, etc., where **0**, **1**, etc. are the processor numbers. If **-T** or **-L** are specified, the sizes apply to each set of buffers and each file. On a uniprocessor system, you may specify **-C all**, but **-C** with a list of processor numbers is ignored.



Attention: The **-C** flag can only be used by the root user.

-d

Disables the automatic start of trace data collection. Delays starting of trace data collection. Normally, the collection of trace data starts automatically when you issue the **trace** daemon. Use the **trcon** command to start the collection of trace data.

-e *string-cmd*

Configures Component Trace by running `ctctrl` with *string-cmd* as an argument before the trace is started. In other words, it runs `ctctrl string-cmd`. Passing multiple **-e** options is allowed and is equivalent to successively running the `ctctrl` command with each *string-cmd* of arguments. This option can be used to configure the system trace mode (by setting the system trace mode to On, changing the level of trace, and so on) for some components just before starting to trace the system.

-f

Runs **trace** in a single mode. Causes the collection of trace data to stop as soon as the in-memory buffer is filled up. The trace data is then written to the trace log. Use the **trcon** command to restart trace data collection and capture another full buffer of data. If you issue the **trcoff** subcommand before the buffer is full, trace data collection is stopped and the current contents of the buffer are written to the trace log.

-g

Starts a trace session on a generic trace channel (channels 1 through 7). This flag works only when **trace** is run asynchronously (**-a**). The return code of the command is the channel number; the channel number must subsequently be used in the generic trace subroutine calls. To stop the generic trace session, use the command **trcstop - <channel_number>**.

Item	Description
-h	Omits the header record from the trace log. Normally, the tracedaemon writes a header record with the date and time (from the date command) at the beginning of the trace log; the system name, version and release, the node identification, and the machine identification (from the uname -a command); and a user-defined message. At the beginning of the trace log, the information from the header record is included in the output of the trcrpt command.
-I	Trace interrupt events. When specified with -A or -t , the -I flag includes interrupt events along with the events for the processes or threads specified. When -I is specified, but neither -A nor -t is specified, only interrupt level events are traced. The -I flag is only valid for trace channel 0; the -I and -g flags are incompatible.
-j <i>EventList</i>	Specifies the user-defined events to collect trace data. The list items specified in the <i>EventList</i> parameter can either be separated by commas or enclosed in quotation marks and separated by commas or spaces. In AIX 6.1 and earlier releases, specifying a two-digit hook ID in the form hh specifies hh00 , hh10 , ..., hhF0 . Specifying a three-digit hook ID in the form hhh specifies hhh0 . Specifying a four-digit hook ID in the form hhhh specifies hhhh . If any of these events is missing, the information reported by the trcrpt command will be incomplete. Consequently, when using the -j flag, include all these events in the <i>EventList</i> . If starting the trace with SMIT, or the -J flag, these events are in the tidhk group.
-J <i>EventgroupList</i>	Specifies the event groups to be included. The list items specified in the <i>EventgroupList</i> parameter can either be separated by commas or enclosed in quotation marks and separated by commas or spaces. The -J and -K flags work like -j and -k , except with event groups instead of individual hook IDs. You can specify each flag -j , -J , -k , and -K within the command.

Item

-k *EventgroupList*

Description

Specifies the user-defined events to exclude trace data. The list items specified in the *EventgroupList* parameter can either be separated by commas or enclosed in quotation marks and separated by commas or spaces. In AIX 6.1 and earlier releases, specifying a two-digit hook ID in the form **hh** specifies **hh00**, **hh10**, ..., **hhF0**. Specifying a three-digit hook ID in the form **hhh** specifies **hhh0**. Specifying a four-digit hook ID in the form **hhhh** specifies **hhhh**.

Tip: The following events are used to determine the **pid**, the **cpuid**, and the **exec** path name in the **trcrpt** report:

```
106 DISPATCH
10C DISPATCH IDLE PROCESS
134 EXEC SYSTEM CALL
139 FORK SYSTEM CALL
465 KTHREAD CREATE
```

If any of these events is missing, the information reported by the **trcrpt** command will be incomplete. When using the **-k** flag, do not include these events in the *EventgroupList* parameter. If starting the trace with SMIT, or the **-J** flag, these events are in the **tidhk** group.

-K *EventgroupList*

Specifies the event groups to be excluded. The list items specified in the *EventgroupList* parameter can either be separated by commas or enclosed in quotation marks and separated by commas or spaces. The **-J** and **-K** flags work like **-j** and **-k**, except with event groups instead of individual hook IDs. You can specify each flag **-j**, **-J**, **-k**, and **-K** within the command.

-l

Runs **trace** in a circular mode. The **trace** daemon writes the trace data to the trace log when the collection of trace data is stopped. Only the last buffer of trace data is captured. When you stop trace data collection using the **trcoff** command, restart it using the **trcon** command.

-L *Size*

Overrides the default trace log file size of 1 MB with the value stated. Specifying a file size of zero sets the trace log file size to the default size.

Note: In the circular and the alternate modes, the trace log file size must be at least twice the size of the trace buffer. In the single mode, the trace log file must be at least the size of the buffer. See the **-T** flag for information on controlling the trace buffer size.

-m *Message*

Specifies text to be included in the message field of the trace log header record.

Item	Description
-M	Dumps the address map of running processes into the trace. The -M flag must be specified if the trace file is to be processed by the tprof command.
-n	Adds information to the trace log header: lock information, hardware information, and, for each loader entry, the symbol name, address, and type.
-N	Dump the address map of specified processes into the trace. The -N option is used in conjunction with -M option.
-o <i>Name</i>	Overrides the /var/adm/ras/trcfile default trace log file and writes trace data to a user-defined file.
-o -	Overrides the default trace log name and writes trace data to standard output. The -c flag is ignored when using this flag. An error is produced if -o - and -C are specified.
-p	Includes the cpuid of the current processor with each hook. This flag is only valid for 64-bit kernel traces. Note: The trcrpt command can report the cpuid whether or not this option is specified.
-P <i>propagation</i>	The propagation is specified with the letters p for propagation across process creation, t for propagation across thread creation, and n for no propagation. Propagation across process creation implies propagation across thread creation. For example, if -A is specified to trace a process, all threads for that process that exist at the time the trace was started are traced. The -Pt flags causes all threads subsequently created by that process to be traced as well. If -Pp is specified, all processes and threads subsequently created by that process are traced. If -t all was specified to trace all threads, -P is ignored. The -P flag is only valid for trace channel 0; the -P and -g flags are incompatible.

Item	Description
-r <i>reglist</i>	Optional, and only valid for a trace run on a 64-bit kernel. <i>reglist</i> items are separated by commas, or enclosed in quotation marks and separated by blanks. Up to 8 registers may be specified. Valid <i>reglist</i> values are: PURR - The PURR Register for this processor SPURR The SPURR register for this processor MCR0, MCR1, MCRA - the MCR Registers, 0, 1, and A PMC1, PMC2, ... PMC8 - PMC Registers 1 through 8. When the -r flag is specified, the trace command dumps the specified register values into the header. Restriction: Not all registers are valid for all processors.
-s	Stops tracing when the trace log fills. The trace daemon normally wraps the trace log when it fills up and continues to collect trace data. During asynchronous operation, this flag causes the trace daemon to stop trace data collection. (During interactive operation, the quit subcommand must be used to stop trace.)
-t <i>ThreadIDList</i>	Traces only the threads specified with the <i>ThreadIDList</i> parameter. A thread ID is a decimal number. Multiple thread IDs can either be separated by commas or enclosed in quotation marks and separated by spaces. Also, the thread list can be all or *, indicating that all threads are to be traced. This is useful for tracing all thread-related events without tracing interrupt-related events. However, if -t all and -I are both specified, this is the same as specifying neither one; all events are traced. Another way to say this is that trace and trace -It all are identical. The -t flag is only valid for trace channel 0, the -t and -g flags are incompatible.

Item**-T** *Size***Description**

Overrides the default trace buffer size of 128 KB with the value stated. You must be root to request more than 1 MB of buffer space. The maximum possible size is 268435184 bytes, unless the **-f** flag is used, in which case it is 536870368 bytes. The smallest possible size is 8192 bytes, unless the **-f** flag is used, in which case it is 16392 bytes. Sizes between 8192 and 16392 will be accepted when using the **-f** flag; however, the actual size used will be 16392 bytes.

Note: In the circular and the alternate modes, the trace buffer size must be one-half or less the size of the trace log file. In the single mode, the trace log file must be at least the size of the buffer. See the **-L** flag for information on controlling the trace log file size. Also note that trace buffers use pinned memory, which means they are not pageable. Therefore, the larger the trace buffers, the less physical memory is available to applications.

The system attempts to allocate the buffer space from the kernel heap. If this request can not be satisfied, the system then attempts to allocate the buffers as separate segments.

The **-f** flag actually uses two buffers, which behave as a single buffer (except that a buffer wraparound trace hook will be recorded when the first buffer is filled).

-W

Use the **-W** flag to include the workload partitionconfigured ID (CID) for the current process with each hook. This flag is only valid in the Global system in a workload partition environment.

Tip: The **trcrpt** command can report the workload partitionCID whether or not this option is specified.

Item	Description
-x <i>program-specification</i>	Traces the specified program. The <i>program-specification</i> specifies a program and parameters as they would be when running the program from the shell, except that the program specification must be in quotes if more than just the program's name is given. The trace is stopped automatically when the program exits, and returns the program's return code. By default, any processes and threads created by the program are also traced; as if -Pp was specified. To change this behavior, use -Pn to specify no trace propagation, or -Pt to propagate trace only to threads created by the program's original process. Tip: The -x flag implies asynchronous tracing, as if the -a flag had also been specified.
-X <i>program-specification</i>	The -X flag works like the -x flag, except that the trace is not automatically stopped when the program exits. This is useful for tracing programs which fork processes, and then terminate, and you want these new processes traced as well.

Subcommands

When run interactively, trace recognizes the following subcommands:

Item	Description
trcon	Starts the collection of trace data.
trcoff	Stops the collection of trace data.
q or quit [-serial -dd]	Stops the collection of trace data and exits trace . If the -s option is specified then this serializes any pending I/O operations. If the -d option is specified, any pending I/O operation is discarded.
! <i>Command</i>	Runs the shell command specified by the <i>Command</i> parameter.
?	Displays the summary of trace subcommands.

Signals

The **INTERRUPT** signal acts as a toggle to start and stop the collection of trace data. Interruptions are set to **SIG_IGN** for the traced process.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

To run the **trace** command on channel 0, you must have the following additional authorizations, if channel 0 **trace** restriction is enabled in the **trcctl** command:

Item	Description
aix.ras.trace.tracech0	Required to run the trace command on channel 0.

Note: By default, the root and system group users are privileged users.

To perform all functionality of all commands including the **trace** command on channel 0, you must have the following additional authorizations:

Item	Description
aix.ras.trace.trace	Required to perform all trace operations.

Examples

1. To use trace interactively, enter **trace**, (the > prompt is displayed), then specify the [subcommands](#) you want. For example, to trace system events during the run of the *anycmd* command, enter:

```
trace
> !anycmd
> q
```

2. To avoid delays when the command finishes, start trace asynchronously (**-a**), using only one command line, enter:

```
trace -a; anycmd; trcstop
```

3. To trace the system itself for a period of 10 seconds, enter:

```
trace -a; sleep 10; trcstop
```

4. To output trace data to a specific trace log file (instead of the **/var/adm/ras/trcfile** default trace log file), :

```
trace -a -o /tmp/my_trace_log; anycmd; trcstop
```

5. To capture the execution of a **cp** command, excluding specific events from the collection process:

```
trace -a -k "20e,20f" -x "cp /bin/track /tmp/junk"
```

In the example above, the **-k** option suppresses the collection of events from the **lockl** and **unlockl** functions (20e and 20f events).

Also notice that the **-x** flag was used, so only hooks associated with the **cp** command process will be traced, and no interrupt activity will be traced.

6. To trace hook 234 and the hooks that will allow you to see the process names, use:

```
trace -a -j 234 -J tidhk
```

This traces the hooks in the event-group "tidhk" plus hook 234.

7. To have trace use one set of buffers per processor, specify:

```
trace -aC all
```

The files produced are **/var/adm/ras/trcfile**, **/var/adm/ras/trcfile-0**, **/var/adm/ras/trcfile-1**, etc. up to **/var/adm/ras/trcfile-(n-1)**, where *n* is the number of processors in the system.

Tip: **trace -aC all -o mylog** produces the files **mylog**, **mylog-0**, **mylog-1**, ...

8. To trace a program that starts a daemon process, and to continue tracing the daemon after the original program has finished, use

```
trace -X "mydaemon"
```

The trace must be stopped with **trcstop**.

9. To trace *mydaemon*, which is currently running, use:

```
trace -A mydaemon-process-id -Pp
```

Where *mydaemon-process-id* is the process for *mydaemon* as returned by the **ps** command. The **-Pp** flag tells trace to also trace any processes and threads created by *mydaemon* while the trace is running.

10. To capture the PURR, and PMC1 and PMC2, type:

```
trace -ar "PURR PMC1 PMC2"
```

11. To trace hooks 1A00,1A10,...,1AF0, DCA0 and 1AB1, enter:

```
trace -aj 1A,DCA,1AB1
```

Files

Item	Description
/usr/include/sys/trcmacros.h	Defines trchhook and utrchhook macros.
/var/adm/ras/trcfile	Contains the default trace log file.

traceauth Command

Purpose

Trace the authorizations that a command needs to run successfully.

Syntax

```
traceauth [ -d ] [ -e ] [ -f ] [ -o outputfile ] Command [ args ]
```

Description

The **traceauth** command records the authorizations that a command attempts to use when the command is run. There are two ways an authorization can be used. The first way is the **accessauths** attribute that grants access to run a specified program. The second way is the **checkauths** attribute that is checked in a program before performing a privileged operation. The **traceauth** command can trace and report both types of authorizations. The **traceauth** command is used either for command investigation when entries are added to the privileged command database or to identify which authorizations to use while creating a role. The **traceauth** command runs the command specified by the *Command* parameter, along with associated arguments for the *Command*.

Generally, run the **traceauth** command with the PV_ROOT privilege or by assuming a role that has **aix** authorization so that any attempt to use authorization would succeed. In this case, the traceauth command can keep track of all of the authorizations that the command specified in the *Command* parameter needs for a successful run without the PV_ROOT privilege or a special role. After the command specified in the *Command* parameter is run, the list of used **accessauths** and **checkauths** are written to the standard output (stdout) file.

Flags

Item	Description
-d	Display the output of the truss command with the authorizations that are required by the command.

Item	Description
-e	Follow the exec subroutine. If the command specified by the <i>Command</i> parameter runs an exec subroutine, the traceauth command reports the authorizations needed so far, and then proceeds with recording the authorizations associated with the new executable file. If the file run by the exec subroutine has its setuid bit set and is not owned by root, the traceauth command cannot properly trace the authorizations use of the file.
-f	Follow the fork subroutine. If the controlled process calls the fork subroutine, the traceauth command also reports the authorizations used by the new child process.
-o	Write the output to the specified file instead of the standard output (stdout) file.

Parameters

Item	Description
<i>args</i>	Specifies the arguments for the associated command in the <i>Command</i> parameter.
<i>Command</i>	Specifies the name of the command whose authorizations you want to trace.
<i>outputfile</i>	If you do not want to write the output to the standard output (stdout) file, use the -o flag. Then, specify the name of the output file to which you want to record the authorizations in the <i>outputfile</i> parameter.

tracepriv Command

Purpose

Traces the privileges that a command needs for a successful run.

Syntax

```
tracepriv [ -d ] [ -e ] [ -f ] [ -o outputfile ] Command [ args ]
```

Description

The **tracepriv** command records the privileges that a command attempts to use when the command is run. The **tracepriv** command is used for command investigation when entries are added to the privileged command database. The **tracepriv** command runs the command specified by the *Command* parameter with the specified arguments (with the *args* parameter). Generally, run the **tracepriv** command with the PV_ROOT privilege so that any attempt to use a privilege succeeds. In this case, the **tracepriv** command can keep track of all of the privileges that the *Command* needs for a successful run without the PV_ROOT privilege. After the *Command* is run or when an **exec** subroutine within the command occurs, the list of used privileges is written to standard output (**stdout**).

Flags

Item	Description
-d	Displays the output of the truss command with the privileges that is required by the command.

Item	Description
-e	Follows the exec subroutine. If the command specified by the <i>Command</i> parameter runs an exec subroutine, the tracepriv command reports the privileges needed so far (and set them if the -a flag is used), and then proceeds with recording (and setting) the privileges associated with the new executable file. If the file run by the exec subroutine has its setuid bit set and is not owned by root, the tracepriv command cannot properly trace the privilege use of the file.
-f	Follows the fork subroutine. If the controlled process calls the fork subroutine, the tracepriv command also reports the privileges used by the new child process.
-o	Writes the output to the specified file instead of the standard output (stdout).

Parameters

Item	Description
<i>args</i>	Specifies the arguments.
<i>Command</i>	Specifies the command.
<i>outputfile</i>	Specifies the file to record the output.

traceroute Command

Purpose

Prints the route that IP packets take to a network host.

Syntax

```
traceroute [ -m Max_ttl ] [ -n ] [ -p Port ] [ -q Nqueries ] [ -r ] [ -d ] [ -g gateway_addr ] [ -s SRC_Addr ]
[ -t TypeOfService ] [ -f flow ] [ -v ] [ -w WaitTime ] Host [ PacketSize ]
```

Description

Attention: The **traceroute** command is intended for use in network testing, measurement, and management. It should be used primarily for manual fault isolation. Because of the load it imposes on the network, the **traceroute** command should not be used during normal operations or from automated scripts.

The **traceroute** command attempts to trace the route an IP packet follows to an Internet host by launching UDP probe packets with a small maximum time-to-live (*Max_ttl* variable), then listening for an ICMP **TIME_EXCEEDED** response from gateways along the way. Probes are started with a *Max_ttl* value of one hop, which is increased one hop at a time until an ICMP **PORT_UNREACHABLE** message is returned. The ICMP **PORT_UNREACHABLE** message indicates either that the host has been located or the command has reached the maximum number of hops allowed for the trace.

The **traceroute** command sends three probes at each *Max_ttl* setting to record the following:

- *Max_ttl* value
- Address of the gateway
- Round-trip time of each successful probe

The number of probes sent can be increased by using the **-q** flag. If the probe answers come from different gateways, the command prints the address of each responding system. If there is no response from a probe within a 3-second time-out interval, an * (asterisk) is printed for that probe.

The **tracert** command prints an ! (exclamation mark) after the round-trip time if the *Max_ttl* value is one hop or less. A maximum time-to-live value of one hop or less generally indicates an incompatibility in the way ICMP replies are handled by different network software. The incompatibility can usually be resolved by doubling the last *Max_ttl* value used and trying again.

Other possible annotations after the round-trip notation are:

Item	Description
!H	Host unreachable
!N	Network unreachable
!P	Protocol unreachable
!S	Source route failed
!F	Fragmentation needed

If the majority of probes result in an error, the **tracert** command exits.

The only mandatory parameter for the **tracert** command is the destination host name or IP number. The **tracert** command will determine the length of the probe packet based on the Maximum Transmission Unit (MTU) of the outgoing interface. The UDP probe packets are set to an unlikely value so as to prevent processing by the destination host.

Flags

Item	Description
-d	Enables socket level debugging.
-f <i>flow</i>	Sets the flow label field in IPv6 packet header. The default value is 0.
-g <i>gateway_addr</i>	Routes the outgoing packets through a specified gateway with the IP source routing option. Before you use this flag, your router must enable IP source routing. This flag is only available for IP version 6 addresses.
-m <i>Max_ttl</i>	Sets the maximum time-to-live (maximum number of hops) used in outgoing probe packets. The default is 30 hops (the same default used for TCP connections).
-n	Prints hop addresses numerically rather than symbolically and numerically. This flag saves a name-server address-to-name lookup for each gateway found on the path.
-p <i>Port</i>	Sets the base UDP port number used in probes. The default is 33434. The tracert command depends on an open UDP port range of <i>base</i> to <i>base + nhops - 1</i> at the destination host. If a UDP port is not available, this option can be used to pick an unused port range.
-q <i>Nqueries</i>	Specifies the number of probes the tracert command sends at each <i>Max_ttl</i> setting. The default is three probes.
-r	Bypasses the normal routing tables and sends the probe packet directly to a host on an attached network. If the specified host is not on a directly attached network, an error is returned. This option can be used to issue a ping command to a local host through an interface that is not registered in the routed daemon's routing table.

Item	Description
-s <i>SRC_Addr</i>	Uses the next IP address in numerical form as the source address in outgoing probe packets. On hosts with more than one IP address, the -s flag can be used to force the source address to be something other than the IP address of the interface on which the probe packet is sent. If the next IP address is not one of the machine's interface addresses, an error is returned and nothing is sent.
-t <i>TypeOfService</i>	Sets the <i>TypeOfService</i> variable in the probe packets to a decimal integer in the range of 0 to 255. The default is 0. This flag can be used to investigate whether different service types result in different paths. For more information, see TCP/IP Protocols in Performance Tools Guide and Reference . Useful values are -t 16 (low delay) and -t 8 (high throughput).
-v	Receives packets other than TIME_EXCEEDED and PORT_UNREACHABLE (verbose output).
-w <i>WaitTime</i>	Sets the time (in seconds) to wait for a response to a probe. The default is 3 seconds.

Parameters

Item	Description
<i>Host</i>	Specifies the destination host, either by host name or IP number. This parameter is required.
<i>PacketSize</i>	Specifies the probe datagram length. The default packet size is determined by the tracert command based on the MTU of the outgoing interface.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. A sample use and output is:

```
[yak 71]% traceroute nis.nsf.net.
traceroute to nis.nsf.net (35.1.1.48), 30 hops max, 56 byte packet
 1 helios.ee.lbl.gov (128.3.112.1) 19 ms 19 ms 0 ms
 2 lilac-dmc.Berkeley.EDU (128.32.216.1) 39 ms 39 ms 19 ms
 3 lilac-dmc.Berkeley.EDU (128.32.216.1) 39 ms 39 ms 19 ms
 4 ccngw-ner-cc.Berkeley.EDU (128.32.136.23) 39 ms 40 ms 39 ms
 5 ccn-nerif22.Berkeley.EDU (128.32.168.22) 39 ms 39 ms 39 ms
 6 128.32.197.4 (128.32.197.4) 40 ms 59 ms 59 ms
 7 131.119.2.5 (131.119.2.5) 59 ms 59 ms 59 ms
 8 129.140.70.13 (129.140.70.13) 99 ms 99 ms 80 ms
 9 129.140.71.6 (129.140.71.6) 139 ms 239 ms 319 ms
10 129.140.81.7 (129.140.81.7) 220 ms 199 ms 199 ms
11 nic.merit.edu (35.1.1.48) 239 ms 239 ms 239 ms
```

Lines 2 and 3 are the same due to a bug in the kernel on the second hop system (lbl - csam. arpa) that forwards packets with a zero time-to-live. Host names are not printed in lines 6 through 10 because the National Science Foundation Network (NSFNet, 129.140) does not provide address-to-name translations for its nodes.

2. Another output example might be:

```
[yak 72]% traceroute rip.Berkeley.EDU (128.32.131.22)
traceroute to rip.Berkeley.EDU (128.32.131.22), 30 hops max
```

```

1 helios.ee.lbl.gov (128.3.112.1) 0 ms 0 ms 0 ms
2 lilac-dmc.Berkeley.EDU (128.32.216.1) 39 ms 19 ms 39 ms
3 lilac-dmc.Berkeley.EDU (128.32.216.1) 19 ms 39 ms 19 ms
4 ccngw-ner-cc.Berkeley.EDU (128.32.136.23) 39 ms 40 ms 19 ms
5 ccn-nerif35.Berkeley.EDU (128.32.168.35) 39 ms 39 ms 39 ms
6 csgw/Berkeley.EDU (128.32.133.254) 39 ms 59 ms 39 ms
7 * * *
8 * * *
9 * * *
10 * * *
11 * * *
12 * * *
13 rip.Berkeley.EDU (128.32.131.22) 59 ms! 39 ms! 39 ms!

```

In this example, exactly half of the 12 gateway hops (13 is the final destination) are "missing." However, these hops were actually not gateways. The destination host, a Sun-3 workstation running Sun OS3.5, used the ttl from the arriving datagram as the ttl in its ICMP reply; thus, the reply timed out on the return path. Because ICMPs are not sent for ICMPs, no notice was received. The ! (exclamation mark) after each round-trip time indicates some type of software incompatibility problem. (The cause was diagnosed after the **traceroute** command issued a probe of twice the path length. The destination host was really only seven hops away.)

tracesoff Command

Purpose

Turns off tracing of a subsystem, a group of subsystems, or a subserver.

Syntax

Subsystem

tracesoff [**-h** *Host*] { **-g** *Group* | **-p** *SubsystemPID* | **-s** *Subsystem* }

Subserver

tracesoff [**-h** *Host*] **-t** *Type* [**-p** *SubsystemPID*] { **-o** *Object* | **-P** *SubserverPID* }

Description

The **tracesoff** command sends the System Resource Controller a subsystem request packet that is forwarded to the subsystem to turn tracing off. Tracing is unsuccessful if the communication method for the subsystems is signals.

Note: Tracing is subsystem dependent.

Flags

Item	Description
-g <i>Group</i>	Specifies a group of subsystems to turn tracing off. The command is unsuccessful if the <i>Group</i> name is not contained in the subsystem object class.
-h <i>Host</i>	Specifies the foreign host on which this trace action is requested. The local user must be running as root. The remote system must be configured to accept remote System Resource Controller requests. That is, the srcmstr daemon (see /etc/inittab) must be started with the -r flag and the /etc/hosts.equiv or .rhosts file must be configured to allow remote requests.
-o <i>Object</i>	Specifies that a subserver <i>Object</i> name is to be passed to the subsystem as a character string.

Item	Description
-p <i>SubsystemPID</i>	Specifies a particular instance of the subsystem to turn tracing off, or a particular instance of the subsystem to which the trace off subserver request is to be passed.
-P <i>SubserverPID</i>	Specifies that a <i>SubserverPID</i> is to be passed to the subsystem as a character string.
-s <i>Subsystem</i>	Specifies a subsystem to turn tracing off. The <i>Subsystem</i> name can be the actual subsystem name or the synonym name for the subsystem. The command is unsuccessful if the <i>Subsystem</i> name is not contained in the subsystem object class.
-t <i>Type</i>	Specifies a subsystem subserver to turn tracing off. The command is unsuccessful if the <i>Type</i> is not contained in the subserver object class.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To turn off the tracing of a group, enter the following command:

```
tracesoff -g tcpip
```

This turns the tracing off for the tcpip group.

2. To turn off tracing of the sendmail subsystem on a foreign host, enter the following command:

```
tracesoff -h odin -s sendmail
```

This turns off the tracing for the sendmail subsystem on the odin foreign host.

Files

Item	Description
/usr/bin/tracesoff	Contains the tracesoff command.
/etc/objrepos/SRCsubsys	Specifies the SRC Subsystem Configuration Object Class.
/etc/objrepos/SRCsubsvr	Specifies the SRC Subserver Configuration Object Class.
/etc/services	Defines the sockets and protocols used for Internet services.
/dev/SRC	Specifies the AF_UNIX socket file.
/dev/.SRC-unix	Specifies the location for temporary socket files.

traceson Command

Purpose

Turns on tracing of a subsystem, a group of subsystems, or a subserver.

Syntax

Subsystem

traceson [-h *Host*] [-l] { -g *Group* | -p *SubsystemPID*| -s *Subsystem*}

Subserver

traceson [-h *Host*] [-l] -t *Type* [-o *Object*] [-p *SubsystemPID*] [-P *SubserverPID*]

Description

The **traceson** command sends the System Resource Controller a subsystem request packet that is forwarded to the subsystem to turn tracing on. Tracing is unsuccessful if the communication method for the subsystems is signals.

Note: Tracing is subsystem dependent.

Tracing may occur in either short or long form. When the -l flag is absent, the trace request is assumed to be a short trace.

Flags

Item	Description
<u>-g</u> <i>Group</i>	Specifies a group of subsystems to turn tracing on. The command is unsuccessful if the <i>Group</i> name is not contained in the subsystem object class.
<u>-h</u> <i>Host</i>	Specifies the foreign host on which this trace action is requested. The local user must be running as "root". The remote system must be configured to accept remote System Resource Controller requests. That is, the srcmstr daemon (see /etc/inittab) must be started with the -r flag and the /etc/hosts.equiv or .rhosts file must be configured to allow remote requests.
<u>-l</u>	Specifies that a long trace is requested.
<u>-o</u> <i>Object</i>	Specifies that a subserver object is to be passed to the subsystem as a character string.
<u>-p</u> <i>SubsystemPID</i>	Specifies a particular instance of the subsystem to turn tracing on, or a particular instance of the subsystem to which the trace subserver request is to be passed.
<u>-P</u> <i>SubserverPID</i>	Specifies that a subserver PID is to be passed to the subsystem as a character string.
<u>-s</u> <i>Subsystem</i>	Specifies the subsystem to turn tracing on. The <i>Subsystem</i> name can be either the actual subsystem name or the synonym name for the subsystem. The command is unsuccessful if the <i>Subsystem</i> name is not contained in the subsystem object class.
<u>-t</u> <i>Type</i>	Specifies a subserver to turn tracing on. The command is unsuccessful if the <i>Type</i> is not contained in the subserver object class.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To turn on tracing of the tcpip group on a foreign host, enter the following command:

```
traceson -h odin -g tcpip
```

This turns on the tracing for the `tcpip` group on the `odin` foreign host.

2. To turn on tracing of the `sendmail` subsystem on a foreign host, enter the following command:

```
traceson -h odin -s sendmail
```

This turns on the tracing for the `sendmail` subsystem on the `odin` foreign host.

Files

Item	Description
<code>/usr/bin/traceson</code>	Contains the traceson command.
<code>/etc/objrepos/SRCsubsys</code>	Specifies the SRC Subsystem Configuration Object Class.
<code>/etc/objrepos/SRCsubsvr</code>	Specifies the SRC Subserver Configuration Object Class.
<code>/etc/services</code>	Defines the sockets and protocols used for Internet services.
<code>/dev/SRC</code>	Specifies the AF_UNIX socket file.
<code>/dev/.SRC-unix</code>	Specifies the location for temporary socket files.

trbsd Command

Purpose

Translates characters (BSD version).

Syntax

```
trbsd [ -c ] [ -d ] [ -s ] [ -A ] [ String1 [ String2 ] ]
```

Description

The **trbsd** command deletes or substitutes characters from standard input and then writes the result to standard output. The **trbsd** command is the BSD version of the **tr** command. The **trbsd** command performs three kinds of operations, depending on the character strings specified by the parameters and flags specified. The default value for either the *String1* or *String2* parameter is a null string.

Transforming Characters

If both the *String1* and *String2* parameters are specified and the **-d** flag is not specified, the **trbsd** command replaces each character from standard input that is specified by the *String1* parameter with the character in the same position in the *String2* parameter.

If the *String1* parameter specifies a character more than once, the character is translated into the character in the *String2* parameter that corresponds to the last occurrence of the character in the *String1* parameter.

Deleting Characters Using the -d Flag

If the **-d** flag is specified, the **trbsd** command deletes each character from standard input that is specified by the *String1* parameter.

Removing Sequences of Characters Using the -s Flag

If the **-s** flag is specified, the **trbsd** command deletes from standard input all but the first character in a sequence of two or more repetitions of any character specified by the *String2* parameter.

Both the *String1* and *String2* parameters must be specified when both the **-d** and **-s** flags are specified.

Note: The **trbsd** command deletes all null characters from standard input before it begins processing.

Special Sequences for Expressing Strings

The strings contained in *String1* and *String2* parameters can be expressed using the following conventions:

Item	Description
<i>C1-C2</i>	Specifies the string of characters that collate between the character specified by the <i>C1</i> string and the character specified by the <i>C2</i> string, inclusive. The character specified by the <i>C1</i> string must collate before the character specified by the <i>C2</i> string.
<i>\Octal</i>	Specifies the character whose encoding is represented by the specified octal value. The octal value can be a one-, two-, or three-digit octal integer. Multibyte characters can be expressed by writing backslash-octal sequences for each byte.
<i>\-</i>	The <i>\-</i> (backslash, minus sign) specifies the minus sign character itself, without any special meaning as an escape character.

If the strings specified by the *String1* and *String2* parameters are not the same length, the **trbsd** command pads the shorter string to equal the length of the longer string. Padding is accomplished by duplicating the last character in the shorter string as many times as necessary.

Flags

Item	Description
-A	Performs all operations on a byte-by-byte basis using the ASCII collation order for ranges and character classes, instead of the collation order of the current locale.
-c	Specifies that the value of the <i>String1</i> parameter be replaced by the complement of that string. The complement is all of the characters in the character set of the current locale, except for the characters specified by the <i>String1</i> parameter. If the -A and -c flags are specified together, characters are complemented with respect to the set of all 8-bit character codes.
-d	Deletes each character from standard input that is contained in the <i>String1</i> parameter.
-s	Deletes from standard input all but the first character in a sequence of two or more repetitions of any character contained in the <i>String2</i> parameter.

Examples

1. To translate braces into parentheses, enter:

```
trbsd '{ }' ' ( )' < textfile > newfile
```

This translates each { (left brace) to ((left parenthesis) and each } (right brace) to) (right parenthesis). All other characters remain unchanged.

2. To interchange plus signs with minus signs, and slash characters with asterisks, enter:

```
trbsd '+\-/ *' '\-+*/' < textfile > newfile
```

The minus sign must be entered with a backslash escape character.

3. To translate lowercase characters to uppercase, enter:

```
trbsd 'a-z' 'A-Z' < textfile > newfile
```

4. To create a list of words in a file, enter:

```
trbsd -cs 'a-zA-Z' '\012' < textfile > newfile
```

This translates each sequence of characters other than lowercase letters and uppercase letters into a single newline character. The octal value 012 is the code for the newline character.

5. To replace every sequence of one or more newlines with a single newline, enter:

```
trbsd -s '\012' < textfile > newfile
```

Files

Item	Description
<code>/usr/bin/trbsd</code>	Contains the trbsd command.
<code>/usr/ucb/tr</code>	Contains a symbolic link to the trbsd command.

trcctl Command

Purpose

Changes and displays system trace parameters.

Syntax

```
trcctl [ -d Directory -l -L LogfileSize -M LMT_log_dir -N NonrootUserBufferMax -o Logfile -r -T  
BufferSize -R disable|enable -S disable|enable ]
```

Description

The `trcctl` command will display or change the system trace default parameters. If the `-l` option (or no parameter) is specified, `trcctl` will show the values as follows:

```
Default Buffer Size: 131072  
Default Log File Size: 1310720  
Default Log File: /var/adm/ras/trcfile  
Non-root User Buffer Size Maximum: 1048576  
Default Components Directory File: /var/adm/ras/trc_ct  
Default LMT Log Dir: /var/adm/ras/mtrcdir  
Restrict non-privileged users from using trace Channel-0: enable  
Restrict non-privileged users from using trcript Channel-0: enable
```

Note that the default buffer and log file sizes initially depend upon the kernel. However, once they are set using this command, the effected value is the same for both kernels. The other parameters allow these default values to be changed. To change a default value, the user must be a member of the system group. Many of the flags used with `trcctl` correspond to those used by the trace daemon.

Flags

Item	Description
<code>-d <i>Directory</i></code>	Specifies the default Component Trace log directory path. The default value is <code>/var/adm/ras/trc_ct</code> .
<code>-l</code>	Lists the current values.
<code>-L <i>Value</i></code>	Specifies the default log file size. The original default value is 1310720 bytes for the 32-bit kernel, and 2621440 bytes for the 64-bit kernel. If specified with <code>-L</code> , the default will apply to both kernels.
<code>-M <i>LMT_log_dir</i></code>	Specifies the default Lightweight Memory Trace log directory path. The default value is <code>/var/adm/ras/mtrcdir</code> .
<code>-N <i>Value</i></code>	Specifies the maximum buffer size a non-root user may specify. The default is 1 MB, 1048576 bytes.

Item	Description
-o <i>Path</i>	Specifies the default log file path. The default value is <code>/var/adm/ras/trcfile</code> .
-R <i>disable</i> <i>enable</i>	Restricts the trace facility of channel 0 to only the privileged users. If the -R option is disabled, the trace facility of channel 0 is available to all users. The default value is <i>enable</i> . For more information, see the Security section of the trcrpt command.
-S <i>disable</i> <i>enable</i>	Restricts the trcrpt facility of channel 0 to only the privileged users. If the -S option is disabled, the trcrpt facility of channel 0 is available to all users. The default value is <i>enable</i> . For more information, see the Security section of the trace command
-r	Restores original default value.
-T <i>Value</i>	Specifies the default trace buffer size. The original default values are 128 KB and 256 KB for a 32- or 64-bit kernel. If specified with -T, the default will apply to both kernels.

Parameters

If you use 'k', 'm', or '#k', '#m' as parameters for the -N, -L, and -T options, `trcctl` will translate these into their respective byte totals.

k = 1024
m = 1048576

Using only 'k' or 'm', `trcctl` assumes you mean 1 kilobyte or 1 megabyte respectively. This way a root user can execute :

```
trcctl -L 10m -N m -T 256k
```

Security

The user must be a member of the system group.

trcdead Command

Purpose

Extracts trace buffers from a system dump image or live dump image.

Syntax

```
trcdead [ -1 -2 -3 ... -7 ] [ -c ] [ -M ] [ -o Name ] DumpImage [ UnixFile ]
```

Description

If the system halts while trace facilities are active, the contents of the internal trace buffers are captured in the system dump. Alternatively, a live dump can also capture partial or complete internal trace buffers if the appropriate pseudo-component. Use the **trcdead** command to extract the eight active system trace channels, all component trace buffers, and the lightweight memory trace buffers from the system dump or the live dump. The system trace channel 0 is extracted when you do not specify any flag. To trace a channel other than channel 0 is identified through a *-channelnum* flag. Use a **-c** flag to identify component trace buffers. Use the **-M** flag to identify lightweight memory trace buffers. You can extract only one type of trace buffer, or one specific system trace channel at one time.

The **-o** flag can be used to indicate that the extracted buffers should be written to a nondefault trace log file or directory. System trace channels are extracted to a trace log file. Component Trace buffers and

Lightweight Memory Trace buffers are extracted to a directory. If the **-o** flag is not chosen, the **trcdead** command writes to the default trace log file or directory. The default log file name and directory names can be viewed and modified using the **trcctl** command.

Use the **trcrpt** command to format a report from the trace log file or files.

Flags

Item	Description
-1, ..., -7	Retrieves the trace buffer entries for channel 1, 2, 3, 4, 5, 6, and 7. The default is channel 0.
-c	Extracts all buffers of all active Component Trace components.
-M	Extracts the Lightweight Memory Trace buffers.
-oName	Specifies the file or directory (-c, -M) to which data is written.

Parameter

Item	Description
<i>DumpImage</i>	Specifies the dump image to operate on.
<i>UNIX File</i>	Specifies the UNIX file that is in use when the system dump or live dump is taken. This is not necessary if you are using the trcdead command on the same system that the dump originated from.

Examples

Note: To determine which example is more appropriate for your system, use the **sysdumpdev** command to display the current dump device assignments.

1. To extract the system trace buffer to the file named `trace_extract` from a dump located at `/var/adm/ras/dumpfile`, enter:

```
trcdead -o trace_extract /var/adm/ras/dumpfile
```

2. To extract the system trace buffer from a dump image written to a device, enter:

```
trcdead /dev/hd7
```

3. To extract lightweight memory trace information from dump image `vmcore.0` and put it into the `/tmp` directory, enter:

```
trcdead -o /tmp -M vmcore.0
```

4. To extract the component trace buffers from the dump image `vmcore.3` that is produced by the **/tmp/unix_64**, enter:

```
trcdead -c vmcore.3 /tmp/unix_64
```

Files

Item	Description
/usr/bin/trcdead	Contains the trcdead command.
/var/adm/ras/dumpfile	Contains the default system dump file.
/var/adm/ras/trcfile	Contains the default system trace log.

Item	Description
<code>/var/adm/ras/trc_ct</code>	Contains the default component trace logs.
<code>/var/adm/ras/mtrcdir</code>	Contains the default lightweight memory trace logs.

trcevrp Command

Purpose

Manipulates trace event groups.

Syntax

List event groups

trcevrp -l [*event-group* [...]]

Remove event groups

trcevrp -r [*event-group* [...]]

Add an event group

trcevrp -a -d "*group-description*" **-h** "*hook-list*" *event-group*

Update an event group

trcevrp -u [**-d** "*group-description*"] [**-h** "*hook-list*"] *event-group*]

Description

The **trcevrp** command is used to maintain the trace event groups. You must be in the system group to add, delete, or change trace event groups. You *cannot* modify or delete event groups whose type is reserved.

In AIX version older than AIX 6.1, you can specify only three-digit hook IDs. In AIX 6.1 or later, you can specify four-digit hook IDs.

Flags

Item	Description
-a [-d <i>group-description</i> -h <i>hook-list</i>]	Creates a new event group. Only one event group name can be specified. Both -d <i>description</i> and -h <i>hook-list</i> must be specified when using the -a flag. If either -d or -h is not specified, an error is produced.
-d <i>group-description</i>	Designates the hook description. A description is required for all new groups.
-h <i>hook-list</i>	The hook list consists of trace hook IDs. The -h flag is required when using the -a flag. When updating an event group (-u flag), the hook-list, if specified, must contain all hook IDs for the group. List parameter items can be separated by commas or enclosed in double quotation marks and separated by commas or blanks.

Item	Description
-l <i>event-group</i>	The specified groups are listed to standard output. If no event group is specified, all are listed. The format of the listing is as follows: <i>group name - group-description (type) "hook list"</i> The following examples shows the listing of the group: * -l tidhk - Hooks needed to display thread name (reserved) "106,10C,134,139,465" * -l gka - GENERAL KERNEL ACTIVITY (files,execs,dispatches) (reserved) "106,10C,134,139,465,107,135,15b,12e,116,117,200,20E,20F" * -l mydriver - My Driver (files,execs,dispatches) (reserved) "106,1AB1,0AC0"
-r <i>event-group</i>	Removes the specified event-groups.
-u [-d " <i>group-description</i> " -h " <i>hook-list</i> "] <i>event-group</i>	Used to update the information for an event-group. Either -d <i>description</i> or -h <i>hook-list</i> must be specified.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To get a listing of all event groups, enter the following command:

```
trcevgrp -l
```

2. To add a new group, enter the following command:

```
trcevgrp -a -d "my group description" -h "500,501,502" mygrp
```

This will add the group called mygrp, give it the description my group description, and will have hooks of 500, 501, and 502.

3. To add another hook to mygrp, enter the following command:

```
trcevgrp -u -d "my group description" -h "500,501,502,503" mygrp
```

Note: You must specify all of the hook IDs.

Files the event groups are currently kept in the SWserveAt ODM database.

trcnm Command

Purpose

Generates a kernel name list.

Syntax

trcnm [**-a** [*FileName*]] [*FileName*] | **-K***Symbol1* ...

Description

The **trcnm** command generates a kernel name list used by the **trcrpt** command. A kernel name list is composed of a symbol table and a loader symbol table of an object file. The **trcrpt** command uses the kernel name list file to interpret addresses when formatting a report from a trace log file. For more information, see the **trcrpt -n** command.

If the *FileName* parameter is not specified, the default *FileName* is /unix.

Flags

Item	Description
-a	Writes all loader symbols to standard output. The default is to write loader symbols only for system calls.
-KSymbol...	Obtains the value of all command line symbols through the knlist command system call.

Examples

1. To obtain the value of the symbols in /unix, enter:

```
trcnm -K environ errno
```

This command sequence displays the following:

```
environ 2FF7FFF8
errno 2FF7FFFC
```

2. To print a symbol table for system calls, enter:

```
trcnm
```

A list similar to the following is generated:

```
pin_obj_start    00000000
header_offset    00000008
ram_disk_start   0000000C
ram_disk_end     00000010
dbg_avail        00000014
base_conf_start  00000018
base_conf_end    0000001C
base_conf_disk   00000020
pin_com_start    00000024
start            00000028
ipl_cb           00000028
...
```

Files

Item	Description
/var/adm/ras/trcfile	Contains the default log file.
/tlo-tvl2/trcnam	Contains the trcnm command.
/etc/trcfmt	Contains the trace format file.

trcrpt Command

Purpose

Formats a report from the trace log.

Syntax

```
trcrpt [ -c ] [ -C [ processorList | all ] ] [ -d List ] [ -D Event-group-list ] [ -e Date ] [ -G ] [ -h ] [ -j ]  
[ -k List ] [ -K Group-list ] [ -m ] [ -n Name ] [ -o File ] [ -p List ] [ -r ] [ -s Date ] [ -t File ] [ -T List ] [ -v ]  
[ -O Options ] [ -x ] [ -@ WparList ] [ -M common | rare | all[:LMT_dir] ] [ -l ComponentList | all[:CT_dir] ]  
[ FileOrDirectory ]
```

Description

The **trcrpt** command reads the trace log specified by the **-M**, **-l** and *File* or *Directory* parameters, formats the trace entries, and writes a report to standard output. The default file from which the system generates a trace report is the **/var/adm/ras/trcfile** file, but you can specify an alternate log file using the **-M**, **-l** and *File* or *Directory* parameters. You can specify one or more files or directories. If you specify a file, it must be a valid trace log file, which is any file that is produced by a trace-related command. If you specify a directory, it must contain a component trace master file. If you specify the **-m** flag, all specified traces will be merged in chronological order.

To include trace entries in a report for the specified workload partition (WPAR), use the **-@** flag.

In AIX 6.1 and later, four-hex-digit hook IDs can be displayed. However, if a four-hex-digit hook ID has a digit of zero, the zero is removed to display only three hex digits. This occurs because four-hex-digit hook IDs in the form **hhh0** are equivalent to three-hex-digit hook IDs in the form **hhh**.

You can use the System Management Interface Tool (SMIT) to run the **trcrpt** command by entering the SMIT fast path:

```
smit trcrpt
```

Flags

Item	Description
-@ <i>WparList</i>	Generates a report containing events that occurred on the workload partitions that you specified. You can specify a list of WPAR configured IDs (CID) or a list of WPAR names with the <i>WparList</i> parameter. The list items can either be separated by commas or enclosed in quotation marks and separated by commas or spaces. Specify 0 or Global in the list to include the Global system in the report.
-c	Checks the template file for syntax errors.
-C [<i>processorList</i> all]	Generates a report containing events that occur on the processors specified. The processors can be separated by commas, or enclosed in double quotation marks and separated by commas or blanks. To report on all processors, specify -C all . The -C flag is not necessary unless you want to see only a subset of the processors traced, or have the processor number show up in the report. If -C is not specified, and the trace is a multi-processor trace, trcrpt generates the trace report for all processors, but the processor number is not shown for each hook unless you specify -O cpuid=on .

Item	Description
-d <i>List</i>	Limits the report to hook IDs specified with the <i>List</i> variable. The <i>List</i> parameter items can be separated by commas or enclosed in double quotation marks and separated by commas or blanks. In AIX 6.1 and later, four-hex-digit hook IDs can be displayed. However, if a four-hex-digit hook ID has a digit of zero, the zero is removed to display only three hex digits. This occurs because four-hex-digit hook IDs in the form hhho are equivalent to three-hex-digit hook IDs in the form hhh .
-D <i>Event-group-list</i>	Limits the report to hook IDs in the <i>Event groups list</i> , plus any hook IDs specified with the -d flag. The list parameter items can be separated by commas or enclosed in double quotation marks and separated by commas or blanks. <i>Event groups</i> are described in Debug and Performance Tracing . The -D flag also reports the trace utility hook id for LMT Restart and LMT Suspend.
-e <i>Date</i>	Ends the report time with entries on, or before, the specified date. The <i>Date</i> variable has the form <i>mmddhhmmssyy</i> (month, day, hour, minute, second, and year). Date and time are recorded in the trace data only when trace data collection is started and stopped. If you stop and restart trace data collection multiple times during a trace session, date and time are recorded each time you start or stop a trace data collection. Use this flag in combination with the -s flag to limit the trace to data collected during a certain time interval. Restriction: The -e and -s flags are only valid for trace log files collected without the <code>trace -C</code> flag.
-G	Lists all event groups. The list of groups, the hook ids in each group, and each group's description is listed to standard output.
-h	Omits the header information from the trace report and writes only formatted trace entries to standard output.
-j	Displays the list of hook IDs. The trcrpt -j command can be used with the trace -j command that includes IDs of trace events or the trace -k command that excludes IDs of trace events.

Item	Description
-k <i>List</i>	Excludes from the report hook IDs specified with the <i>List</i> variable. The <i>List</i> parameter items can be separated by commas or enclosed in double quotation marks and separated by commas or blanks. In AIX 6.1 and above, specifying a two-digit hook ID in the hh form results in hh00, hh10,...,hhF0. Specifying a three-digit hook ID in the hhh form results in hhh0. Specifying a four-digit hook ID in the hhhh form results in hhhh.
-K <i>Event-group-list</i>	Excludes from the report hook IDs in the <i>event-groups</i> list, plus any hook IDs specified with the -k flag. List parameter items can be separated by commas or enclosed in double quotation marks and separated by commas or blanks. Event groups are described in Debug and Performance Tracing.
-l <i>ComponentList</i>	Generates a report for a multi-component trace with <code>ctctl</code> -D or trcdead. The components can be separated by commas, or enclosed in double quotation marks and separated by commas or blanks. The -l flag is not necessary unless you want to see only a subset of the components traced. If -l is not specified, the command assumes the trace is a multi-component trace if a directory is given as input on the command line. Multi-component trace log files not in the default directory must either have their directory specified on the command line or with the <i>CT_dir</i> parameter in conjunction with the -l flag. The -l all option can be used to select all available components. Multiple -l flags can be used to specify components in different directories.
-m	Merges all specified trace files based on time stamps. Files merged from another partition, system or from two or more separate boots of the same system will produce unpredictable results. Without the -m flag, reports for each log file are appended to the specified output file.

Item

-M **common** | **rare** | **all**[:*LMT_dir*]

Description

Generates a report from the LMT log files obtained via the **mtrcsave** or **trcdead** command.

Use the **common** keyword if you only want events from the common LMT buffers to be reported; use the **rare** keyword if you only want events from the rare LMT buffers to be reported; use the **all** keyword if you want common and rare events to be reported.

This flag searches only the default LMT log directory unless the *LMT_dir* parameter is specified. With this parameter, the **trcrpt** command will search for the LMT files in the specified directory rather than the default LMT log directory. To merge common and rare buffers you must use the **all** keyword and the **-m** flag. The **-M** flag can only appear once.

-n *Name*

Specifies the kernel name list file to be used to interpret address for output. Usually, this flag is used when moving a trace log file to another system.

-o *File*

Writes the report to a file instead of to standard output.

-O *Options*

Specifies options that change the content and presentation of the **trcrpt** command. Arguments to the options must be separated by commas or enclosed in double quotation marks and separated by commas or spaces. Valid options are:

2line=[on|off]

Uses two lines per trace event in the report instead of one. The default value is **off**.

component=[on|off]

Displays the full component name in the trace report. The default value is **off**.

cpuid=[on|off]

Displays the physical processor number in the trace report. The default value is **off**.

cid=[on|off]

Displays the workload partition configured ID (CID) in the trace report. The default value is **off**.

endtime=Seconds

Displays the trace report data for events recorded before the seconds specified. Seconds can be given in either an integral or rational representation. If this option is used with the **starttime** option, a specific range can be displayed.

exec=[on|off]

Displays the exec path names in the trace report. The default value is **off**.

Item

Description

filename=[on|off]

Displays the file name from which an event was retrieved. The file name will be truncated from the left if it exceeds 40 characters. The default value is **off**.

hist=[on|off]

Logs the number of instances that each hook ID is encountered. This data can be used for generating histograms. The default value is **off**. This option cannot be run with any other option.

ids=[on|off]

Displays the trace hook identification numbers in the first column of the trace report. The default value is **on**.

pagesize=Number

Controls the number of lines per page in the trace report and is an integer within the range of 0 through 500. The column headings are included on each page. No page breaks are present when the default value of 0 is set.

pid=[on|off]

Displays the process IDs in the trace report. The default value is **off**.

reportedprocessors=[on | off]

Displays the number of processors remaining. This option is only meaningful for a multi-processor trace, `trace -C`. For example, if you are reading a report from a system with 4 processors, and the reported processor's value goes from 4 to 3, then you know that there are no more hooks to be reported for that processor.

PURR=[on | off]

Tells `trcrpt` to show the PURR along with any timestamps. The PURR is displayed following any timestamps.

If the PURR is not valid for the processor traced, the elapsed time is shown instead of the PURR. If the PURR is valid, or the `cpuid` is unknown, but wasn't traced for a hook, the PURR field contains asterisks (*).

Item

Description

removedups=[on | off]

Enables duplicate event detection. A count in the DUPS column displays the number of events that each event in the report represents. If this option is set to **off**, duplicate event detection will be disabled. The default value is **on**. This option is only valid when merging log files via the **-m** flag. Duplicate entries can only be detected when the processor ID is known from the trace entry itself, not when it must be inferred. The processor ID can be obtained from the entry in the following cases:

- A lightweight memory trace
- A multi-processor system trace, where the **trace -C** command option was used
- A 64-bit system trace initiated with the **-p** option
- A 64-bit component trace.

wparname= [on | off]

Displays the workload partition names in the trace report. The default value is **off**.

Item

Description

starttime=Seconds

Displays trace report data for events recorded after the seconds specified. The specified seconds are from the beginning of the trace file. Seconds can be given in either an integral or rational representation. If this option is used with the **endtime** option, a specific range of seconds can be displayed.

svc=[on|on_noblank|off]

Displays the value of the system call in the trace report. The default value is **off**.

This option can have following values:

on

Prints the name of the current system call in the trace report.

on_noblank

Prints the - - - - string in the trace report when the svc option is not set.

off

Does not print any information that is related to the system call.

tid=[on|off]

Displays the thread ID in the trace report. The default value is **off**.

timestamp=[0|1|2|3|4]

Controls the reporting of the time stamp associated with an event in the trace report. The possible values are:

0

Time elapsed since the trace was started and delta time from the previous event. The elapsed time is in seconds and the delta time is in milliseconds. Both values are reported to the nearest nanosecond. This is the default.

1

Short elapsed time. Reports only the elapsed time (in seconds) from the start of the trace. Elapsed time is reported to the nearest microsecond.

2

Microsecond delta time. This is like 0, except the delta time is in microseconds, reported to the nearest microsecond.

3

No time stamp.

4

Raw timestamp from the trace event.

Item	Description
-p <i>List</i>	Reports the process IDs for each event specified by the <i>List</i> variable. The <i>List</i> variable may be a list of process IDs or a list of process names. List items that start with a numeric character are assumed to be process IDs. The list items can be separated by commas or enclosed in double quotation marks and separated by commas or blanks.
-r	Outputs unformatted (raw) trace entries and writes the contents of the trace log to standard output one entry at a time. Use the -h flag with the -r flag to exclude the heading. To get a raw report for processors in a multi-processors trace, use both the -r and -C flags.
-s <i>Date</i>	Starts the report time with entries on, or before, the specified date. The <i>Date</i> variable has the form <i>mmddhhmmssyy</i> (month, day, hour, minute, second, and year). Date and time are recorded in the trace data only when trace data collection is started and stopped. If you stop and restart trace data collection multiple times during a trace session, date and time are recorded each time you start or stop a trace data collection. Use this flag in combination with the -e flag to limit the trace to data collected during a certain time interval. Restriction: The -e and -s flags are only valid for trace log files collected without the <code>trace -C</code> flag.
-t <i>File</i>	Uses the file specified in the <i>File</i> variable as the template file. The default is the /etc/trcfmt file.
-T <i>List</i>	Limits the report to the kernel thread IDs specified by the <i>List</i> parameter. The list items are kernel thread IDs separated by commas or enclosed in double quotation marks and separated by commas or spaces. Starting the list with a kernel thread ID limits the report to all kernel thread IDs in the list. Starting the list with a ! (exclamation point) followed by a kernel thread ID limits the report to all kernel thread IDs not in the list.
-v	Prints file names as the files are opened. Changes to verbose setting.
-x	Displays the exec path name and value of the system call.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

To run the **trcrpt** command on channel 0, you must have the following additional authorizations, if channel 0 **trcrpt** restriction is enabled in the **trcctl** command:

Item	Description
aix.ras.trace.trcrptch0	Required to run the trcrpt command on channel 0.

Note: By default, the root and system group users are privileged users.

To perform all functionality of all commands including the **trcrpt** command on channel 0, you must have the **aix.ras.trace** authorization.

Examples

1. To format the trace log file and print the result, enter:

```
trcrpt | qprt
```

2. To send a trace report to the **/tmp/newfile** file, enter:

```
trcrpt -o /tmp/newfile
```

3. To display process IDs and exec path names in the trace report, enter:

```
trcrpt -O pid=on,exec=on
```

4. To create trace ID histogram data, enter:

```
trcrpt -O hist=on
```

5. To produce a list of all event groups, enter:

```
trcrpt -G
```

The format of this report is shown under the **trcevggrp** command.

6. To generate back-to-back LMT reports from the common and rare buffers, enter:

```
trcrpt -M all
```

7. If, in the above example, the LMT files reside at **/tmp/mydir**, and we want the LMT traces to be merged, enter:

```
trcrpt -m -M all:/tmp/mydir
```

8. To merge the system trace with the **scdisk.hdisk0** component trace, enter:

```
trcrpt -m -l scdisk.hdisk0 /var/adm/ras/trcfile
```

9. To merge LMT with the system trace while not eliminating duplicate events, enter:

```
trcrpt -O removedups=off -m -M all /var/adm/ras/trcfile
```

10. To merge all component traces in **/tmp/mydir** with the LMT traces in the default LMT directory while showing the source file for each trace event, enter:

```
trcrpt -O filename=on -m -M all /tmp/mydir
```

Tip: This is equivalent to the following command:

```
trcrpt -O filename=on -m -M all -l all:/tmp/mydir
```

Tip: If the traces are from a 64-bit kernel, duplicate entries will be removed. However, on the 32-bit kernel, duplicate entries will not be removed since we do not know the processor IDs of the entries in the components traces.

Files

Item	Description
<code>/usr/bin/trcrpt</code>	Contains the trcrpt command.
<code>/var/adm/ras/trcfile</code>	Contains the default log file.
<code>/var/adm/ras/mtrcdir</code>	Location of the default LMT dump directory
<code>/var/adm/ras/trc_ct</code>	Location of the default CT dump directory.
<code>/etc/trcfmt</code>	Contains the trace format file.

trcstop Command

Purpose

Stops the trace function.

Syntax

trcstop [**-<channel>**][**-s** | **-d**]

Description

The **trcstop** command ends a trace session.

You can use the System Management Interface Tool (SMIT) to run the **trcstop** command. To use SMIT, enter:

```
smit trcstop
```

Flags

Item	Description
-<channel>	Specifies the channel which stop the trace. The valid value range is from 0-7. If unspecified, then the default value is 0.
-s	Enables the serialization of trace I/O from multiple processor buffers into the trace file during the <i>tracestop</i> operation. The -s flag is mutually exclusive with the -d flag. Note: The serial -s option is available for all modes (single, circular, and alternate). In previous releases, the -s option was available only for circular mode.
-d	Discards any captured trace buffers which are yet to be written into the file.

Example

To terminate the trace background process, enter:

```
trcstop
```

File

Item	Description
<code>/usr/bin/trcstop</code>	Contains the trcstop command.

trcupdate Command

Purpose

Adds, replaces, or deletes trace report format templates.

Syntax

trcupdate [**-o**] [**-t** *File*] [**-v**] [**-x** *IDList*] [*File*]

Description

The **trcupdate** command adds, replaces, or deletes trace report format templates in the **/etc/trcfmt** or the **/etc/trcfmt.Z** file. When the **/etc/trcfmt.Z** file is used, the **trcupdate** command uncompresses the file, updates it, and recompresses it. The **trcupdate** command creates an "undo" file named **File.undo.trc** in the specified directory.

The **trcupdate** command adds the extension **.trc** to the file name and reads update commands from that file. The undo file is input to the **trcupdate** command if the **-o** (override) flag is specified. When the **-o** flag is specified, the **trcupdate** command undoes the changes previously made to the file.

The first field of each template contains an operator:

Item	Description

+	The plus sign indicates that a template is to be added or replaced. The field that follows this operator contains the template to be replaced.
-	The minus sign indicates that a template is to be deleted. The field after this operator contains the hook ID of the template to delete. Operations are performed in the order in which they appear.

The input to the **trcupdate** command must contain the following as the first line:

```
* /etc/trcfmt
```

The following is a sample trace file:

```
* /etc/trcfmt
+ 15A 1.0 new_fmt
- 1B3
- A14
```

When adding or replacing, the **trcupdate** command compares the version numbers of each input template with the version number of the template with the same hook ID. If the version number of the input template is greater than or equal to the version of the existing template, the **trcupdate** command replaces the old template with the input template. If a template does not exist, then the input template is added to the file.

The **trcupdate** command will not modify the **/etc/trcfmt** file if a syntax error is detected in the update file.

Flags

Item	Description
-o	Overrides the old template with the input template without verifying the version number of either template.
-t <i>File</i>	Specifies a file, instead of the /etc/trcfmt or the /etc/trcfmt.Z file, to be used as the template file.

Item	Description
-v	Prints the file names as each file is opened.
-x <i>IDList</i>	Extracts the templates specified in the <i>IDList</i> from the template file and writes them to standard output. The <i>IDList</i> parameter lists the hook IDs.

Security

Access Control: None, but you must have write authority to the template file you are changing. The default is **/etc/trcfmt**.

Examples

1. To add a template, enter the following command:

```
trcupdate
* /etc/trcfmt
+ 15A 1.0 new_fmt
```

Tip: In AIX 6.1 and later versions, this is equivalent to:

```
trcupdate
* /etc/trcfmt
+ 15A0 1.0 new_fmt
```

2. To delete a template, type the following command:

```
trcupdate
* /etc/trcfmt
- 15A 1.0 new_fmt
```

Tip: In AIX 6.1 and later versions, this is equivalent to:

```
trcupdate
* /etc/trcfmt
- 15A0 1.0 new_fmt
```

3. To replace a template, enter the following command:

```
trcupdate
* /etc/trcfmt
+ 15A 1.0 new_fmt
```

Tip: In AIX 6.1 and later versions, this is equivalent to:

```
trcupdate
* /etc/trcfmt
+ 15A0 1.0 new_fmt
```

4. In AIX 6.1 and later versions, to add a template for hook ID 0AB0, enter the following command:

```
trcupdate
* /etc/trcfmt
+ 0AB0 1.0 new_fmt
```

The above command is equivalent to the following command:

```
trcupdate
* /etc/trcfmt
+0AB 1.0 new_fmt
```


5. In AIX 6.1 and above, to add a template for hook ID 1AB1, enter the following command:

```
trcupdate
* /etc/trcfmt
+ 1AB1 1.0 new_fmt
```

Files

Item	Description
<code>/usr/bin/trcupdate</code>	Contains the trcupdate command.
<code>/etc/trcfmt</code>	Contains the trace format file.
<code>/usr/include/sys/trcmacros.h</code>	Defines trchhook and utrchhook macros.

troff Command

Purpose

Formats text for printing on typesetting devices.

Syntax

```
troff [ -a ] [ -i ] [ -q ] [ -z ] [ -F Directory ] [ -n Number ] [ -o List ] [ -r ANumber ]  
[ -s Number ] [ -T Name ] [ -mm | -me | -mptx | -ms | -man | -mv ] [ -M Media ] [ File ... | - ]
```

Description

The **troff** command reads one or more files and formats the text for printing on a phototypesetter or comparable device. A postprocessor is then required to post process the output of the **troff** command to the target device. See the accompanying [example](#).

If no file is specified or oh.. the - (minus) flag is not the last parameter, standard input is read by default.

For the 3812, 3816, and Hewlett-Packard LaserJet Series II printer, the default fonts are the native fonts for the printer. Additional fonts also are available for these printers, which can be loaded through the use of the **troff .fp** directive. These fonts are stored on the host in the directory `/usr/lib/font/devPrinter/bitmaps`, and downloaded to the printer as necessary.

Typefaces

Three different typefaces are provided in four styles. The following chart shows the relationship between typeface, style, and the name that the **troff** command uses to access the font.

Note: The fonts in this set are based on the Computer Modern letter forms developed by Donald E Knuth. (Refer to Knuth, Donald: *Computer Modern Typefaces*. Addison-Wesley, 1986.)

Typeface	Regular	Italic	Bold	Italic
Roman	cr	cR	Cr	CR
Sans Serif	cs	cS	Cs	CS
Typewriter	ct	cT	Ct	CT
troff special sp				

These fonts are all provided in the standard 15 troff sizes: 6, 7, 8, 9, 10, 11, 12, 14, 16, 28, 20, 22, 24, 28, and 36 points.

For example, `.fp 1 Cr` loads the Roman bold font into position 1.

Note: The `.tl` request cannot be used before the first break-producing request in the input to the **troff** command.

Flags

Item	Description
-a	Sends a printable ASCII approximation of the results to standard output.
-F <i>Directory</i>	Accesses font information from the <i>Directory</i> / devName directory instead of the default /usr/lib/font/devName directory (where <i>Name</i> is specified by the -T flag).
-i	Reads standard input after there are no more files.
-M <i>Media</i>	Specifies a paper size in order to determine the amount of imageable area on the paper. Valid values for the <i>Media</i> variable are: A4 Specifies a paper size of 8.3 X 11.7 inches (210 X 297 mm). A5 Specifies a paper size of 5.83 X 8.27 inches (148 X 210 mm). B5 Specifies a paper size of 6.9 X 9.8 inches (176 X 250 mm). EXEC Specifies a paper size of 7.25 X 10.5 inches (184.2 X 266.7 mm). LEGAL Specifies a paper size of 8.5 X 14 inches (215.9 X 355.6 mm). LETTER Specifies a paper size of 8.5 X 11 inches (215.9 X 279.4 mm). This is the default value. Note: The <i>Media</i> variable is not case-sensitive.
-n <i>Number</i>	Numbers the first printed page with the value specified by the <i>Number</i> variable.
-o <i>List</i>	Prints only pages specified by the <i>List</i> variable, which consists of a comma-separated list of page numbers and ranges: • A range of <i>Start-Stop</i> means print pages <i>Start</i> through <i>Stop</i>. For example: 9-15 prints pages 9 through 15. • An initial <i>-Stop</i> means print from the beginning to page <i>Stop</i>. • A final <i>Start-</i> means print from page <i>Start</i> to the end. • A combination of page numbers and ranges prints the specified pages. For example: -3, 6-8, 10, 12- prints from the beginning through page 3, pages 6 through 8, page 10, and page 12 to the end. Note: When this flag is used in a pipeline (for example, with one or more of the pic , eqn , or tbl commands), you might receive a broken pipe message if the last page in the document is not specified in the <i>List</i> variable. This broken pipe message is not an indication of any problem and can be ignored.
-q	Calls the simultaneous input and output mode of the .rd request.
-r <i>A</i> <i>Number</i>	Sets the register specified by the <i>A</i> variable to the specified number. The <i>A</i> variable value must have a one-character ASCII name.
-s <i>Number</i>	Generates output to make the typesetter stop every specified number of pages.

Item	Description
-TName	Prepares the output for the specified printing device. Phototypesetters or comparable printing devices use the following <i>Name</i> variables for operating system international extended characters. The default is ibm3816. Note: You get a message that reads <code>bad point size</code> if your device does not support the point size that you specified. The troff command uses the closest valid point size to continue formatting. canonls Canon Lasershot LBP-B406S/D/E,A404/E,A304E. ibm3812 3812 Pageprinter II. ibm3816 3816 Pageprinter. hplj Hewlett-Packard LaserJet II. ibm5585H-T 5585-H01 Traditional Chinese Language support. ibm5587G 5587-G01, 5584-H02, 5585-H01, 5587-H01, and 5589-H01 Kanji Printer multibyte language support. psc PostScript printer. X100 AIXwindows display. Note: You also can set the TYPESETTER environment variable to one of the preceding values instead of using the -TName flag of the troff command.
-man	Selects the man macro processing package.
-me	Selects the me macro processing package.
-mm	Selects the mm macro processing package.
-mptx	Selects the mptx macro processing package.
-ms	Selects the ms macro processing package.
-mv	Selects the mv macro processing package.

See [Macro Packages for Formatting Tools](#) for more information on the macros.

Item Description

- z** Prints only messages generated by **.tm** (workstation message) requests.
- Forces input to be read from standard input.

Environment Variables

Item	Description
TYPESETTER	Contains information about a particular printing device.

Examples

The following is an example of the **troff** command:

Macro Packages for Formatting Tools

The following macro packages are part of the Formatting Tools in the Text Formatting System and are described in more detail on the next pages:

Item	Description
<u>man</u>	Enables you to create your own manual pages from online manual pages.
<u>me</u>	Provides macros for formatting papers.
<u>mm</u>	Formats documents with nroff and troff formatters.
<u>mptx</u>	Formats a permuted index.
<u>ms</u>	Provides a formatting facility for various styles of articles, theses, and books.
<u>mv</u>	Typesets English-language view graphs and slides by using the troff command.

man Macro Package for the nroff and troff Commands

The **man** macro package is provided to enable users to create their own manual pages from online manual pages that have been processed with either the **nroff** command or **troff** command. The **man** macro package is used with either the **nroff** command or the **troff** command.

Special macros, strings, and number registers exist, internal to the **man** macro package, in addition to the following lists of [format macros](#), [strings](#), and [registers](#). Except for the names predefined by the **troff** command and the **d**, **m**, and **y** number registers, all such internal names are of the form *SymbolAlpha*, where *Symbol* is one of **),** **l,** or **},** and *Alpha* is any alphanumeric character.

The **man** macro package uses only the Roman font. If the input text of an entry contains requests for other fonts (for example, the **.I** format macro, **.RB** request, or **\fI** request) the corresponding fonts must be mounted.

Format Macros

The following macros are used to alter the characteristics of manual pages that are formatted using the **man** macro package.

Type font and size are reset to default values before each paragraph and after processing font- and size-setting macros (for example, the **.I** format macro, **.SM** format macro, and **.B** format macro).

Tab stops are neither used nor set by any of the format macros except the **.DT** format macro and the **.TH** format macro.

.B [*Text*]

Makes text bold.

The *Text* variable represent up to six words; use “ ” (double quotation marks) to include character spaces in a word. If the variable is empty, this treatment is applied to the next input text line that contains text to be printed. For example, use the **.I** format macro to italicize an entire line, or use the **.SM** and **.B** format macros to produce an entire line of small-bold text. By default, hyphenation is turned off for the **nroff** command, but remains on for the **troff** command.

.DT

Restores default tab settings every 5 ens for the **nroff** command and every 7.2 ens for the **troff** command.

.HP [*Indent*]

Begins a paragraph with a hanging indent as specified by the *Indent* variable.

If the *Indent* variable is omitted, the previous *Indent* value is used. This value is set to its default (5 ens for the **nroff** command and 7.2 ens for the **troff** command) by the **.TH** format macro, **.P** format

macro, and **.RS** format macro, and restored by the **.RE** format macro. The default unit for *Indent* is ens.

.I [Text]

Makes text italic.

The *Text* variable represent up to six words; use “ ” (double quotation marks) to include character spaces in a word. If the variable is empty, this treatment is applied to the next input text line that contains text to be printed. For example, use the **.I** format macro to italicize an entire line, or use the **.SM** and **.B** format macros to produce an entire line of small-bold text. By default, hyphenation is turned off for the **nroff** command, but remains on for the **troff** command.

.IP [Tag] [Indent]

Same as the **.TP** *Indent* macro with the *Tag* variable; if the value of the *Tag* variable is **NULL**, begin indented paragraph. This macro is often used to get an indented paragraph without a tag.

If the *Indent* variable is omitted, the previous *Indent* value is used. This value is set to its default (5 ens for the **nroff** command and 7.2 ens for the **troff** command) by the **.TH** format macro, **.P** format macro, and **.RS** format macro, and restored by the **.RE** format macro. The default unit for *Indent* is ens.

.P

Begins paragraph with normal font, point size, and indent. The **.PP** macro is a synonym for the **mm** macro package **.P** macro.

.PD [Number]

Sets inter-paragraph distance the number of vertical spaces specified by the *Number* parameter. The default *Number* variable value is 0.4v for the **troff** command and 1v for the **nroff** command.

.PM [Indicator]

Sets proprietary marking as follows:

Indicator	Marking
P	PRIVATE
N	NOTICE
No <i>Indicator</i> specified	Turns off proprietary marking.

.RE [Number]

Ends relative indent (**.RS**) at indent level position specified by the *Number* variable. If the *Number* variable value is omitted, return to the most recent lower indent level.

.RI Character1Character2...

Concatenates the Roman *Character1* with the italic *Character2*; alternate these two fonts up to six sets of *Character1Character2*. Similar macros alternate between any two of Roman, italic, and bold: the **.IR**, **.RB**, **.BR**, **.IB**, and **.BI** macros.

.RS [Indent]

Increases relative indent (initially zero). Indent all output an extra number of units from the left margin as specified by the *Indent* variable.

If the *Indent* variable is omitted, the previous *Indent* value is used. This value is set to its default (5 ens for the **nroff** command and 7.2 ens for the **troff** command) by the **.TH** format macro, **.P** format macro, and **.RS** format macro, and restored by the **.RE** format macro. The default unit for *Indent* is ens.

.SH [Text]

Places subhead text.

The *Text* variable represent up to six words; use “ ” (double quotation marks) to include character spaces in a word. If the variable is empty, this treatment is applied to the next input text line that contains text to be printed. For example, use the **.I** format macro to italicize an entire line, or use

the **.SM** and **.B** format macros to produce an entire line of small-bold text. By default, hyphenation is turned off for the **nroff** command, but remains on for the **troff** command.

.SM [Text]

Makes text one point smaller than default point size.

The *Text* variable represent up to six words; use “ ” (double quotation marks) to include character spaces in a word. If the variable is empty, this treatment is applied to the next input text line that contains text to be printed. For example, use the **.I** format macro to italicize an entire line, or use the **.SM** and **.B** format macros to produce an entire line of small-bold text. By default, hyphenation is turned off for the **nroff** command, but remains on for the **troff** command.

.SS [Text]

Places sub-subhead text.

The *Text* variable represent up to six words; use “ ” (double quotation marks) to include character spaces in a word. If the variable is empty, this treatment is applied to the next input text line that contains text to be printed. For example, use the **.I** format macro to italicize an entire line, or use the **.SM** and **.B** format macros to produce an entire line of small-bold text. By default, hyphenation is turned off for the **nroff** command, but remains on for the **troff** command.

.TH [Title][Section][Commentary][Name]

Sets the title and entry heading. This macro calls the **.DT** format macro.

Variable	Marking
<i>Title</i>	Title
<i>Section</i>	Section number
<i>Commentary</i>	Extra commentary
<i>Name</i>	New manual name.

Note: If the **.TH** format macro values contain character spaces that are not enclosed in “ ” (double quotation marks), irregular dots are displayed on the output.

.TP [Indent]

Begins indented paragraph with hanging tag. The next input line that contains text is the tag. If the tag does not fit, it is printed on a separate line.

If the *Indent* variable is omitted, the previous *Indent* value is used. This value is set to its default (5 ens for the **nroff** command and 7.2 ens for the **troff** command) by the **.TH** format macro, **.P** format macro, and **.RS** format macro, and restored by the **.RE** format macro. The default unit for *Indent* is ens.

Strings

Item	Description
*R	Adds trademark, (Reg.) for the nroff command and the registered trademark symbol for the troff command.
*S	Changes to default type size.
*(Tm	Adds trademark indicator.

Registers

Item	Description
IN	Indent left margin relative to subheads. The default is 7.2 ens for the troff command and 5 ens for the nroff command.
LL	Line length including the value specified by the IN register.

Item	Description
PD	Current inter-paragraph distance.

Flags

Item	Description
-rs1	Reduces default page size of 8.5 inches by 11 inches with a 6.5-inch by 10-inch text area to a 6-inch by 9-inch page size with a 4.75-inch by 8.375-inch text area. This flag also reduces the default type size from 10-point to 9-point and the vertical line spacing from 12-point to 10-point.

Examples

1. To process the file `your.book` and pipe the formatted output to the local line printer, `qprt`, type:

```
nroff -Tlp -man your.book | qprt -dp
```

2. To process the files `my.book` and `dept.book`, which contain tables, and pipe the formatted output to the local line printer, `qprt`, type:

```
tbl my.book dept.book | nroff -Tlp -man | col -Tlp | qprt -dp
```

Note: Before the output is sent to `qprt`, it is first filtered through the **col** command to process reverse linefeeds used by the **tbl** command.

3. To process the file `group`, which contains pictures, graphs, and tables, and prepare the formatted output for processing on the IBM 3816 printer, enter:

```
grap group | pic | tbl | troff -Tibm3816 -man \
| ibm3816 | qprt -dp
```

Note:

1. If manual pages created with the **man** macro package are intended for an online facility, components requiring the **troff** command, such as the **grap** or **pic** command, should be avoided.
2. The **grap** command precedes the **pic** command because it is a preprocessor to the **pic** command; the reverse does not format correctly.
3. The **col** command is not required as a filter to the **tbl** command; typeset documents do not require reverse linefeeds.

me Macro Package for the nroff and troff Commands

The **me** package of the **nroff** and **troff** command macro definitions provides a formatting facility for technical papers in various formats. The **col** command may be required to postprocess **nroff** output in certain cases.

The macro requests are defined in the following section, in **me Requests**. Many **nroff/troff** requests can have unpredictable results in conjunction with this package. However, the following requests can be used after the first **.pp** request:

Item	Description
.bp	Begins new page.
.br	Breaks output line here.
.ce [<i>Number</i>]	Centers next specified number of lines. Default is 1 (one).
.ls [<i>Number</i>]	Sets line spacing. Text is single-spaced if <i>Number</i> is set to 1 (one); double-spaced if the value is set to 2.
.na	Leaves right margin unjustified.

Item	Description
.sp [<i>Number</i>]	Inserts the specified number of spacing lines.
.sz [<i>+</i>] <i>Number</i>	Adds the specified number to point size.
.ul [<i>Number</i>]	Underlines next specified number of lines. Default is 1 (one).

Output of the **eqn**, **neqn**, **refer**, and **tbl** commands preprocessors for equations and tables can be used as input.

me Requests

The following list contains all macros, strings, and number registers available in the **me** macros. Selected **troff** commands, registers, and functions are included.

Item	Description
\(space)	Defines unpadding space (troff command built-in function).
\“	Comments to end of line (troff command built-in function).
*#	Indicates optional delayed text tag string.
\\$Number	Interpolates the value specified by the <i>Number</i> variable (troff command built-in function).
\n(\$0	Defines section depth (number register).
.\$0	Started after section title printed (user-definable macro).
\n(\$1	Defines first section number (number register).
.\$1	Started before printing depth 1 (one) section (user-definable macro).
\n(\$2	Defines second section number (number register).
.\$2	Started before printing depth 2 section (user-definable macro).
\n(\$3	Defines third section number (number register).
.\$3	Started before printing depth 3 section (user-definable macro).
\n(\$4	Defines fourth section number (number register).
.\$4	Started before printing depth 4 section (user-definable macro).
\n(\$5	Defines fifth section number (number register).
.\$5	Started before printing depth 5 section (user-definable macro).
\n(\$6	Defines sixth section number (number register).
.\$6	Started before printing depth 6 section (user-definable macro).
.\$C	Called at beginning of chapter (user-definable macro).
.\$H	Indicates text header (user-definable macro).
\n(\$R	Defines relative vertical spacing in displays (number register defined by default; changing is not recommended).
\n(\$c	Defines current column header (number register).
.\$c	Prints chapter title (macro defined by default; changing is not recommended).
\n(\$d	Indicates delayed text number (number register).
\n(\$f	Indicates footnote number (number register).
.\$f	Prints footer (macro defined by default; changing is not recommended).
.\$h	Prints header (macro defined by default; changing is not recommended).

Item	Description
<code>\n(\$i</code>	Defines paragraph base indent (number register).
<code>\n(\$l</code>	Defines column width (number register).
<code>\n(\$m</code>	Indicates number of columns in effect (number register).
<code>*(\$n</code>	Indicates section name (string).
<code>\n(\$p</code>	Defines numbered paragraph number (number register).
<code>.\$p</code>	Prints section heading (macro defined by default; changing is not recommended).
<code>\n(\$r</code>	Defines relative vertical spacing in text (number register defined by default; changing is not recommended).
<code>\n(\$s</code>	Defines column indent (number register).
<code>.\$s</code>	Separates footnotes from text (macro defined by default; changing is not recommended).
<code>\n%</code>	Defines current page number (number register defined by default; changing is not recommended).
<code>\&</code>	Indicates zero-width character; useful for hiding controls (troff command built-in function).
<code>\(XX</code>	Interpolates special character specified by the <i>XX</i> variable (troff command built-in function).
<code>.(b</code>	Begins block (macro).
<code>.(c</code>	Begins centered block (macro).
Item	Description
<code>.(d</code>	Begins delayed text (macro).
<code>.(f</code>	Begins footnote (macro).
<code>.(l</code>	Begins list (macro).
<code>.(q</code>	Begins quote (macro).
<code>.(xIndex</code>	Begins indexed item in the specified index (macro).
<code>.(z</code>	Begins floating keep (macro).
<code>.)b</code>	Ends block (macro).
<code>.)c</code>	Ends centered block (macro).
<code>.)d</code>	Ends delayed text (macro).
<code>.)f</code>	Ends footnote (macro).
<code>.)l</code>	Ends list (macro).
<code>.)q</code>	Ends quote (macro).
<code>.)x</code>	Ends index entry (macro).
<code>.)z</code>	Ends floating keep (macro).
<code>*String</code>	Interpolates the value specified by the <i>String</i> variable (troff command built-in function).
<code>*String1String2</code>	Interpolates the value specified by the <i>String1String2</i> variable (troff command built-in function).

Item	Description
**	Indicates optional footnote tag string.
.++mH	Macro to define paper section. The value specified by the <i>m</i> variable defines the part of the paper. The <i>m</i> variable can have the following values: C Defines chapter. A Defines appendix. P Defines preliminary information, such as abstract and table of contents. B Defines bibliography. RC Defines chapters to be renumbered from page 1 (one) of each chapter. RA Defines appendix to be renumbered from page 1 (one). The <i>H</i> parameter defines the new header. If there are any spaces in it, the entire header must be quoted. If you want the header to have the chapter number in it, use the string <code>\\n(ch</code> . For example, to number appendixes A.1, A.2, ..., type: <code>.++ RA ' ' '\\n(ch.%'</code> . Each section (such as chapters and appendixes) should be preceded by the .+c request.
.+cTitle	Begins chapter (or appendix, for instance, as set by the .++ macro). The value specified by the <i>Title</i> variable is the chapter title (macro).
*,	Indicates cedilla (string).
\-	Indicates minus sign (troff command built-in function).
*-	Indicates 3/4 em dash (string).
\0	Defines unpaddable digit-width space (troff command built-in function).
.1c	Reverts to single-column output (macro).
.2c	Begins two-column output (macro).
*:	Indicates umlaut (string).
*<	Begins subscript (string).
*>	Ends subscript (string).
.EN	Ends equation. Space after equation produced by the eqn command or neqn command (macro).
.EQXY	Begins equation; breaks out and adds space. The value specified by the <i>Y</i> variable is the equation number. The optional <i>X</i> variable value might be any of the following: I Indents equation (default). L Left-adjusts equation. C Centers equation (macro).
\L'Distance'	Indicates vertical line-drawing function for the specified distance (troff command built-in function).

Item	Description
.PE	Ends pic picture (macro).
.PF	Ends pic picture with flyback (macro).
.PS	Starts pic picture (macro).
.TE	Ends table (macro).
.TH	Ends header of table (macro).
.TS X	Begins table. If the value of the <i>X</i> variable is H , the table has a repeated heading (macro).
*[	Begins superscript (string).
\n(.\$	Defines number of options to macro (number register defined by default; changing is not recommended).
\n(.i	Indicates current indent (number register defined by default; changing is not recommended).
\n(.l	Indicates current line length (number register defined by default; changing is not recommended).
Item	Description
\n(.s	Indicates current point size (number register defined by default; changing is not recommended).
*(4	Indicates acute accent (string).
*(`	Indicates grave accent (string).
\(4	Indicates acute accent (troff command built-in function).
\(`	Indicates grave accent (troff command built-in function).
*]	Ends superscript (string).
\^	Indicates 1/12 em narrow space (troff command built-in function).
*^	Indicates caret (string).
.ac$AuthorNumber$	Sets up for ACM-style output. The <i>Author</i> variable specifies the author name or names. The <i>Number</i> variable specifies the total number of pages. Must be used before the first initialization (macro).
.ad	Sets text adjustment (macro).
.af	Assigns format to register (macro).
.am	Appends to macro (macro).
.ar	Sets page numbers in Arabic (macro).
.as	Appends to string (macro).
.b X	Prints in boldface the value specified by the <i>X</i> variable. If the <i>X</i> variable is omitted, boldface text follows (macro).
.ba +$Number$	Augments the base indent by the specified <i>Number</i> value. Sets the indent on regular text such as paragraphs (macro).
.bc	Begins new column (macro).
.bi X	Prints in bold italic the value specified by the <i>X</i> parameter, in no-fill mode only. If the <i>X</i> parameter is not used, bold italic text follows (macro).
\n(bi	Displays block indent (number register).

Item	Description
.bl	Requests blank lines, even at top of page (macro).
\n(bm	Sets bottom title margin (number register).
.bp	Begins page (macro).
.br	Sets break; starts new line (macro).
\n(bs	Displays block pre- or post-spacing (number register).
\n(bt	Blocks keep threshold (number register).
.bu	Begins bulleted paragraph (macro).
.bx X	Prints in no-fill mode only the value specified by the X variable in box (macro).
\c	Continues input (troff command built-in function).
.ce	Centers lines (macro).
\n(ch	Defines current chapter number (number register).
.de	Defines macro (macro).
\n(df	Displays font (number register).
.ds	Defines string (macro).
\n(dw	Defines current day of week (number register).
*(dw	Defines current day of week (string).
\n(dy	Defines current day of month (number register).
\e	Indicates printable version of \ (backslash) (troff command built-in function).
.ef'X'Y'Z'	Sets even-page footer to the values specified by the XYZ variables (macro).
.eh'X'Y'Z'	Sets even-page header to the values specified by the XYZ variables (macro).
.el	Specifies the else part of an if/else conditional (macro).
.ep	Ends page (macro).

Item	Description
\n(es	Indicates equation pre- or post-space (number register).
\fFont	Sets inline font change to the specified <i>Font</i> variable value (troff command built-in function).
\f(Fontf	Sets inline font change to the specified <i>Fontf</i> variable value (troff command built-in function).
.fc	Sets field characters (macro).
\n(ff	Sets footnote font (number register).
.fi	Fills output lines (macro).
\n(fi	Indicates footnote indent, first line only (number register).
\n(fm	Sets footer margin (number register).
.fo 'X'Y'Z'	Sets footer to the values specified by the XYZ variables (macro).
\n(fp	Sets footnote point size (number register).
\n(fs	Sets footnote pre-space (number register).
\n(fu	Sets footnote indent from right margin (number register).

Item	Description
<code>\h'Distance'</code>	Sets local horizontal motion for the specified distance (troff command built-in function).
<code>.hc</code>	Sets hyphenation character (macro).
<code>.he 'X'Y'Z'</code>	Sets header to the values specified by the XYZ variables (macro).
<code>.hl</code>	Draws horizontal line (macro).
<code>\n(hm</code>	Sets header margin (number register).
<code>.hx</code>	Suppresses headers and footers on next page (macro).
<code>.hy</code>	Sets hyphenation mode (macro).
<code>.i X</code>	Italicizes the value specified by the X variable. If the Xvariable is omitted, italic text follows (macro).
<code>.ie</code>	Specifies the else part of an if/else conditional (macro).
<code>.if</code>	Designates a conditional (macro).
<code>\n(ii</code>	Sets indented paragraph indent (number register).
<code>.in</code>	Indents (transient); use the <code>.ba</code> macro if pervasive (macro).
<code>.ip X Y</code>	Starts indented paragraph, with hanging tag specified by the X variable. Indentation is the en value specified by the Y variable. Default is 5 (macro).
<code>.ix</code>	Indents, no break (macro).
<code>\l'Distance'</code>	Starts horizontal line-drawing function for the specified distance (troff command built-in function).
<code>.lc</code>	Sets leader repetition character (macro).
<code>.lh</code>	Interpolates local letterhead (macro).
<code>.ll</code>	Sets line length (macro).
<code>.lo</code>	Reads in a file of local macros of the form <code>. *x</code> . Must be used before initialization (macro).
<code>.lp</code>	Begins left-justified paragraph (macro).
<code>*(lq</code>	Designates left quotation marks (string).
<code>.ls</code>	Sets multi-line spacing (macro).
<code>.m1</code>	Sets space from top of page to header (macro).
<code>.m2</code>	Sets space from header to text (macro).
<code>.m3</code>	Sets space from text to footer (macro).
<code>.m4</code>	Sets space from footer to bottom of page (macro).
<code>.mc</code>	Inserts margin character (macro).
<code>.mk</code>	Marks vertical position (macro).
<code>\n(mo</code>	Defines month of year (number register).
Item	Description
<code>*(mo</code>	Defines current month (string).
<code>\nX</code>	Interpolates number register specified by the X variable value (number register).
<code>\n(XX</code>	Interpolates number register specified by the XX variable (number register).

Item	Description
.n1	Sets number lines in margin (macro).
.n2	Sets number lines in margin (macro).
.na	Turns off text adjustment (macro).
.neNumber	Sets the specified number of lines of vertical space (macro).
.nf	Leaves output lines unfilled (macro).
.nh	Turns off hyphenation (macro).
.np	Begins numbered paragraph (macro).
.nr	Sets number register (macro).
.ns	Indicates no-space mode (macro).
*o	Indicates superscript circle (such as for Norse A; string).
.of'X'Y'Z'	Sets odd footer to the values specified by the XYZ variables (macro).
.oh'X'Y'Z'	Sets odd header to the values specified by the XYZ variables (macro).
.pa	Begins page (macro).
.pd	Prints delayed text (macro).
\n(pf	Indicates paragraph font (number register).
\n(pi	Indicates paragraph indent (number register).
.pl	Sets page length (macro).
.pn	Sets next page number (macro).
.po	Sets page offset (macro).
\n(po	Simulates page offset (number register).
.pp	Begins paragraph, first line indented (macro).
\n(pp	Sets paragraph point size (number register).
\n(ps	Sets paragraph pre-space (number register).
.q	Indicates quoted (macro).
*(qa	For all (string).
*qe	There exists (string).
\n(qi	Sets quotation indent; also shortens line (number register).
\n(qp	Sets quotation point size (number register).
\n(qs	Sets quotation pre- or post-space (number register).
.r	Sets Roman text to follow (macro).
.rb	Sets real bold font (macro).
.re	Resets tabs to default values (macro).
.rm	Removes macro or string (macro).
.rn	Renames macro or string (macro).
.ro	Sets page numbers in Roman (macro).
*(rq	Indicates right quotation marks (string).
.rr	Removes register (macro).

Item	Description
.rs	Restores register (macro).
Item	Description
.rt	Returns to vertical position (macro).
\sSize	Changes inline size to specified size (troff command built-in function).
.sc	Reads in a file of special characters and diacritical marks. Must be used before initialization (macro).
\n(sf	Sets section title font (number register).
.shLevelTitle	Indicates section head to follow; font automatically bold. The <i>Level</i> variable specifies the level of section. The <i>Title</i> variable specifies the title of section (macro).
\n(si	Sets relative base indent-per-section depth (number register).
.sk	Leaves the next page blank. Only one page is remembered ahead (macro).
.smX	Sets, in a smaller point size, the value specified by the <i>X</i> variable (macro).
.so	Indicates source input file (macro).
\n(so	Sets additional section title offset (number register).
.sp	Indicates vertical space (macro).
\n(sp	Indicates section title point size (number register).
\n(ss	Indicates section prespace (number register).
.sx	Changes section depth (macro).
.sz +Number	Augments point size by the specified number of points (macro).
.ta	Sets tab stops (macro).
.tc	Sets tab repetition character (macro).
*(td	Sets today's date (string).
n(tf	Indicates title font (number register).
.th	Produces paper in thesis format. Must be used before initialization (macro).
.ti	Indicates temporary indent, next line only (macro).
.tl	Indicates 3-part title (macro).
\n(tm	Sets top title margin (number register).
.tp	Begins title page (macro).
\n(tp	Sets title point size (number register).
.tr	Translates (macro).
.u X	Underlines the value specified by the <i>X</i> variable, even in the troff command. No-fill mode only (macro).
.uh	Sets section head to follow; font automatically bold. Similar to the .sh macro, but unnumbered (macro).
.ul	Underlines next line (macro).
\v'Distance'	Local vertical motion for the specified distance (troff command built-in function).
*v	Inverts v for Czech e (string).

Item	Description
<code>\w'<i>String</i>'</code>	Returns width of the specified string (troff command built-in function).
<code>.xl</code>	Sets local line length (macro).
<code>.xp<i>Index</i></code>	Prints the specified index (macro).
<code>\n(xs</code>	Sets index entry prespace (number register).
<code>\n(xu</code>	Sets index indent, from right margin (number register).
<code>\n(yr</code>	Indicates year, last two digits only (number register).
<code>\n(zs</code>	Sets floating keep pre- or post-space (number register).
<code>\{</code>	Begins conditional group (troff command built-in function).
<code>\ </code>	1/6 em, narrow space (troff command built-in function).
<code>\}</code>	Ends conditional group (troff command built-in function).
<code>*~</code>	Indicates tilde (string).

For further information, see the *-ME Reference Manual* by E. P. Allman.

mm Macro Package for the mm, mmt, nroff, and troff Commands

The **mm** macro package provides macros to format text in a wide variety of document forms, such as memos, letters, and reports. The manner in which you type and edit a document is essentially independent of whether the document is later formatted at a terminal or phototypeset.

The **col** command may be required to postprocess **nroff** output. See the **col** command for specific requirements.

The **mm** macros and additional information are summarized under the following headings:

- [Beginning Macros for Formal Memoranda](#)
- [Business Letter Macros](#)
- [Ending Macros \(Trailing Information\)](#)
- [Paragraphs](#)
- [Section Headings](#)
- [Lists](#)
- [Displays, Tables, Equations, and Footnotes](#)
- [Page Headers and Footers](#)
- [Miscellaneous Macros](#)
- [mm Registers](#)
- [mm Strings](#)
- [String Names](#)
- [Reserved Names.](#)

Beginning Macros for Formal Memoranda

Item	Description
<code>.ND <i>Date</i></code>	Sets new date.
<code>.TL [<i>ChgNumber</i>] [<i>FileNumber</i>]</code>	Sets title information. Text on the following line is used as the title of the document.
<code>.AF [<i>CompanyName</i>]</code>	Specifies author's company name.

Item	Description
.AU <i>Name</i> [<i>Initials</i>] [<i>Loc</i>] [<i>Dept</i>] [<i>Ext</i>] [<i>Room</i>] [<i>Option...</i>]	Sets author information.
.AT <i>AuthorTitle</i> [...]	Specifies title to follow signer's name (up to nine options).
.TM [<i>Number</i>]	Sets technical memorandum number.
.AS [0 1 2] [<i>Indent</i>]	Starts abstract, for technical memorandum and released paper only: 0 Abstract on cover sheet and first page 1 Abstract only on cover sheet 2 Abstract only on memorandum for file cover sheet.
.AE	Ends abstract.
.NS	Starts notation, allowed on memorandum for file cover sheets following an .AS 2/.AE macro pair (see "Ending Macros").
.NE	Ends notation, allowed on memorandum for file cover sheets following an .AS 2/.AE macro pair (see "Ending Macros").
.OK [<i>Keyword ...</i>]	Specifies other keywords (up to nine options).
.MT [<i>type</i>] [<i>title</i>]	Sets document type: "" No type. 0 No type (internal letter). 1 Memorandum for file. 2 Programmer's notes. 3 Engineer's notes. 4 Released paper. 5 External letter. "String" The specified string is printed.
<i>Title</i>	User-supplied text prefixed to page number

Business Letter Macros

Item	Description
.WA	Starts writer's address.

Item	Description
.WE	Ends writer's address.
.LO CN [<i>Notation</i>]	Specifies confidential notation.
.LO RN [<i>Notation</i>]	Specifies reference notation.
.IA	Starts inside (recipient's) address.
.IE	Ends inside (recipient's) address.
.LO AT [<i>Notation</i>]	Specifies attention line.
.LO SA [<i>Notation</i>]	Specifies salutation.
.LO SJ [<i>Notation</i>]	Specifies subject line.
.LT [{ none BL SB FB SP }]	Specifies business letter type:
	none Blocked
	BL Blocked
	SB Semiblocked
	FB Full-Blocked
	SP Simplified.

Ending Macros (Trailing Information)

Item	Description
.FC [<i>Closing</i>]	Prints formal closing.
.SG [<i>Initials</i>] [1]	Prints signature line.

Item	Description
.NS [{"0 1 2 3 4 5 6 7 8 9 10 11 12 13 <i>String</i> }]	Starts notation: " "
	Copy to
0	Copy to
1	Copy (with attachment) to
2	Copy (without attachment) to
3	Attachment
4	Attachments
5	Enclosure
6	Enclosures
7	Under Separate Cover
8	Letter to
9	Memorandum to
10	Copy (with attachments) to
11	Copy (without attachments) to
12	Abstract Only to
13	Complete Memorandum to
	<i>String</i> Copy (<i>String</i>) to.
.NE	Ends notation.
.AV <i>Name</i> [1]	Prints approval signature.
.CS [<i>Pgs</i>] [<i>Other</i>] [<i>Tot</i>] [<i>Figs</i>] [<i>TbIs</i>] [<i>Ref</i>]	Prints cover sheet.
.TX	Calls user exit for table-of-contents titles.
.TY	Calls user exit for table-of-contents header.
.TC [<i>Slev</i>] [<i>Spacing</i>] [<i>Tlev</i>] [<i>Tab</i>] [<i>H1</i>] [<i>H2</i>] [<i>H3</i>] [<i>H4</i>] [<i>H5</i>]	Prints table of contents.

Paragraphs

Item	Description
.P [{0 1 2}]	Starts paragraph:
0	Left-justified (default)
1	Indented
2	Indented except after .H , .LE , .DE .

Section Headings

Item	Description
.H { 1 2 3 4 5 6 7 } [<i>HeadingText</i>] [<i>FootnoteMark</i>]	Specifies numbered headings.
.HU <i>HeadingText</i>	Specifies unnumbered headings.
.HM { 1 0001 A a I i }...	Specifies heading mark style:
1	Arabic
0001	Arabic with leading 0s (zeros)
A	Uppercase alphabetic
a	Lowercase alphabetic
I	Uppercase Roman
i	Lowercase Roman.
.HX [<i>Dlev</i>] [<i>Rlev</i>] [<i>HeadingText</i>]	Calls user-defined exit macro before headings.
.HY [<i>Dlev</i>] [<i>Rlev</i>] [<i>HeadingText</i>]	Calls user-defined exit macro in the middle of headings.
.HZ [<i>Dlev</i>] [<i>Rlev</i>] [<i>HeadingText</i>]	Calls user-defined exit macro after headings.

Lists

If the last option [**1**] is present in the list-start macros, there is no space between items.

Item	Description
.AL [{ 1 A a I i }] [<i>TextIndent</i>] [1]	Starts automatically incremented list (1).
.BL [<i>TextIndent</i>] [1]	Starts a bullet list.
.DL [<i>TextIndent</i>] [1]	Starts a dash list.
.ML <i>Mark</i> [<i>TextIndent</i>] [1]	Starts a list in which each list item is tagged with a specified mark. If the value of the <i>TextIndent</i> is NULL or omitted, it is set to [<i>Mark</i> - width + 1]. If the 3rd argument is specified, no blank lines separate items in the list.
.RL [<i>TextIndent</i>] [1]	Starts a reference list.
.VL <i>TextIndent</i> [<i>MarkIndent</i>] [1]	Starts a variable tag list.

Item**Description****.LI** [*Mark*] [**1**]

Starts list item; **1** means that the *Mark* variable value is to be prefixed to the current mark.

.LE [**1**]

Ends list item; **1** means to output a blank line after list. The default is no blank line.

.LB *TextIndent MarkIndent Pad Type* [*Mark*] [{**0 1**}]
[{**0 1**}]

Begins list:

The value of the *Type* variable is:

1= . **2**=) **3**=() **4**=[] **5**=<> **6**={}.

Sixth option:

0

No blank line before each list item.

Seventh option:

0

No blank line before list.

Clears list status up to the *Level* variable value.

.LC [*Level*]**Displays, Tables, Equations, and Footnotes****.DS** [{**0 1 2 3**}] [{**0 1**}] [*Number*]**.DS** [{**L I C CB**}] [{**N F**}] [*Number*]

Starts static display:

0 or L

No indent

1 or I

Indent from left

2 or C

Center each line

3 or CB

Center as a block

0 or N

No-fill

1 or F

Fill.

Number

Indent from right the number of spaces specified by the *Number* parameter.

.DF [{**0 1 2 3**}] [{**0 1**}] [*Number*]**.DF** [{**L I C CB**}] [{**N F**}] [*Number*]

Starts floating display:

0 or L

No indent

1 or I

Indent from left

2 or C

Center each line

3 or CB

Center as a block

0 or N

No-fill

1 or F

Fill.

Number

Indent from right the number of spaces specified by the *Number* parameter.

.DE

Ends display.

.FG [Title] [Override] [0 1 2]

The value of the *Override* variable replaces or enhances the default numbering. Specifies figure caption:

0

Override value is used as a prefix.

1

Override value becomes a suffix.

2

Replace *Override* value becomes a replacement.

.TS [H]

Starts table:

H

Multipage table.

.TH [N]

Must be used when specifying option **H** to **.TS**:

N

Suppresses table headers unless on top of new page.

.TE

Ends table.

.TB [Title] [Override] [0 1 2]

The value of the *Override* variable replaces or enhances the default numbering. Specifies table caption:

0

Override value is used as a prefix.

1

Override value becomes a suffix.

2

Replace *Override* value becomes a replacement.

.EX [Title] [Override] [0 1 2]

The value of the *Override* variable replaces or enhances the default numbering. Specifies exhibit caption:

0

Override value is used as a prefix.

1

Override value becomes a suffix.

2

Replace *Override* value becomes a replacement.

.EQ [Label]

Starts equation display using the specified label.

.EN

Ends equation display.

.EC [Title] [Override] [0 1 2]

The value of the *Override* variable replaces or enhances the default numbering. Specifies equation caption:

0

Override value is used as a prefix.

1

Override value becomes a suffix.

2

Replace *Override* value becomes a replacement.

.FS [Label]

Starts footnote using the specified label as an indicator. Default is numbered footnote.

.FE

Ends footnote.

.FD [{0 1 2 3 4 ... 11}] [1]

Sets footnote format:

First option:

Set up formatting style for footnote text. Default is 0 for **mmt** command. Default is 10 for **mm** command. See the following table for the value.

Second option:

Reset footnote counter on first-level heading.

.FD Arg.	Format
0	Hyphens .nh Adjusted .ad Text Indented Yes Label Justified Left
1	Hyphens .hy Adjusted .ad Text Indented Yes Label Justified Left

.FD Arg.	Format
2	Hyphens .nh Adjusted .na Text Indented Yes Label Justified Left
3	Hyphens .hy Adjusted .na Text Indented Yes Label Justified Left
4	Hyphens .nh Adjusted .ad Text Indented No Label Justified Left
5	Hyphens .hy Adjusted .ad Text Indented No Label Justified Left
6	Hyphens .nh Adjusted .na Text Indented No Label Justified Left

.FD Arg.	Format
7	Hyphens . hy Adjusted . na Text Indented No Label Justified Left
8	Hyphens . nh Adjusted . ad Text Indented Yes Label Justified Right
9	Hyphens . hy Adjusted . ad Text Indented Yes Label Justified Right
10	Hyphens . nh Adjusted . na Text Indented Yes Label Justified Right
11	Hyphens . hy Adjusted . na Text Indented Yes Label Justified Right

Page Headers and Footers

Item

.PH "'Left'Center'Right'"

Description

Specifies page header.

Item	Description
.OH <i>""Left'Center'Right""</i>	Specifies odd-page header.
.EH <i>""Left'Center'Right""</i>	Specifies even-page header.
.PF <i>""Left'Center'Right""</i>	Specifies page footer.
.OF <i>""Left'Center'Right""</i>	Specifies odd-page footer.
.EF <i>""Left'Center'Right""</i>	Specifies even-page footer.
.BS	Starts bottom-block.
.BE	Ends bottom-block.
.PX	Calls user exit for page-header.
.TP	Calls top of page macro.

Miscellaneous Macros

Item	Description
.B <i>[Option] [Prev-Font-option]</i>	Prints in bold (up to six options).
.I <i>[Option] [Prev-Font-option]</i>	Prints in italics (up to six options); underlines with the nroff command.
.R	Returns to Roman font.
.PM <i>[Option]</i>	Sets proprietary marking. If you do not give the .PM macro an option, you turn off proprietary markings. The /usr/lib/macros/string.mm file contains some proprietary markings. This file should be edited to meet the user's needs.
.RD <i>[Prompt] [Diversion] [String]</i>	Stops code macro. The <i>Prompt</i> variable should be a user-defined string without spaces. The <i>Diversion</i> variable allows the typed-in text to be saved. The <i>String</i> variable contains the first line typed following the prompt.
.RP <i>[{0 1}] [{0 1 2 3}]</i>	Produces reference page: First option: 0 Resets reference counter (default). 1 Does not reset reference counter. Second option: 0 Causes an .SK macro after (default). 1 Does not cause an .SK macro after. 2 Does not cause an .SK macro before. 3 Does not cause an .SK macro before or after.
.RS/.RF	Numbers references automatically.

Item	Description
.WC [{ N WF -WF FF -FF WD -WD FB -FB }]	Controls width for footnotes and displays when using two columns: N Normal mode (-WF , -FF , -WD). WF Footnotes always wide. -WF Footnotes follow page style. FF First footnote determines width of remaining footnotes on that page. -FF Footnotes follow setting of WF or -WF option. WD Always wide displays. -WD Displays follow page style. FB Floating display causes page break (default). -FB Floating display does not cause page break.
.SP [<i>Lines</i>]	Skips lines down.
.SK [<i>Number</i>]	Skips the specified number of pages. (The default is 1.)
.OP	Breaks to an odd page.
.2C	Prints output in two columns.
.1C	Prints output in one column (normal line width restored).
.SA [<i>Option</i>]	Sets right-margin justification Options: 0 Sets default to off (default for the nroff command). 1 Sets default to on (default for the troff command). If no option is specified, macro reverts to current default.
.SM <i>String1</i> [<i>String2</i>] [<i>String3</i>]	Reduces size of the <i>String1</i> variable value by 1 point if the <i>String3</i> variable value is omitted; otherwise, reduces size of the <i>String2</i> variable value by one point.
.HC <i>Character</i>	Sets hyphenation character to the <i>Character</i> variable value.

Item	Description
.S [<i>PointSize</i>] [<i>VerticalSpacing</i>]	Sets point size and vertical spacing (the troff command only). Defaults: Point size = 10p Vertical spacing = 12p Options 1 and 2: Number New value. +/-Number Increment to current value. D Default. C Current value. P Previous value.
.VM [<i>Top</i>] [<i>Bottom</i>]	Sets variable vertical margins.
.nP	Starts double-line indent on paragraph.

The following macros are for alternating fonts and all take one to six options:

Item	Description
.IB	Alternates italics (underlines for nroff) and bold.
.BI	Alternates bold and italics.
.RI	Alternates Roman and italics.
.IR	Alternates italics (underlines for nroff) and Roman.
.RB	Alternates Roman and bold.
.BR	Alternates bold and Roman.

mm Registers

If an * (asterisk) follows a register name, that register can be set one of two ways: from the command line (see the example in the **mm** command) or before the formatter reads **mm** macro definitions. In the following list, the number shown in parentheses is the default value.

Item	Description
A *	Handle preprinted forms.
Au	Inhibit author information on first page (1).
C *	Copy type (such as Original and Draft) (0).
Cl	Contents level (2).
Cp	Placement of figures, tables, equations, and exhibits (1).
D *	Debug flag (0). If set to 1, the mm command continues even if it encounters an error that is usually fatal.
De	Eject page after floating displays (0).
Df	If set to 1, format register for floating displays (5).

Item	Description
Ds	Static display pre- and post-space (1).
E *	Control font of the Subject/Date/From fields (0): 0 = bold; 1 = Roman.
	0 Bold (0)
	1 Roman.
Ec	Equation counter.
Ej	Page-ejection flag for headings (0).
Eq	Equation label placement (0).
Ex	Exhibit counter.
Fg	Figure counter.
Fs	Vertical footnote separation (1).
H1...H7	Heading counters.
Hb	Heading break level (after .H and .HU) (2).
Hc	Heading centering level for .H and .HU (0).
Hi	Heading temporary indent (after .H and .HU) (1).
Hs	Heading space level (after .H and .HU) (2).
Ht	Heading type:
	0 Concatenated numbers (0)
	1 Single numbers (0).
Hu	Heading level for unnumbered heading (2).
Hy	Hyphenation control:
	0 No hyphenation (0)
	1 Enable hyphenation.
L *	Length of page (66v).
Le	List of equations following table of contents (0):
	0 Do not print
	1 Print.
Lf	List of figures following table of contents (0):
	0 Do not print
	1 Print.
Li	List indent (5, troff command); (6, nroff command).
Ls	List level down to which there is spacing between items (6).

Item	Description
Lt	List of tables following table of contents (0): 0 Do not print 1 Print
Lx	List of exhibits following table of contents (1): 0 Do not print 1 Print.

Item	Description
N *	Numbering style (0).
Np	Numbered paragraphs: 0 Unnumbered 1 Numbered (0).
O *	Offset of page.
Oc	Page numbering style for table of contents: 0 Lowercase Roman 1 Arabic (0).
Of	Figure caption style (0).
P	Page number; managed by the mm command (0). The register accepts a value of 0, or positive integers.
Pi	Paragraph indent (5).
Ps	Paragraph spacing (1).
Pt	Paragraph type (0).
Pv	PRIVATE header: 0 Do not print PRIVATE 1 On first page only 2 On all pages (0).
Rf	Reference counter; used by .RS macro.
S *	The troff command's default point size (10).
Si	Display indent (5).
T *	Type of the nroff command output device (0).
Tb	Table counter.
U *	Underlining style (the nroff command) for .H and .HU (0).

Item	Description
W *	Width of page (line and title length).

mm Strings

Print special strings by using the following escape sequences:

Item	Description
*x	For strings with single-character names (x)
*(xx)	For strings with two-character names (xx).

String Names

Item	Description
BU	Bullet.
Ci	Indent of heading levels in the table of contents.
DT	Current date. The locale-specific date format specified by the locale setting for the LC_TIME category is used as the default setting. This corresponds to the %x format specifier of the strftime subroutine. Use the .ND macro to change the current date.
EM	Em dash.
F	Footnote numbering.
HF	Heading level font string:
1	Roman
2	italics
3	Bold (2 2 2 2 2 2).
HP	Point sizes of the various heading levels.
Le	Title of the list of equations.
Lf	Title of the list of figures.
Lt	Title of the list of tables.
Lx	Title of the list of exhibits.
RE	SCCS SID of mm macros.
Rf	Reference numberer.
Rp	Title of the reference page.
Tm	Trademark.
`	Grave accent.
'	Acute accent.
^	Circumflex.
~	Tilde.
:	Lowercase umlaut.
;	Uppercase umlaut.
,	Cedilla.

Reserved Names

If you define your own strings, macros, and registers, use only names that consist of either a single lowercase letter, or a lowercase letter followed by any character other than a lowercase letter. The names **c2** and **nP** are exceptions to this; they are reserved.

mptx Macro Package for the nroff and troff Commands

The **mptx** macro package provides a definition for the **.xx** macro that is used for formatting a permuted index produced by the **ptx** command. The **mptx** macro package does not provide any other formatting capabilities, such as headers and footers. Use the **mptx** macro package in conjunction with the **mm** macro package if such capabilities are required. In this case, call the **-mptx** option after the **-mm** call, as follows:

nroff -mm -mptx *File...* | *Printer*

ms Macro Package for the nroff and troff Commands

The **ms** macro package of **nroff** and **troff** command macro definitions provides a formatting facility for various styles of articles, theses, and books. In certain cases, the **col** command may be required to postprocess output.

The macro requests are defined in the **ms** Requests section. Many **nroff** and **troff** command requests can have unpredictable results in conjunction with this package. However, the first 4 requests in the following list can be used after initialization, and the last 2 requests can be used before initialization.

Item	Description
.bp	Begins new page.
.br	Breaks output line.
.ce [<i>Number</i>]	Centers the next specified number of lines.
.ls [<i>Number</i>]	Sets line spacing. Set the value of the <i>Number</i> variable to 1 (one) to single-space text; and to 2 to double-space text.
.na	Turns off alignment of right margin.
.sp [<i>Number</i>]	Inserts the specified number of spacing lines.

Font and point-size changes with the **\f** and **\s** macros are also allowed. For example, **\fIword\fR** italicizes word. Output of the **tbl**, **eqn**, and **refer** command preprocessors for equations, tables, and references is acceptable as input.

Formatting distances can be controlled in **ms** macros by means of built-in number registers. For example, the following number register sets the line length to 6.5 inches:

```
.nr LL 6.5i
```

For more information on **ms** macro registers, see **ms Registers**.

ms Requests

Following are external **ms** macro requests:

Item	Description
.AB [X]	Begins abstract. If X is no, do not label abstract. Initial Value: - Break: yes

Item	Description
.AE	Ends abstract. Break: yes Initial Value: - Break: yes
.AIDName	Author's institution. Initial Value: - Break: yes
.AM	Sets accent mark definitions. Initial Value: - Break: no
.AUName	Sets author's name. Initial Value: - Break: yes
.B [X]	Puts X in boldface. If no X , switches to boldface. Initial Value: - Break: no
.B1	Begins text to be enclosed in a box. Initial Value: - Break: yes
.B2	Ends boxed text and prints it. Initial Value: - Break: yes
.BT	Prints bottom title at foot of page. Initial Value: date Break: no
.BX X	Prints word X in a box. Initial Value: - Break: no
.CM	Cuts mark between pages. Initial Value: if t Break: no
.CT	Indicates chapter title; page number moved to CF (TM). Initial Value: - Break: yes Reset: yes

Item	Description
.DA [X]	Forces date X at bottom of page. If no X , date is today. Initial Value: if n Break: no
.DE	Ends display (unfilled text) of any kind. Initial Value: - Break: yes
.DS X Y	Begins display with keep. X =I, L, C, B; Y =indent. Initial Value: I Break: yes
.ID Y	Indents display with no keep; Y =indent. Initial Value: 8n, .5i Break: yes
.LD	Sets left display with no keep. Initial Value: - Break: yes
.CD	Centers display with no keep. Initial Value: - Break: yes
.BD	Block display; centers entire block. Initial Value: - Break: yes
.EF X	Sets even page footer X (3 part as for troff command, .tl request). Initial Value: - Break: no
.EH X	Sets even page header X (3 part as for troff command, .tl request). Initial Value: - Break: no
.EN	Ends displayed equation produced by eqn command. Initial Value: - Break: yes
.EQ [X] [Y]	Breaks out equation. X =L, I, C; Y is equation number. Initial Value: - Break: yes

Item	Description
.FE	Ends footnote to be placed at bottom of page. Initial Value: - Break: no
.FP	Numbers footnote paragraph; can be redefined. Initial Value: - Break: no
FS [X]	Starts footnote; X is optional footnote label. Initial Value: - Break: no
.HD	Sets optional page header below header margin. Initial Value: undef Break: no
.I [X]	Italicizes X . If no X , equivalent to italics font .ft 2 . Initial Value: - Break: no
.IP X Y	Indents paragraph, with hanging tag X . Y specifies spaces to indent. Initial Value: - Break: yes Reset: yes
.IX X Y	Indexes words such as X and Y , up to five levels. Initial Value: - Break: yes
.KE	Ends keep of any kind. Initial Value: - Break: no
.KF	Begins floating keep; text fills remainder. Initial Value: - Break: no
.KS	Begins keep; keeps unit together on a single page. Initial Value: - Break: yes
Item	Description
.LG	Sets larger type size; increases point size by 2. Valid only for the troff command. Initial Value: - Break: no

Item	Description
.LP	Begins left block paragraph. Initial Value: - Break: yes Reset: yes
.MC X	Sets multiple columns. X is column width. Initial Value: - Break: yes Reset: yes
.ND [X]	Indicates no date in page footer; X is date on cover. Initial Value: if t Break: no
.NH X Y	Sets numbered header: X =level; X =0, resets; X =S, sets to Y . Initial Value: - Break: yes Reset: yes
.NL	Sets point size back to default. Valid for the troff command only. Initial Value: 10p Break: no
.OF X	Sets odd page footer X (3 part as for me macro, .tl request). Initial Value: - Break: no
.OH X	Sets odd page header X (3 part as for me macro, .tl request). Initial Value: - Break: no
.P1	Prints header on first page. Initial Value: if TM Break: no
.PP	Indents first line of paragraph. Initial Value: - Break: yes Reset: yes
.PT	Prints page title at head of page. Initial Value: % Break: no

Item	Description
.PX X	Prints index (table of contents); X =do not suppress title. Initial Value: - Break: yes
.QP	Quotes paragraph (indented and shorter). Initial Value: - Break: yes Reset: yes
.R [X]	Returns to Roman font. Prints in Roman font. If X is missing, equivalent to font .ft1 . Initial Value: on Break: no
.RE	Retreats (end level of relative indentation). Used with the .RS request. Initial Value: 5n Break: yes Reset: yes
.RP [X]	Prints title page in released paper format; X =no, stops title on first page. Initial Value: - Break: no
.RS	Right-shifts in one indentation level (start level of relative indentation). Used with the .IP request. Initial Value: 5n Break: yes Reset: yes
.SG	Sets signature line.
.SH	Sets unnumbered section header (in boldface). Initial Value: - Break: yes Reset: yes
.SM	Sets smaller type size; decrease point size by 2. Valid for the troff command only. Initial Value: - Break: no
.TA	Sets tabs to 8n, 16n, ... (nroff); 5n, 10n, ... (troff). Initial Value: 8n, 5n Break: no
.TC X	Prints table of contents at end; X =do not suppress title. Initial Value: - Break: yes

Item	Description
.TE	Ends table processed by tbl command. Initial Value: - Break: yes
.TH	Ends multipage header of table. Must be used with the .TS H request. Initial Value: - Break: yes
.TL	Sets title line (in boldface and 2 points larger). Initial Value: - Break: yes
.TM	Sets UC Berkeley thesis mode. Initial Value: off Break: no
.TS X	Begins table. If X is H, table prints header on all pages. Initial Value: - Break: yes Reset: yes
.UL X	Underlines X , even for the troff command. Initial Value: - Break: no
.UX X	Sets UNIX; trademark message first time; X appended. Initial Value: - Break: no
.XA X Y	Sets another index entry; X =page; X =no, for none. Initial Value: - Break: yes
.XE	Ends index entry or series of .IX request entries. Initial Value: - Break: yes
.XP	Exdents first line of paragraph; others indented. Initial Value: - Break: yes Reset: yes
.XS X Y	Begins index entry; X =page; X =no, for none; Y =indent. Initial Value: - Break: yes

Item	Description
.1C	Begins one-column format, on a new page. Initial Value: on Break: yes Reset: yes
.2C	Begins two-column format. Initial Value: - Break: yes Reset: yes
.]-	Sets beginning of refer command reference. Initial Value: - Break: no
.]0	Sets end of unclassifiable type of reference. Initial Value: - Break: no
.]N	For journal article, N =1 (one). For book, N =2. For book article, N =3. Initial Value: - Break: no

ms Registers

Following is a list of number registers and their default values:

Item	Description
PS	Sets point size. Takes effect for paragraph. Default is 10.
VS	Sets vertical spacing. Takes effect for paragraph. Default is 12.
LL	Sets line length. Takes effect for paragraph. Default is 6i.
LT	Sets title length. Takes effect on next page. Defaults to the LL register value.
FL	Sets footnote length. Takes effect at next .FS request. Default is 5.5i.
PD	Sets paragraph distance. Takes effect for paragraph. Default is 1v (in nroff), .3v (in troff).
DD	Sets display distance. Takes effect for displays. Default is 1v (in nroff), .5v (in troff).
PI	Sets paragraph indent. Takes effect for paragraph. Default is 5n.
QI	Sets quotation indent. Takes effect at next .QP request. Default is 5n.
FI	Sets footnote indent. Takes effect at next .FS request. Default is 2n.
PO	Sets page offset. Takes effect on next page. Default is 0 (zero) (in nroff), 1i (in troff).
HM	Sets header margin. Takes effect on next page. Default is 1i.
FM	Sets footer margin. Takes effect on next page. Default is 1i.
FF	Sets footnote format. Takes effect at next .FS request. Default is 0 (zero) (1, 2, 3 available).

When resetting number register values, make sure to specify the appropriate units. Set the line length to 7i instead of just 7, which would result in output with one character per line. Setting the **FF** register to

1 (one) suppresses footnote superscripting. Setting it to 2 also suppresses indentation of the first line. Setting the **FF** register to 3 produces a footnote paragraph like the **.IP** request.

Following is a list of string registers available in the **ms** macros. These string registers can be used anywhere in the text.

Item	Description
*Q	Open quotation marks (” in nroff ; ` ` in troff)
*U	Close quotation marks (“ in nroff ; ' ' in troff)
*-	Dash (— in nroff ; - in troff)
*(MO	Month of year
*(DY	Day (current date)
**	Automatically numbered footnote
*' 	Acute accent (before letter)
*` 	Grave accent (before letter)
*^ 	Circumflex accent (before letter)
*, 	Cedilla (before letter)
*: 	Umlaut (before letter)
*~ 	Tilde (before letter).

When using the extended accent mark definitions available with the **.AM** request, these strings should come after, rather than before, the letter to be accented.

Note:

1. It is important to note that floating keeps and regular keeps are diverted to the same space, so they cannot be mixed.
2. The date format is restricted to U.S. English format.

mv Macro Package for the mvt and troff Commands

This package simplifies the typesetting of view graphs and projection slides in a variety of sizes. Although a few macros accomplish most of the formatting tasks needed in making transparencies, the entire facilities of the **troff**, **tbl**, **pic**, and **grap** commands are available for more difficult tasks.

The output can be previewed on most terminals, in particular the Tektronix 4014. For this device, specify the **-rX1** flag (which is automatically specified by the **mvt** command when that command is called with the **-D4014** flag). To preview output on other terminals, specify the **-a** flag.

The **mv** macros are summarized under the following headings:

- Foil-Start Macros
- Level Macros
- Text-Control Macros
- Default-Setting Macros.

Foil-Start Macros

For the following nine macros, the first character of the name (**V** or **S**) distinguishes between view graphs and slides, respectively, while the second character indicates whether the foil is square (**S**), small wide (**w**), small high (**h**), big wide (**W**), or big high (**H**). Slides are narrower than the corresponding view graphs. The ratio of the longer dimension to the shorter one is larger for slides than for view graphs. As a result, slide foils can be used for view graphs, but view graphs cannot be used for slide foils. On the other hand, view graphs can accommodate a bit more text.

Item	Description
.VS [<i>FoilNumber</i>] [<i>FoilID</i>] [<i>Date</i>]	Starts a square view graph. Foil size is to be 7 inches by 7 inches. The foil-start macro resets all variables (such as indent and point size) to initial default values, except for the values of the <i>FoilID</i> and <i>Date</i> variables inherited from a previous foil-start macro. The .VS macro also calls the .A macro.
.Vw, .Vh, .VW, .VH, .Sw, .Sh, .SW, .SH	Same as the .VS macro, except that these macros start view graphs (V) or slides (S) that are small wide (w), small high (h), large wide (W), or large high (H). The following macros are recommended: • .VS for square view graphs and slides • .Sw (and, if necessary, .Sh) for 35mm slides.
.Vw [<i>FoilNumber</i>] [<i>FoilID</i>] [<i>Date</i>]	Same as the .VS macro, except that foil size is 7 inches wide by 5 inches high.
.Vh [<i>FoilNumber</i>] [<i>FoilID</i>] [<i>Date</i>]	Same as the .VS macro, except that foil size is 5 inches wide by 7 inches high.
.VW [<i>FoilNumber</i>] [<i>FoilID</i>] [<i>Date</i>]	Same as the .VS macro, except that foil size is 7 inches wide by 5.4 inches high.
.VH [<i>FoilNumber</i>] [<i>FoilID</i>] [<i>Date</i>]	Same as the .VS macro, except that foil size is 7 inches wide by 9 inches high.
.Sw [<i>FoilNumber</i>] [<i>FoilID</i>] [<i>Date</i>]	Same as the .VS macro, except that foil size is 7 inches wide by 5 inches high.
.Sh [<i>FoilNumber</i>] [<i>FoilID</i>] [<i>Date</i>]	Same as the .VS macro, except that foil size is 5 inches wide by 7 inches high.
.SW [<i>FoilNumber</i>] [<i>FoilID</i>] [<i>Date</i>]	Same as the .VS macro, except that foil size is 7 inches wide by 5.4 inches high.
.SH [<i>FoilNumber</i>] [<i>FoilID</i>] [<i>Date</i>]	Same as the .VS macro, except that foil size is 7 inches wide by 9 inches high.

Note: The **.VW** and **.SW** foils are meant to be 9 inches wide by 7 inches high. However, because the typesetter paper is generally only 8 inches wide, **.VW** and **.SW** foils are printed 7 inches wide by 5.4 inches high and have to be enlarged by a factor of 9/7 before use as view graphs.

Level Macros

Item	Description
.A [<i>X</i>]	Places text that follows at the first indentation level (left margin). The presence of the <i>X</i> variable suppresses the half-line spacing from the preceding text.
.B [<i>Mark</i>] [<i>Size</i>]	Places text that follows at the second indentation level. Text is preceded by a specified mark (default is a large bullet). The <i>Size</i> variable is the increment or decrement to the point size of the mark with respect to the <i>prevailing</i> point size (default is 0). A value of 100 for the <i>Size</i> variable makes the point size of the mark equal to the default value of the <i>Mark</i> variable.
.C [<i>Mark</i>] [<i>Size</i>]	Same as the .B macro, but for the third indentation level. The default value of the <i>Mark</i> variable is an em dash.
.D [<i>Mark</i>] [<i>Size</i>]	Same as the .B macro, but for the fourth indentation level. The default value of the <i>Mark</i> variable is a small bullet.

Text-Control Macros

Item	Description
.I [+/-] [<i>Indentation</i>] [A[X]]	Changes the current text indent (does not affect titles). The specified indentation is in inches unless dimensioned. The default is 0. If the <i>Indentation</i> variable is signed, it is an increment or decrement. The presence of the <i>A</i> variable calls the .A macro and passes the <i>X</i> variable (if any) to it.
.S [<i>Size</i>] [<i>Length</i>]	Sets the point size and the line length. The value specified in the <i>Size</i> variable is the point size (default is previous). If the <i>Size</i> variable value is 100, the point size reverts to the <i>initial</i> default for the current foil-start macro. If the <i>Size</i> variable is signed, it is an increment or decrement (default is 18 for the .VS , .VH , and .SH macros, and 14 for the other foil-start macros). The <i>Length</i> variable specifies the line length (in inches unless dimensioned; the default is 4.2 inches for the .Vh macro, 3.8 inches for the .Sh macro, 5 inches for the .SH macro, and 6 inches for the other foil-start macros).
.T <i>String</i>	Prints the <i>String</i> variable value as a centered, enlarged title.
.U <i>String1</i> [<i>String2</i>]	Underlines the <i>String1</i> variable value and concatenates the <i>String2</i> variable value (if any) to it. Using this operation is not recommended.

Default-Setting Macros

Item	Description
.DF [<i>Number Name</i>]...	Sets font positions. It cannot be displayed within foil input text; that is, it must follow the input text for a foil, but it must precede the next foil-start macro. The specified number is the position of the font specified by the <i>Name</i> variable. The .DF macro takes up to four pairs of <i>Number Name</i> variables, such as 1 H. The first <i>Name</i> variable specifies the prevailing font. For example: .DF 1 H 2 I 3 B 4 S .
.DV [A] [B] [C] [D]	Alters the vertical spacing between indentation levels. The value specified by the <i>A</i> , <i>B</i> , <i>C</i> , or <i>D</i> variable is the spacing for the .A , .B , .C , or .D macro, respectively. All non-null parameters must be dimensioned. Null parameters leave the corresponding spacing unaffected. The default setting is: .DV .5v .5v 0v .

The **.S**, **.DF**, **.DV**, and **.U** macros do not cause a break. The **.I** macro causes a break only if it is called with more than one variable. All the other macros cause a break.

The **mv** macro package also recognizes the following uppercase synonyms for the following corresponding lowercase **troff** command requests:

- **.AD**
- **.BR**
- **.CE**
- **.FI**
- **.HY**
- **.NA**
- **.NF**
- **.NH**
- **.NX**
- **.SO**
- **.SP**
- **.TA**
- **.TI**

The **Tm** string produces the trademark symbol.

Environment Variable

Item	Description
LANG	Determines the locale's equivalent of y for yes or no queries. The allowed affirmative responses are defined in the locale variable YESSTR . If LANG is not set, or if it is set to an empty string, the YESSTR from the default C locale is used.

nroff and troff Requests for the nroff and troff Commands

The following **nroff** and **troff** requests are included in a specified working file or in standard input. The **nroff** and **troff** requests control the characteristics of the formatted output when the file or standard input is processed with the **nroff** or **troff** commands. The **nroff** and **troff** requests are grouped by function, in the following sections:

- [Numerical Parameter Input](#)
- [Font and Character Size Control](#)
- [Page Control](#)
- [Text Filling, Adjusting, and Centering](#)
- [Vertical Spacing](#)
- [Line Length and Indenting](#)
- [Macros, Strings, Diversions, and Position Traps](#)
- [Number Registers](#)
- [Tabs, Leaders, and Fields](#)
- [Input and Output Conventions and Character Translations](#)
- [Hyphenation](#)
- [Three-Part Titles](#)
- [Output Line Numbering](#)
- [Conditional Acceptance of Input](#)
- [Environment Switching](#)
- [Insertions from Standard Input](#)
- [Input and Output File Switching](#)
- [Miscellaneous](#)

For number variables written as *+Number*, the variable can be expressed as follows:

- The *Number* variable by itself is an absolute value.
- The *+Number* variable increases the currently set value.
- The *-Number* variable decreases the variable relative to its current value.

Note: For all numeric parameters, numbers are expressed using ASCII Arabic numerals only.

The notes at the end of this command are referenced in the specific requests where applicable.

Numerical Parameter Input

Both **nroff** and **troff** requests accept numerical input with the appended scale indicators shown in the following table, where *S* is the current type size in points, *V* is the current vertical line spacing in basic units, and *C* is a nominal character width in basic units.

Indicator	Meaning	Number of Basic nroff Units
i	Inch (machine-dependent for troff)	240

Indicator	Meaning	Number of Basic nroff Units
c	Centimeter	240x50/127
P	Pica = 1/6 inch	240/6
m	Em = S points	C
n	En = Em/2	C (same as Em)
p	Point = 1/72 inch	240/72
u	Basic unit	1
v	Vertical line space	V
<u>k</u>	Width single-width kana	C
<u>K</u>	Width double-width kanji	Two Cs
none	Default	

Note:

1. If a non-kanji output device is selected, an en-width is used instead.
2. If a non-kanji output device is selected, an em-width is used instead.

In the **nroff** request, both the em and the en are taken to be equal to the C, which is output-device dependent; frequent values are 1/10 and 1/12 inch. Actual character widths in the **nroff** request need not be all the same, and characters constructed with predefined strings such as - > are often extra wide.

Japanese Language Support: In the output from the **nroff** command, all double-width Japanese characters such as all kanji and some katakana characters have a fixed width equal to two Cs. All single-width Japanese characters such as some katakana characters have a fixed width equal to C.

The scaling for horizontally-oriented control characters, vertically-oriented control characters, and the requests **.nr**, **.if**, and **.ie** are as follows:

Orientation	Default Measure	Request or Function
Horizontal	Em (m)	.ll , .in , .ta , .lt , .po , .mc , \h , \l
Vertical	Vertical line space (v)	.pl , .wh , .ch , .dt , .sp , .sv , .ne , .rt , \v \x , \L
Register-oriented or Conditional	Basic unit (u)	.nr , .if , .ie
Miscellaneous	Point (p)	.ps , .vs , \H , \s

All other requests ignore scale indicators. When a number register containing an already appropriately scaled number is interpreted to provide numerical input, the unit scale indicator **u** might need to be appended to prevent an additional inappropriate default scaling. The *Number* might be specified in decimal-fraction form, but the parameter that is finally stored is rounded to an integer number of basic units.

Font and Character Size Control

Item	Description
.bd <i>Font Number</i>	Makes the characters in the specified font artificially bold by overstriking them the specified number of times when using nroff, or by printing each character twice separated by <i>Number</i> -1 basic units when using troff. If the <i>Number</i> variable is not specified, the bold mode is turned off. The <i>Font</i> value must be an ASCII font name or font position. For the nroff command, the default setting of the .bd request is 3 3, specifying that characters on the font mounted at position 3 (usually bold) are to be overstruck 3 times (that is, printed in place a total of 4 times). The font name itself can be substituted for the font position; for example, .bd I 3. The <i>Number</i> variable is functionally identical to the -u flag of the nroff command. (The bold mode must be in effect when the characters are physically printed.) This request can affect the contents of the .b general-number register. The bold mode still must be in effect, or restarted at the time of physical output. You cannot turn off the bold mode in the nroff command if it is being controlled locally by the printing device as with, for example, a DASI 300. Initial Value: Off If No Value Specified: -
.bd S <i>Font Number</i>	Makes the characters in the special font bold whenever the specified font is the current font. The mode must be in effect when the characters are physically printed. The <i>Font</i> value must be an ASCII font name or font position. The mode still must be in effect, or again so, at the time of physical output. Initial Value: Off If No Value Specified: -
.cs <i>Font Number M</i>	Sets constant character space (width) mode to the <i>Font</i> variable value (if mounted). The width of every character is taken to be the value specified in the <i>Number</i> variable divided by 36 ems. If the <i>M</i> variable is not specified, the em width is that of the character's point size; if the <i>M</i> variable is given, the width is the value specified by the <i>M</i> variable minus points. All affected characters are centered in this space, including those with an actual width larger than this space. Special font characters occurring while the specified font is the current font are also so treated. The <i>Font</i> value must be an ASCII font name or font position. If the <i>Number</i> variable is absent, the mode is turned off. The mode must be in effect when the characters are physically printed. This request is ignored by the nroff command. Relevant values are part of the current environment. The mode still must be in effect, or again so, at the time of physical output. Initial Value: Off If No Value Specified: -

Item	Description
.fp <i>Font Number</i> [<i>File</i>]	Specifies the font position. This is a statement that the specified font is mounted on the position specified by the <i>Number</i> variable. The <i>Font</i> variable must be a one- or two-character ASCII font name.  Attention: It is an irrecoverable error if the <i>Font</i> variable is not specified. The .fp request accepts a third optional variable, the <i>File</i> variable, which is the actual path name of the file containing the specified font. The <i>File</i> variable value can be any legal file name and can contain extended characters. Japanese Language Support: The <i>File</i> value can be any legal file name. Values are typesetter- or printer-dependent. Initial Value: - If No Value Specified: Ignored
.ft <i>Font</i>	Changes the font style to the specified font, or if <i>Font</i> value is numeric, to the font mounted on that position. Alternatively, embed \fFont command. The font name P is reserved to mean the previous font. The <i>Font</i> variable value must be an ASCII font name or font position. If using a font name consisting of two characters, use the alternative form of .ft, \f. Relevant values are part of the current environment. Values are typesetter or printer-dependent. Initial Value: Roman If No Value Specified: Previous
.ps [+/-][<i>Number</i>]	Sets the point size to that specified by the +/-<i>Number</i> variable. Although any positive size value can be requested, an invalid size results in the nearest valid size being used. Size 0 refers to the previous size. Alternatively, \sNumber or \s+/-Number; if the <i>Number</i> value is two digits, use \s(Number or \s+/(Number. For compatibility with older versions of the troff command, the form is valid for two-digit values of <i>n</i> = 10, 11, 12, 14, 16, 18, 20, 22, 24, 28, and 36. This request is ignored by the nroff command. Relevant values are part of the current environment. Initial Value: 10 point If No Value Specified: Previous
.ss <i>Number</i>	Sets space-character size to the specified number divided by 36 ems. This size is the minimum word spacing in adjusted text. This request is ignored by the nroff command. Relevant values are part of the current environment. Initial Value: 12/36 em If No Value Specified: Ignored

Page Control

Item	Description
.bp [+/-][<i>Number</i>]	Specifies a break page. The current page is ejected and a new page is begun. If the +/-<i>Number</i> variable is specified, its value becomes the new page number. Also refer to the .ns request. This request usually causes a line break similar to the .br request. Calling this request with the control character " " (instead of ".") suppresses that break function. Initial Value: <i>Number</i>=1 If No Value Specified: -
.mk <i>Register</i>	Marks the current vertical place (or a place in the current diversion) in an internal register (associated with the current diversion level) or in the specified register, if given. The <i>Register</i> variable is the ASCII name of a number register. Mode or relevant values are associated with the current diversion level. For more information, refer to the .rt request. Initial Value: None If No Value Specified: Internal
.ne <i>Number D</i>	Indicates a need for the specified vertical space. If the page space needed (<i>Number</i>) is greater than the distance to the next trap (<i>D</i>), a forward vertical space of size <i>D</i> occurs, which springs the trap. If there are no remaining traps on the page, the size specified by the <i>D</i> variable is the distance to the bottom of the page. If the distance to the next trap (<i>D</i>) is less than one vertical line space (v), another line could still be output before the trap is sprung. In a diversion, the size specified by <i>D</i> is the distance to the diversion trap, if any, or is very large. The value of <i>D</i> is also usually contained in the .t <i>Number</i> register. Mode or relevant values are associated with the current diversion level. Initial Value: <i>Number</i>=1V If No Value Specified: -
.pl [+/-][<i>Number</i>]	Sets page length to the +/-<i>Number</i> variable value. The internal limitation is approximately 136 inches in the nroff command, but varies with the device type in the troff command. A good working maximum for the troff command is 75 inches. The current page length is available in the .p register. Initial Value: 11 inches If No Value Specified: 11 inches
.pn [+/-][<i>Number</i>]	Specifies that the next page (when it occurs) has the page number specified by the +/-<i>Number</i> variable. A .pn request must occur either before text is initially printed or before a break occurs to affect the page number of the first page. The current page number is in the % register. Initial Value: <i>Number</i>=1 If No Value Specified: Ignored
.po [+/-][<i>Number</i>]	Specifies a page offset. The current left margin is set to the +/-<i>Number</i> variable value. The initial troff command value provides 1 inch of left margin. For more information, refer to "Line Length and Indenting". The current page offset is available in the .o register. Initial Value: 0 for the nroff command; 1 for the troff command. If No Value Specified: Previous

Item	Description
.rt [+/-][<i>Number</i>]	Returns upward only to a marked vertical place in the current diversion. If the +/-<i>Number</i> variable value (relative to the current place) is given, the place is the value specified by the +/-<i>Number</i> variable from the top of the page or diversion. If the <i>Number</i> variable is not specified, the place is marked by a previous .mk request. Mode or relevant values are associated with the current diversion level. The .sp request can be used in all cases, instead of the .rt request, by spacing to the absolute place stored in an explicit register as, for example, when using the sequence .mk Register sp \nRu. Initial Value: None If No Value Specified: Internal

Text Filling, Adjusting, and Centering

Item	Description
.ad <i>Indicator</i>	Begins line adjustment. If the fill mode is not on, adjustment is deferred until the fill mode is back on. If the <i>Indicator</i> variable is present, the adjustment type is changed as shown in the following list: Indicator Adjustment Type <i>l</i> Adjust left margin only. <i>r</i> Adjust right margin only. <i>c</i> Center. <i>b</i> or <i>n</i> Adjust both margins. blank Unchanged. The adjustment indicator can also be a number obtained from the .j register.

Japanese Language Support:

Indicator	Adjustment Type
<i>k</i>	Turn on kinsoku shori processing (turned off with .ad n, .ad b, or .ad l). Usually, lines of Japanese text are filled to the margins without regard for the characters beginning or ending lines. When kinsoku shori processing is enabled, lines are prevented from ending with an open bracket character or from beginning with a close bracket or punctuation character. If a line ends with an open bracket, the line is left short and the bracket begins the next line. If a line begins with a close bracket or punctuation character, the preceding line is extended and the character ends the preceding line. Requesting Japanese kinsoku shori processing on an output device that does not support kanji characters has no effect. Relevant values are part of the current environment. Initial Value: Adjust, both If No Value Specified: Adjust

Item	Description
.br	Specifies a break. The filling of the line currently being collected is stopped and the line is output without adjustment. Text lines beginning with space characters and empty text lines (blank lines) also cause a break. Initial Value: - If No Value Specified: -
.ce [<i>Number</i>]	Centers the next specified number of input text lines within the current line length, minus indent. If the <i>Number</i> variable equals 0, any residual count is cleared. A break occurs after each of the <i>Number</i> variable input lines. If the input line is too long, it is left adjusted. Relevant values are part of the current environment. This request usually causes a line break similar to the .br request. Calling this request with the control character " ' " (instead of ".") suppresses that break function. Initial Value: Off If No Value Specified: <i>Number</i>=1
.fi	Fills subsequent output lines. The .u register has a value of 1 (one) in fill mode and a value of 0 (zero) in no-fill mode. Relevant values are part of the current environment. This request usually causes a line break similar to the .br request. Calling this request with the control character " ' " (instead of ".") suppresses that break function. Initial Value: Fill If No Value Specified: -
.na	Specifies no-adjust mode. Adjustment is turned off; the right margin is ragged. The adjustment type for the .ad request is not changed. Output-line filling still occurs if the fill mode is on. Relevant values are part of the current environment. Initial Value: None If No Value Specified: -
.nf	Specifies no-fill mode. Subsequent output lines are neither filled nor adjusted. Input-text lines are copied directly to output lines without regard for the current line length. Relevant values are part of the current environment. This request usually causes a line break similar to the .br request. Calling this request with the control character " ' " (instead of ".") suppresses that break function. Initial Value: Fill If No Value Specified: -

Vertical Spacing

Item	Description
Blank text line	Causes a break and outputs a blank line exactly like an .sp 1 request.
.ls <i>Number</i>	Sets line spacing to the value specified by the +/-<i>Number</i> variable. The <i>Number -1</i> Vs (blank lines) variable values are appended to each output-text line. Appended blank lines are omitted if the text or previous appended blank line reached a trap position. Relevant values are part of the current environment. Initial Value: 1 If No Value Specified: Previous

Item	Description
.ns	Turns on no-space mode. When on, the no-space mode inhibits .sp and .bp requests without a next page number. The no-space mode is turned off when a line of output occurs or with the .rs request. This request usually causes a break. Initial Value: Space If No Value Specified: -
.os	Outputs saved vertical space. The no-space mode has no effect. Used to output a block of vertical space requested by the previous .sv request. Initial Value: - If No Value Specified: -
.rs	Restores spacing. The no-space mode is turned off. This request usually causes a break. Initial Value: None If No Value Specified: -
.sp <i>Number</i>	Spaces vertically in either direction. If the <i>Number</i> variable value is negative, the motion is backward (upward) and is limited to the distance to the top of the page. Forward (downward) motion is truncated to the distance to the nearest trap. If the no-space mode is on, no spacing occurs. Refer to the .ns and .rs requests. This request usually causes a line break similar to the .br request. Calling this request with the control character <code>""</code> (instead of <code>."</code>) suppresses that break function. Initial Value: - If No Value Specified: 1V
.sv <i>Number</i>	Saves a contiguous vertical block of the specified size. If the distance to the next trap is greater than the <i>Number</i> variable value, the specified vertical space is output. The no-space mode has no effect. If this distance is less than the specified vertical space, no vertical space is immediately output, but is remembered for later output (refer to the .os request). Subsequent .sv requests overwrite any still-remembered <i>Number</i> variable value. Initial Value: - If No Value Specified: <i>Number</i>=1V
.vs <i>Number</i>	Sets vertical base-line spacing size <i>V</i> to the <i>Number</i> variable. Transient extra vertical space can be specified by <code>\x N</code>. Relevant values are part of the current environment. Initial Value: The <i>Number</i> variable equals 1/16 inch for the nroff command and 12 points for the troff command. If No Value Specified: Previous

Line Length and Indenting

Item	Description
.in [<i>+/-</i>] <i>Number</i>	Sets indent to the <i>+/-Number</i> variable value. The indent is prepended to each output line. Relevant values are part of the current environment. This request usually causes a line break similar to the .br request. Calling this request with the control character <code>"'"</code> (instead of <code>."</code>) suppresses that break function. Initial Value: <i>Number</i>=0 If No Value Specified: Previous

Item	Description
.ll [+/-] <i>Number</i>	Sets line length to the +/-<i>Number</i> variable value. In the troff command, the maximum line length plus page offset is device-dependent. Relevant values are part of the current environment. Initial Value: 6.5 inches If No Value Specified: Previous
.ti [+/-] <i>Number</i>	Specifies a temporary indent. The next output text line is indented a distance of the value specified by the +/-<i>Number</i> variable with respect to the current indent. A negative value for the <i>Number</i> variable can result in spacing backward over the current indent, so that the resulting total indent can be a value of 0 (zero) (equal to current page offset), but cannot be less than the current page offset. The temporary indent applies only for the one output line following the request; the value of the current indent, which is stored in the .i register, is not changed. Relevant values are part of the current environment. This request usually causes a line break similar to the .br request. Calling this request with the control character " ' " (instead of ".") suppresses that break function. Initial Value: - If No Value Specified: Ignored

Macros, Strings, Diversions, and Position Traps

Item	Description
.am <i>Macro1</i> [<i>Macro2</i>]	Appends to <i>Macro 1</i>; appends version of the .de request. Both the <i>Macro1</i> and <i>Macro2</i> variables must be either one or two ASCII characters. <i>Macro2</i> is a termination sequence to end the diversion. Initial Value: - If No Value Specified: .Macro2=.
.as <i>StringName</i> <i>String</i>	Appends the specified string to the value specified by the <i>StringName</i> variable; appended version of the .ds request. The <i>StringName</i> variable value must be one or two ASCII characters. Initial Value: - If No Value Specified: Ignored
.ch <i>Macro</i> [<i>Number</i>]	Changes the trap position for the specified macro to the value specified by the <i>Number</i> variable. In the absence of the <i>Number</i> variable, the trap, if any, is removed. The <i>Macro</i> variable value must be one or two ASCII characters. Initial Value: - If No Value Specified: -
.da [<i>Macro</i>]	Diverts, appending to the specified macro and appends version of the .di request. The <i>Macro</i> variable must be one or two ASCII characters. Mode or relevant values are associated with the current diversion level. Initial Value: - If No Value Specified: End current diversion

Item	Description
.de <i>Macro1</i> [<i>Macro2</i>]	Defines or redefines the value specified by the <i>Macro1</i> variable. The contents of the macro begins on the next input line. Input lines are copied in copy mode until the definition is stopped by a line beginning with <i>.Macro2</i>. In the absence of the <i>Macro2</i> variable, the definition is stopped by a line beginning with <i>..</i>. A macro can contain .de requests, provided the stopping macros differ or the contained definition terminator is concealed. The <i>..</i> can be concealed as <i>\\.</i>, which copies as <i>\\.</i> and is reread as <i>..</i>. The <i>Macro1</i> and <i>Macro2</i> variables must each be one or two ASCII characters. Initial Value: - If No Value Specified: <i>.Macro2=..</i>
.di [<i>Macro</i>]	Diverts output to the specified macro. Normal text processing occurs during diversion except that page offsetting is not performed. The diversion ends when the .di or .da request is encountered without a variable. Extraneous requests of this type should not be displayed when nested diversions are being used. The <i>Macro</i> variable must be one or two ASCII characters. Mode or relevant values are associated with the current diversion level. Initial Value: - If No Value Specified: End
.ds <i>StringName</i> <i>String</i>	Defines a string specified by the <i>StringName</i> variable to contain the value specified by the <i>String</i> variable. Any initial double-quote in <i>String</i> is stripped off to permit initial blanks. The <i>StringName</i> variable must be one or two ASCII characters.

Item

.ds *StringName* ^A
<*SetNumber*>
<*MessageNumber*>
[*A*"<*DefaultMessage*>
"] [*A*<*Argument*>
^B<*Argument*> ^B
<*Argument*>...]

Description

Provides an alternate **.ds** syntax that allows the use of a message catalog for language-independent string definitions.

Based on the message *SetNumber* and the *MessageNumber* within the locale-specific catalog, the message catalog is read in copy mode and the corresponding message is placed into the *StringName* variable. The initial sequence specifying the message set and message number can be omitted for backward compatibility. The ASCII code Control-A (^A) delimits message identification, default message and optional argument list. The ASCII code Control-B (^B) delimits an individual optional argument list.

In the following example,

```
.ds {c ^A2 41^A"ERROR: (%1$s) input line \  
%2$s" ^A\n(.F^B\n(.c
```

2 is the message set number.

41 is the message number.

text within quotes ("*...*") is the default message.

\n(.F is the name of the current input file.

\n(.c is the number of lines read from the input file.

If you assume the **troff** command runs with these conditions:

- The message at set 2 and number 41 matches the default message
- The current input file is `paper.doc`
- The **.ds** directive is on line 124 in the input file.

then the string {c would be defined as:

```
ERROR: (paper.doc)input line 123
```

Other examples are:

```
.ds {c ^A2 41  
/* Without optional default message */  
  
.ds {c ^A2 41^A"ERROR: (%1$s) input file \  
%2$s" /* Without optional arguments */
```

Item**Description**

If both the set number and the message number are set to zero, then the current date is returned in the current local's format. A user defined date format string can be defined in the default message field. The user defined format string must conform to the conversion specifications outlined by the **strftime** function in *Technical Reference: Base Operating System and Extensions*.

In the following examples:

```
.ds DT^A0 0
```

If the current date were July 10, 1991, in an English U.S. locale, DT would be defined as 7/10/91.

```
.ds DT^A0 0^A"Today is %B %d, %Y"
```

If the current date were July 10, 1991, in an English U.S. locale, DT would be defined as Today is July 10, 1991.

The second syntax method is not intended for general use. It is used in the **nroff** and **troff** macro files supplied with the system to facilitate internationalization of internally generated messages.

Initial Value: -

If No Value Specified: Ignored

.dt Number Macro

Installs a diversion trap at the position specified by the *Number* variable in the current diversion to start the specified macro. Another **.dt** request redefines the diversion trap. If no variables are given, the diversion trap is removed. The *Macro* variable must be one or two ASCII characters. Mode or relevant values are associated with the current diversion level.

Initial Value: -

If No Value Specified: Off

.em Macro

Calls the specified macro when all input has ended. The effect is the same as if the contents of the specified macro had been at the end of the last file processed. The specified macro must be one or two ASCII characters.

Initial Value: None

If No Value Specified: None

.it Number Macro

Sets an input-line-count trap to call the specified macro after the number of lines of text input specified by the *Number* variable have been read (control or request lines are not counted). The text can be inline text or text provided by macros called explicitly (through inline calls) or implicitly (through traps). The *Macro* variable must be one or two ASCII characters. Relevant values are part of the current environment.

Initial Value: -

If No Value Specified: Off

.rm Name

Removes the specified request, macro, or string. The *Name* variable value is removed from the name list and any related storage space is freed. Subsequent references have no effect. The *Name* variable must be one or two ASCII characters.

Initial Value: -

If No Value Specified: Ignored

Item	Description
.rn <i>Name1 Name2</i>	Renames the request, macro, or string value specified by the <i>Name1</i> variable to the value specified by the <i>Name2</i> variable. The <i>Name1</i> and <i>Name2</i> variable values must each be one or two ASCII characters. Initial Value: Ignored If No Value Specified: -
.wh <i>Number Macro</i>	Installs a trap to call the specified macro at the page position specified by the <i>Number</i> variable. A negative <i>Number</i> variable value is interpreted with respect to the page bottom. Any macro previously planted at the page position specified by the <i>Number</i> variable is replaced by the <i>Macro</i> variable value. A <i>Number</i> variable value of 0 refers to the top of a page. In the absence of the <i>Macro</i> variable, the first trap found at the page position specified by the <i>Number</i> variable, if any, is removed. The <i>Macro</i> variable must be one or two ASCII characters. Initial Value: - If No Value Specified: -

Number Registers

Item	Description
.af <i>Register Indicator</i>	Assigns the format as specified by the <i>Indicator</i> variable to the specified register. The <i>Register</i> variable must be one or two ASCII characters. The available format <i>Indicator</i> variable values are as follows: Indicator Numbering Sequence 1 0,1,2,3,4,5, . . . 001 000,001,002,003,004,005, . . . i 0,i,ii,iii,iv,v, . . . I 0,I,II,III,IV,V, . . . a 0,a,b,c, . . . ,z,aa,ab, . . . ,zz,aaa, . . . A 0,A,B,C, . . . ,Z,AA,AB, . . . ,ZZ,AAA, . . . An Arabic format indicator having <i>N</i> digits (for example, 000000001) indicates a field width of <i>N</i> digits. The read-only registers and the width function are always Arabic. Japanese Language Support: The following value specifies the character width for formatting Japanese numeric output in kanji: k The number is formatted as a kanji string. If this is requested when a non-kanji codeset is specified, a warning message is printed and the 1 format is used. Initial Value: Arabic If No Value Specified: -

Item	Description
.nr <i>Register +/-Number1 Number2</i>	Assigns the specified register the value specified by the <i>+/-Number</i> variable with respect to the previous value, if any. The increment for auto-incrementing is set to the <i>Number2</i> variable value. The <i>Register</i> variable must be one or two ASCII characters. Initial Value: - If No Value Specified: -
.rr <i>Register</i>	Removes the specified register. If many registers are being created dynamically, it can become necessary to remove registers that are not needed to recapture internal storage space for new registers. The <i>Register</i> variable must be one or two ASCII characters. Initial Value: - If No Value Specified: -

Tabs, Leaders, and Fields

Item	Description
.fc <i>Delimiter Indicator</i>	Sets the field delimiter to the specified delimiter; the padding indicator is set to the space character or to the specified indicator. In the absence of variables, the field mechanism is turned off. The <i>Delimiter</i> variable value and the <i>Indicator</i> variable value must be ASCII characters. Initial Value: Off If No Value Specified: Off
.lc <i>Character</i>	Sets the leader repetition character to the specified character, or removes specifying motion. The <i>Character</i> variable value must be an ASCII character. Relevant values are part of the current environment. Initial Value: . If No Value Specified: None
.ta <i>Stop [Type]...</i>	Sets tab stops. Default tab stops are set at every eight characters for the nroff command and every half inch for the troff command. Multiple <i>StopType</i> pairs can be specified by separating them with spaces; a value preceded by + (plus sign) is treated as an increment to the previous stop value. The specified type determines how the text is adjusted at the tab stops. The <i>Type</i> variable values are as follows: Type Adjustment R Right-adjusting C Centering blank Left-adjusting Relevant values are part of the current environment. Initial Value: 8 ens for the nroff command and 0.5 inch for the troff command If No Value Specified: None

Item	Description
.tc <i>Character</i>	Sets the tab repetition character to the specified character, or removes specifying motion. The <i>Character</i> variable value must be an ASCII character. Relevant values are part of the current environment. Initial Value: None If No Value Specified: None

Input/Output Conventions and Character Translations

Item	Description
.cc <i>Character</i>	Sets the basic control character to the specified character, or resets to ”.”. The <i>Character</i> variable value must be an ASCII character. Relevant values are part of the current environment. Initial Value: . If No Value Specified: .
.cu [<i>Number</i>]	A variant of the .ul request that causes every character to be underlined and causes no line breaks to occur in the affected input lines. That is, each output space following a .cu request is similar to an unpaddable space. The .cu request is identical to the .ul request in the troff command. Relevant values are part of the current environment. Initial Value: Off If No Value Specified: <i>Number</i> =1
.c2 <i>Character</i>	Sets the no-break control character to the specified character or resets to ” ’ “. The <i>Character</i> variable value must be an ASCII character. Relevant values are part of the current environment. Initial Value: ' If No Value Specified: '
.ec <i>Character</i>	Sets the escape character to \ (backslash) or to the value specified by the <i>Character</i> variable, if given. The <i>Character</i> variable value must be an ASCII character. Initial Value: \ If No Value Specified: \
.eo	Turns off the escape mechanism. Initial Value: On If No Value Specified: -
.lg [<i>Number</i>]	Turns on the ligature mode if the <i>Number</i> variable value is absent or nonzero; turns off ligature mode if the <i>Number</i> variable value is 0. If the <i>Number</i> variable value is 2, only the two-character ligatures are automatically called. The ligature mode is inhibited for request, macro, string, register, or file names, and in the copy mode. This request has no effect in the nroff command. Initial Value: On, for the troff command If No Value Specified: On

Item	Description
.tr <i>Character1</i> <i>Character2</i> <i>Character3</i> <i>Character4</i>	Translates, among other things, the character value specified by the <i>Character1</i> variable into the <i>Character2</i> variable value, the character value specified by the <i>Character3</i> variable into the <i>Character4</i> variable value. If an odd number of characters is given, the last one is mapped into the space character. To be consistent, a particular translation must stay in effect from input to output time. All specified characters must be ASCII characters. To reset the .tr request, follow the request with previous variables given in duplicate. For example, the following .tr request: <pre>.tr aAbBc<C,></pre> can be reset by entering: <pre>.tr aabbcc</pre> It must stay in effect until logical output. Initial Value: None If No Value Specified: -
.ul [<i>Number</i>]	Underlines in the nroff command (or italicizes in the troff command) the number of input-text lines specified by the <i>Number</i> variable. Actually switches to underline font, saving the current font for later restoration. Other font changes within the span of a .ul request take effect, but the restoration undoes the last change. Output generated by the .tl request is affected by the font change, but does not decrement the <i>Number</i> variable value. For more information, refer to the section "Three-Part Titles". If the specified number is greater than 1, there is the risk that a trap-called macro can provide text lines within the span; environment switching can prevent this. Relevant values are part of the current environment. Initial Value: Off If No Value Specified: <i>Number</i>=1
.uf <i>Font</i>	Underlines the font set to the value specified by the <i>Font</i> variable. In the nroff command, the <i>Font</i> variable cannot be on position 1 (initially Times Roman). The <i>Font</i> variable value must be an ASCII font name. Initial Value: Italic If No Value Specified: Italic

Hyphenation

Item	Description
.hc <i>Character</i>	Sets the hyphenation indicator character to the value specified by the <i>Character</i> variable or to the default. The indicator is not displayed in the output. The <i>Character</i> variable value must be an ASCII character. Relevant values are part of the current environment. Initial Value: \% If No Value Specified: \%

Item	Description
.hw <i>Word1...</i>	Specifies hyphenation points in words with embedded minus signs. Versions of a word with a terminal s are implied; that is, <i>dig-it</i> implies <i>dig-its</i> . This list is examined initially and after each suffix stripping. The space available is 1024 characters, or about 50 to 100 words. Initial Value: If No Value Specified: Ignored
.hy <i>Number</i>	Turns on automatic hyphenation if the specified number is equal to or greater than 1; turns it off if the specified number is equal to 0 (equal to the .nh request). If the specified number is 2, the last lines (ones that cause a trap) are not hyphenated. If the specified number is 4 or 8, the last or first two characters, respectively, of a word are not split off. These values are additive; for example, a value of 14 calls all three restrictions (number equal to 2, number equal to 4, and number equal to 8). Relevant values are part of the current environment. Initial Value: No hyphenation If No Value Specified: Hyphenate
.nh	Turns off automatic hyphenation. Relevant values are part of the current environment. Initial Value: No hyphenation If No Value Specified: -

Three-Part Titles

Item	Description
.lt [+/-][<i>Number</i>]	Sets the length of title value specified by the +/- <i>Number</i> variable. The line length and the title length are independent. Indents do not apply to titles, although page offsets do. Relevant values are part of the current environment. Initial Value: 6.5 inches If No Value Specified: Previous
.pc <i>Character</i>	Sets the page number character to the specified character or removes it. The page-number register remains %. The <i>Character</i> variable value must be an ASCII character. Initial Value: % If No Value Specified: Off
.tl ' <i>Left</i> ' <i>Center</i> ' <i>Right</i> '	The strings represented by the <i>Left</i> , <i>Center</i> , and <i>Right</i> variables, respectively, are left-adjusted, centered, and right-adjusted in the current title length. Any of the strings can be empty, and overlapping is permitted. If the page-number character (initially %) is found within any of the fields, it is replaced by the current page number having the format assigned to the % register. Any ASCII character that is not displayed in the strings can be used as the string delimiter. Initial Value: - If No Value Specified: -

Output-Line Numbering

Item	Description
.nm [+/-] [<i>Number</i>] [<i>M</i>] [<i>S</i>] [<i>I</i>]]	Turns on line-number mode. If the <i>M</i> variable is specified, only those line numbers that are multiples of the <i>M</i> variable value are to be printed. Every line number is printed if the <i>M</i> variable is absent (default is <i>M</i>=1). When line-number mode is in effect, a three-digit Arabic number plus a digit space are prepended to output text lines. The text lines are thus offset by four digit spaces, but otherwise retain their line length. If the <i>S</i> variable is given, it specifies the number of digit spaces to be displayed between the line number and the text (default is <i>S</i>=1). If the <i>I</i> variable is given, it specifies the number of digit spaces to indent before the line number (default is <i>I</i>=0). Relevant values are part of the current environment. Initial Value: - If No Value Specified: Off
.nn <i>Number</i>	Suspends line numbering. The specified number of lines are not numbered. Relevant values are part of the current environment. Initial Value: - If No Value Specified: <i>Number</i>=1

Conditional Acceptance of Input

The *Condition* variable specifies one of the following one-character names:

Item	Description
o	If the current page number is odd.
e	If the current page number is even.
t	If the formatter is the troff command.
n	If the formatter is the nroff command.
.if <i>Condition Anything</i>	If the value specified by the <i>Condition</i> variable is true, accepts the value specified by the <i>Anything</i> variable as input; in multiline case, uses \{Anything\} .
.if <i>!Condition Anything</i>	If the value specified by the <i>Condition</i> variable is false, accepts the value specified by the <i>Anything</i> variable as input.
.if <i>Number Anything</i>	If the expression states that the <i>Number</i> variable value is greater than 0, accept the value specified by the <i>Anything</i> variable as input.
.if <i>!Number Anything</i>	If the expression states that the <i>Number</i> variable value is less than or equal to 0, accepts the value specified by the <i>Anything</i> variable as input.
.if <i>'String1'String2' Anything</i>	If the <i>String1</i> variable value is identical to the <i>String2</i> variable value, accepts the value specified by the <i>Anything</i> variable as input. Any nonblank ASCII character not in the <i>String1</i> and <i>String2</i> variables can be used as the delimiter.
.if <i>!'String1'String2' Anything</i>	If the <i>String1</i> variable value is not identical to the <i>String2</i> variable value, accepts the value specified by the <i>Anything</i> variable as input. Any nonblank ASCII character not in the <i>String1</i> and <i>String2</i> variables can be used as the delimiter.
.el <i>Anything</i>	Specifies the else portion of an if/else conditional.

Item	Description
.ie <i>Condition Anything</i>	Specifies the if portion of an if/else conditional dependent on the value of the <i>Condition</i> variable. Can be used with any of the preceding forms of the .if request.

Environment Switching

Item	Description
.ev <i>Environment</i>	Switches to the specified environment. The value specified by the <i>Environment</i> variable must be 0, 1, or 2. Switching is done in push-down fashion so that restoring a previous environment must be performed with the .ev request rather than with a specific reference. Initial Value: <i>Environment</i> =0 If No Value Specified: Previous

Insertions from Standard Input

Item	Description
.ex	Exits from the nroff command or troff command. Text processing is stopped exactly as if all input had ended. Initial Value: - If No Value Specified: -
.rd <i>Prompt</i>	Reads insertion from standard input until two newline characters in a row are found. If the standard input is the user's keyboard, the specified prompt (or the ASCII BEL character) is written onto the user's terminal. The .rd request behaves like a macro, and additional variables can be placed after the <i>Prompt</i> variable. Initial Value: - If No Value Specified: <i>Prompt</i> =the ASCII BEL character

Input and Output File Switching

Item	Description
.cf <i>File</i>	Copies the contents of the specified file, uninterrupted, into the troff command output file at this point. Problems occur unless the motions in the file restore the current horizontal and vertical position. Initial Value: - If No Value Specified: -
.lf <i>Number File</i>	Corrects the troff command interpretation of the current line number (as specified by the <i>Number</i> variable) and the current file (as specified by the <i>File</i> variable) for use in error messages. Initial Value: - If No Value Specified: -
.nx <i>File</i>	Uses the specified file as the input file. The current file is considered ended and the input is immediately switched to the specified file. Initial Value: - If No Value Specified: End of file

Item	Description
.pi <i>Program</i>	Pipes output to the specified program. This request must occur before any printing occurs. No variables are transmitted to the specified program. Initial Value: - If No Value Specified: -
.so <i>File</i>	Switches the source file. The top input (file-reading) level is switched to the specified file. When this file ends, input is again taken from the original file. The .so request can be nested. When a .so request is encountered, the processing of the specified file is immediate. Processing of the original file (for example, a macro that is still active) is suspended. A file should be preprocessed, if necessary, before being called by the .so request. The eqn, tbl, pic, and grap commands do not reach through a .so request to process an object file. Initial Value: - If No Value Specified: -

Miscellaneous

Item	Description
.ab <i>Text</i>	Prints the value specified by the <i>Text</i> variable to the diagnostic output (usually the terminal) and ends without further processing. If text is missing, the message <code>User Abort</code> is printed and the output buffer is flushed. This request is used in interactive debugging to force output.

Item

.ab ^A<SetNumber>
<MessageNumber>
[^A"<Default> "
[^A<Argument>
^B<Argument>
^B<Argument>...]

Description

Provides alternate syntax to allow use of a message catalog for language-independent abort messages. Prints the appropriate message specified by the parameter on the diagnostic output (usually the terminal) and ends without further processing. If there are no parameters, the message catalog equivalent to the following:

```
troff: User Abort, line no. file filename
```

is output. The output buffer is flushed. This request is used in interactive debugging to force output.

Based on the message *SetNumber* and the *MessageNumber* variables within the locale-specific catalog, the message catalog is read in copy mode and the corresponding message is written to the user's terminal. The initial sequence specifying the message set and message number can be omitted for backward compatibility. The ASCII code Control-A (^A) delimits message identification, default message, and optional argument list. The ASCII code Control-B (^B) delimits individual optional argument list.

In the following example:

```
.ab ^A2 42^A"Processing has been terminated \  
at line %1$s."^A\n(c.
```

2 is the message set number.

42 is the message number.

Text within quotes " . . " is the default message.

\n(c . is the number of lines read from the input file.

If you assume the **troff** command runs with the following conditions:

- The message at set 2 and number 42 matches the default message.
- The **.ab** directive is on line 124 in the input file.

then the following would be displayed on the user's terminal:

```
Processing has been terminated at line 123.
```

Initial Value: -

If No Value Specified: User cancel

Item	Description
.Dt <i>Parameter</i>	Defines the format for returning the date within the nroff or troff request. By default, without the optional <i>Parameter</i>, the locale-specific date format specified by the current locale setting for the LC_TIME category is used. This corresponds to the "%x" format specifier of strftime. <i>Parameter</i> is a format string identical to the format string used with the strftime function in <i>Technical Reference: Base Operating System and Extensions</i>. Reference this function for a complete list of the format specifiers. For example, <pre>.Dt "%A, %B %d, %Y (%T)"</pre> provides the following output for an English-speaking locale: <pre>Thursday, January 31, 1991 (10:40:00)</pre> The %A format is replaced by the locale-specific weekday name. The %B format is replaced by the locale-specific month name. The %d format is replaced by the day of the month in a two-digit format. The %Y format is replaced by the year with the century as a decimal number. The %T format is replaced by the time in hours (24-hour clock), minutes, and seconds in decimal numbers. This format provides for leap seconds and double leap seconds.
.fl	Flushes output buffer. This request usually causes a line break similar to the .br request. Calling this request with the control character "' " (instead of ".") suppresses that break function. Initial Value: - If No Value Specified: -
.ig <i>Macro</i>	Ignores input lines. The .ig request works exactly like the .de request, except that the input is discarded. For more information, refer to "Macros, Strings, Diversions, and Position Traps". The input is read in copy mode, and any auto-incremented registers are affected. The <i>Macro</i> variable must be one or two ASCII characters. Initial Value: - If No Value Specified: .Macro=.
.mc [<i>Character</i>] [<i>N</i>]	Uses the specified character as the margin character to display the specified distance (<i>N</i>) to the right of the margin after each non-empty text line (except those produced by the .tl request). If the output line is too long (as can happen in no-fill mode), the character is appended to the line. If the <i>N</i> variable is not given, the previous <i>N</i> variable is used. The first <i>N</i> variable is 0.2 inches in the nroff command and 1 em in the troff command. Relevant values are part of the current environment. Initial Value: .2 inches in nroff; 1 em in troff If No Value Specified: Off
.pm [<i>Character</i>]	Prints macros. The names and sizes of all of the defined macros and strings are printed on the user's terminal. If any ASCII alphanumeric character is given as a variable, only the total of the sizes is printed. The size is given in blocks of 128 characters. Initial Value: - If No Value Specified: All

Item**Description****.sy Command** [*Flags*]

The specified command is run but its output is not captured at this point. The standard input for the specified command is closed. Output must be explicitly saved in an output file for later processing. Often the **.sy** directive is followed by a subsequent **.so** directive to include the results of the previous command.

For example:

```
.sy date > /tmp/today
Today is
.so /tmp/today
```

Initial Value: -

If No Value Specified: -

.tm String

The specified string is written to the user's terminal.

.tm ^A<SetNumber>
<MessageNumber>
[^A" <DefaultMessage>
"] [^A<Argument>
^B <Argument>
^B<Argument> ...]

Based on the message set number and the message number within the locale-specific catalog, the message catalog is read in copy mode and the corresponding message is written to the user's terminal. The initial sequence specifying the message set and message number can be omitted for backward compatibility. The ASCII code Control-A ^A delimits message identification, default message, and optional argument list. The ASCII code Control-B ^B delimits individual optional argument list.

In the following example:

```
.tm ^A2 23^A"The typesetter is %1$s.On line
%2$s."^A\*(.T^B\n(c.
```

2 is the message set number.

23 is the message number.

Text within quotes " . . " is the default message.

*(.T is the first argument in troff for value of **-T**.

\n(c. is the number of lines read from the input file.

If you assume the **troff** command runs with the following conditions:

- The message at set 2 and number 23 matches the default message.
- The command line has **troff** using the **-T** option with device PSC.
- The **.tm** directive is on line 539 in the input file.

Then the following would be displayed on the user's terminal:

```
The typesetter is psc. On line 538.
```

The locale-specific message catalog is found in **/usr/lib/nls/msg/\$LANG/macros.cat**.

Initial Value: -

If No Value Specified: Newline

Note:

The following notes apply to the **nroff** and **troff** requests. They are referenced by number in the requests where they apply.

1. The **.L** string register contains the current program locale value of all the categories.
2. The **.m** string register contains the locale value of the **LC_MESSAGES** category.

3. The **.t** string register contains the locale value for the **LC_TIME** category.
4. While the **.L**, **.t**, and **.mstring** registers provide access to some environment values, a more general technique can be used to access any other environment variable. For example, if the **TED** environment variable is exported, the following **troff** commands:

```
.sy echo .ds z $TED >x
.SO x
.sy im x
```

set the **z** string register to contain the value of **\$TED**.

Environment Variables

Item	Description
<u>LC_ALL</u>	Specifies the locale to be used for all the locale categories. It overrides any setting of the other locale environment variables.
<u>LC_MESSAGES</u>	Specifies the locale value for the LC_MESSAGES category. This is used if the LC_ALL environment variable is not set.
<u>LC_TIME</u>	Specifies the locale value for the LC_TIME category. This is used if the LC_ALL environment variable is not set.
LANG	Specifies the locale value to be used for all the locale categories. This is used if none of the above environment variables are set. This is the most often used environment variable to specify the locale.

Files

Item	Description
/usr/share/lib/tmac/tmac.*	Contains the pointers to standard macro files.
/usr/share/lib/macros/*	Denotes standard macro files.
/usr/share/lib/tmac/tmac.an	Contains the pointer to the man macro package.
/usr/share/lib/macros/an	Contains the man macro package.
/usr/share/lib/tmac/tmac.e file	Contains the me macro definition file.
/usr/share/lib/me directory	Contains the macro definition files.
/usr/share/lib/tmac/tmac.m	Contains the pointer to the mm macro package.
/usr/share/lib/macros/mmn	Contains the mm macro package.
/usr/share/lib/macros/mmt	Contains the mm macro package.
/usr/share/lib/tmac/tmac.ptx	Points to the macro package.
/usr/share/lib/macros/ptx	Contains the macro package.
/usr/share/lib/tmac/tmac.x	Contains the macro definition files.
/usr/share/lib/ms	Contains the ms macro definitions.
/usr/share/lib/tmac/tmac.v	Contains macro definitions.
/usr/share/lib/macros/vmca	Contains macro definitions.
/usr/lib/nls/msg/\$LANG/macros.cat	Contains locale-specific message catalog for the mm , me , ms , and mv macro packages.
/usr/lib/font/dev*/*	Contains the font width tables.
/var/tmp/trtmp*	Denotes a temporary file.

trpt Command

Purpose

Performs protocol tracing on TCP sockets.

Syntax

trpt [**-a**] [**-f**] [**-j**] [**-pAddress**]... [**-s**] [**-t**]

Description

The **trpt** command queries the buffer for Transmission Control Protocol (TCP) trace records. This buffer is created when a socket is marked for debugging with the **setsockopt** subroutine. The **trpt** command then prints a description of these trace records.

Note: You can use the **traceson** command to turn on socket level debugging for daemons.

When you specify no options, the **trpt** command prints all the trace records found in the system and groups them according to their TCP/IP connection protocol control block (PCB).

Before you can use the **trpt** command, you must:

1. Isolate the problem and mark for debugging the socket or sockets involved in the connection.
2. Find the address of the protocol control blocks associated with these sockets by using the **netstat -aA** command.
3. Then you can run the **trpt** command, using the **-p** flag to supply the associated protocol control block addresses. You can specify multiple **-pAddress** flags with a single **trpt** command.

The **-f** flag can be used to follow the trace log once it is located. The **-j** flag can be used to check the presence of trace records for the socket in question.

If the system image does not contain the proper symbols to find the trace buffer, the **trpt** command cannot succeed.

Output Fields

The information put out by the **trpt** command varies with the flag you use. Definitions of the fields contained in the various types of output follow:

Item	Description
Protocol Control Block identifier	Identifies the protocol block to be traced, as shown in the following example: 4c500c :
Timestamp	Specifies the time at which the connection is attempted, as shown in the following example: 500

Item	Description
Connection State	Specifies the state of the connection with the protocol control block: CLOSED Connection is closed. LISTEN Listening for a connection. SYN_SENT Active; have sent SYN. Represents waiting for a matching connection request after having sent a connection request. SYN_RCVD Have sent and received SYN. Represents waiting for a confirming connection request acknowledgment after having both received and sent connection requests. ESTABLISHED Connection established. CLOSE_WAIT Have received FIN; waiting to receive CLOSE. LAST_ACK Have received FIN and CLOSE; awaiting FIN ACK. FIN_WAIT_1 Have closed; sent FIN. CLOSING Closed; exchanged FIN; awaiting FIN. FIN_WAIT_2 Have closed; FIN is acknowledged; awaiting FIN. TIME_WAIT In 2MSL (twice the maximum segment length) quiet wait after close.
Action	Specifies the current status of the packet trace connection. The output of the command changes depending on the action. Input Receiving input packets. The syntax of the output is: <pre>input (SourceAddress, Port, DestinationAddress, Port) <Sequence Number of the First Data Octet> @ AcknowledgementNumber</pre> as in the following example: <pre>input (src=129.353173176,23, dst=129.35.17.140, 1795) fb9f5461@fb9e4c68</pre> Output Transmitting packets. The syntax of the output is: <pre>output (SourceAddress, Port, DestinationAddress, Port) <Sequence Number Of The First Data Octet>.. <Sequence Number of the Last Data Octet>@ AcknowledgementNumber)</pre> as in the following example: <pre>output (src=129.35.17.140,1795, dst=129.35.17.176, 23) fb9e4c68@fb9f5462</pre> Window Size Specifies the size of the window sending or receiving packets, as shown in the following example: <pre>(win=1000)</pre>

Item	Description
User	Specifies user request. The following is an example of a user request:
	<pre>SLOWTIMO<KEEP></pre>
	Types of user requests and their definitions follow:
	PRU_ATTACH Attach protocol to up.
	PRU_DETACH Detach protocol from up.
	PRU_BIND Bind socket to address.
	PRU_LISTEN Listen for connection.
	PRU_CONNECT Establish connection to peer.
	PRU_ACCEPT Accept connection from peer.
	PRU_DISCONNECT Disconnect from peer.
	PRU_SHUTDOWN Will not send any more data.
	PRU_RCVD Have taken data; more room now.
	PRU_SEND Send this data.
	PRU_ABORT Abort (fast DISCONNECT, DETACH).
	PRU_CONTROL Control operations on protocol.
	PRU_SENSE Return status into m.
	PRU_RCVOOB Retrieve out of band data.
	PRU_SENDOOB Send out of band data.
	PRU_SOCKADDR Fetch socket's address.
	PRU_PEERADDR Fetch peer's address.
	PRU_CONNECT2 Connect two sockets.
	PRU_FASTTIMO 200 milliseconds timeout.
	PRU_SLOTIMO 500 milliseconds timeout.
	PRU_PROTORCV Receive from below.
	PRU_PROTOSEND Send to below.
Drop	Specifies that data was in preceding segment; data is dropped.

Item	Description
Window and Sequence Variables	Types of window and sequence variables follow:
<i>rcv_nxt</i>	Next sequence number expected on incoming segments.
<i>rcv_wnd</i>	Size of receive window.
<i>snd_una</i>	Oldest unacknowledged sequence number.
<i>snd_nxt</i>	Next sequence number to be sent.
<i>snd_max</i>	Highest sequence number sent.
<i>snd_sl1</i>	Window update segment sequence number.
<i>snd_wl1</i>	Window update segment ack number.
<i>snd_wnd</i>	Send window.

Flags

Item	Description
-a	Prints the values of the source and destination addresses for each packet recorded, in addition to the normal output.
-f	Follows the trace as it occurs, waiting briefly for additional records each time the end of the log is reached.
-j	Lists just the protocol control block addresses for which trace records exist.
-pAddress	Shows only trace records associated with the protocol control block specified in hexadecimal by the <i>Address</i> variable. You must repeat the -p flag with each <i>Address</i> variable specified.
-s	Prints a detailed description of the packet-sequencing information, in addition to the normal output.
-t	Prints the values for all timers at each point in the trace, in addition to the normal output.

Examples

1. To print trace information as well as the source and destination addresses for each packet recorded, enter:

```
$ trpt -a
```

This might display the following output:

```
124b0c:
900 ESTABLISHED:input (src=192.9.201.3,4257, dst=192.9.201.2,102
5)2326e6e5@ad938c02(win=200)<ACK,FIN,PUSH> -> CLOSE_WAIT
900 CLOSE_WAIT:output (src=192.9.201.2,1025, dst=192.9.201.3,425
7)ad938c02@2326e6e6(win=4000)<ACK> -> CLOSE_WAIT
900 LAST_ACK:output (src=192.9.201.2,1025, dst=192.9.201.3,4257)
ad938c02@2326e6e6(win=4000)<ACK,FIN> -> LAST_ACK
900 CLOSE_WAIT:user DISCONNECT -> LAST_ACK
900 LAST_ACK:user DETACH -> LAST_ACK 12500c:
800 ESTABLISHED:output (src=192.9.201.2,1024, dst=192.9.201.3,51
2)ad8eaa13@2326e6e5(win=4000)<ACK> -> ESTABLISHED
800 ESTABLISHED:input (src=192.9.201.3,512, \
dst=192.9.201.2,1024)
[2326e6e5..2326e727]@ad8eaa13(win=1ef)<ACK,PUSH> -> ESTABLISHED
800 ESTABLISHED:user RCVD -> ESTABLISHED
900 ESTABLISHED:output (src=192.9.201.2,1024, dst=192.9.201.3,51
2)ad8eaa13@2326e727(win=4000)<ACK> -> ESTABLISHED
900 ESTABLISHED:input (src=192.9.201.3,512, \
dst=192.9.201.2,1024)
[2326e727..2326e82f]@ad8eaa13(win=1ef)<ACK,PUSH> -> ESTABLISHED
900 ESTABLISHED:user RCVD -> ESTABLISHED
900 ESTABLISHED:output (src=192.9.201.2,1024, dst=192.9.201.3,51
2)ad8eaa13@2326e82f(win=4000)<ACK> -> ESTABLISHED
900 ESTABLISHED:input (src=192.9.201.3,512, \
dst=192.9.201.2,1024)
2326e82f@ad8eaa13(win=1ef)<ACK,FIN,PUSH> -> CLOSE_WAIT
900 CLOSE_WAIT:output (src=192.9.201.2,1024, \
dst=192.9.201.3,512)
```

```
ad8eaa13@2326e830(win=4000)<ACK> -> CLOSE_WAIT
900 LAST_ACK:output (src=192.9.201.2,1024, dst=192.9.201.3,512)a
d8eaa13@2326e830(win=4000)<ACK,FIN> -> LAST_ACK
900 CLOSE_WAIT:user DISCONNECT -> LAST_ACK
900 LAST_ACK:user DETACH -> LAST_ACK
$ _
```

2. To list the protocol control blocks that have trace records, enter:

```
trpt -j
```

This might display the following output:

```
124b0c, 12500c
```

3. To print the trace records associated with a single protocol control block, enter:

```
trpt -p 12500c
```

This might display the following output:

```
800 ESTABLISHED:output ad8eaa13@2326e6e5(win=4000)<ACK> ->
ESTABLISHED
800 ESTABLISHED:input [2326e6e5..2326e727]@ad8eaa13(win=1ef)
<ACK,PUSH> -> ESTABLISHED
800 ESTABLISHED:user RCVD -> ESTABLISHED
900 ESTABLISHED:output ad8eaa13@2326e727(win=4000)<ACK> -> ESTABLISHED
900 ESTABLISHED:input [2326e727..2326e82f]@ad8eaa13(win=1ef) <ACK,PUSH> -> ESTABLISHED
900 ESTABLISHED:user RCVD -> ESTABLISHED
900 ESTABLISHED:output ad8eaa13@2326e82f(win=4000)<ACK> -> ESTABLISHED
900 ESTABLISHED:input 2326e82f@ad8eaa13(win=1ef)<ACK,FIN,PUSH> -> CLOSE_WAIT
900 CLOSE_WAIT:output ad8eaa13@2326e830(win=4000)<ACK> -> CLOSE_WAIT
900 LAST_ACK:output ad8eaa13@2326e830(win=4000)<ACK,FIN> -> LAST_ACK
900 CLOSE_WAIT:user DISCONNECT -> LAST_ACK
900 LAST_ACK:user DETACH -> LAST_ACK
$ _
```

true or false Command

Purpose

Returns an exit value of zero (true) or a nonzero exit value (false).

Syntax

true

false

Description

The **true** command returns a zero exit value. The **false** command returns a nonzero exit value. These commands are most often used as part of a shell script.

Examples

To construct a loop that displays the date and time once each minute, use the following code in a shell script:

```
while true
do
    date
    sleep 60
done
```

truss Command

Purpose

Traces a process's system calls, dynamically loaded user level function calls, received signals, and incurred machine faults.

Syntax

```
truss [ -f ] [ -c ] [ -a ] [ -l ] [ -d ] [ -D ] [ -e ] [ -i ] [ { -t | -x } [!] Syscall [...] ] [ -s [!] Signal [...] ] [ { -m } [!] Fault [...] ] [ { -u-r | -w } [!] FileDescriptor [...] ] [ { } [!] LibraryName [...] :: [!] FunctionName [ ... ] ] [ -o Outfile ] { Command | -p pid [ . . . ] } [ -X ]
```

Description

The **truss** command executes a specified command, or attaches to listed process IDs, and produces a trace of the system calls, received signals, and machine faults a process incurs. Each line of the trace output reports either the *Fault* or *Signal* name, or the *Syscall* name with parameters and return values. The subroutines defined in system libraries are not necessarily the exact system calls made to the kernel. The **truss** command does not report these subroutines, but rather, the underlying system calls they make. When possible, system call parameters are displayed symbolically using definitions from relevant system header files. For path name pointer parameters, **truss** displays the string being pointed to. By default, undefined system calls are displayed with their name, all eight possible arguments and the return value in hexadecimal format.

When the **-o** flag is used with **truss**, or if standard error is redirected to a non-terminal file, **truss** ignores the hangup, interrupt, and signals processes. This facilitates the tracing of interactive programs which catch **interrupt** and **quit** signals from the terminal.

If the trace output remains directed to the terminal, or if existing processes are traced (using the **-p** flag), then **truss** responds to **hangup**, **interrupt**, and **quit** signals by releasing all traced processes and exiting. This enables the user to terminate excessive trace output and to release previously existing processes. Released processes continue to function normally.

For those options which take a list argument, the name **all** can be used as a shorthand to specify all possible members of the list. If the list begins with a **!**, the meaning of the option is negated (for example, exclude rather than trace). Multiple occurrences of the same option may be specified. For the same name in a list, subsequent options (those to the right) override previous ones (those to the left).

Every machine fault, with the exception of a page fault, results in posting a signal to the process which incurred the fault. A report of a received signal immediately follows each report of a machine fault, unless that signal is being blocked by the process.

To avoid collisions with other controlling processes, **truss** does not trace a process which it detects is being controlled by another process with the **/proc** interface.

The trace output for multiple processes is not produced in strict time order. For example, a read on a pipe may be reported before the corresponding write. However, for each process the output is strictly time-ordered. The trace output contains tab characters and standard tab stops are set at every eight positions.

The system may run out of per-user process slots when tracing children. This is because when tracing more than one process, **truss** runs as one controlling process for each process being traced, doubling the number of process slots being used for any given process. The usual system-imposed limit of 25 processes per user should be taken into account prior to running a trace on multiple processes.

The operating system enforces certain security restrictions on the tracing of processes. You must have access privileges to the commands you are tracing. The **set-uid** and **set-gid** processes can only be traced by a privileged user. The **truss** command loses control of any process which performs an execution of a set-id or unreadable object file, unless it is run by a privileged user. These untraced processes continue normally and independently of truss from the point of the execution.

The lightweight processes (LWP) mentioned in truss output are really kernel threads. The option **-l** displays the LWP id (i.e. the thread id) on each line of the trace output.

User library functions in AIX libraries have both static and dynamic loaded function calls. The tracing with option **-u** is done for dynamically loaded function calls only.

User level function call tracing for dynamically loaded function calls is provided with **-u** option. This option will produce an entry/exit trace of the function calls.

Flags

Item	Description
-a	Displays the parameter strings which are passed in each exec system call.
-c	Counts traced system calls, faults, and signals rather than displaying trace results line by line. A summary report is produced after the traced command terminates or when truss is interrupted. If the -f flag is also used, the counts include all traced Syscalls, Faults, and Signals for child processes.
-d	A timestamp will be included with each line of output. Time displayed is in seconds relative to the beginning of the trace. The first line of the trace output will show the base time from which the individual time stamps are measured. By default timestamps are not displayed.
-D	Delta time is displayed on each line of output. The delta time represents the elapsed time for the LWP that incurred the event since the last reported event incurred by that thread. By default delta times are not displayed.
-e	Displays the environment strings which are passed in each exec system call.
-f	Follows all children created by the fork system call and includes their signals, faults, and system calls in the trace output. Normally, only the first-level command or process is traced. When the -f flag is specified, the process id is included with each line of trace output to show which process executed the system call or received the signal.
-i	Keeps interruptible sleeping system calls from being displayed. Certain system calls on terminal devices or pipes, such as open and kread , can sleep for indefinite periods and are interruptible. Normally, truss reports such sleeping system calls if they remain asleep for more than one second. The system call is then reported a second time when it completes. The -i flag causes such system calls to be reported only once, upon completion.
-l	Display the id (thread id) of the responsible LWP process along with truss output. By default LWP id is not displayed in the output.

Item	Description
-m [!] <i>Fault</i>	Traces the machine faults in the process. Machine faults to trace must be separated from each other by a comma. Faults may be specified by name or number (see the sys/procfs.h header file). If the list begins with the "!" symbol, the specified faults are excluded from being traced and are not displayed with the trace output. The default is -mall -m!fltpage .
-o <i>Outfile</i>	Designates the file to be used for the trace output. By default, the output goes to standard error.
-p	Interprets the parameters to truss as a list of process ids for an existing process rather than as a command to be executed. truss takes control of each process and begins tracing it, provided that the user id and group id of the process match those of the user or that the user is a privileged user.
-r [!] <i>FileDescriptor</i>	Displays the full contents of the I/O buffer for each read on any of the specified file descriptors. The output is formatted 32 bytes per line and shows each byte either as an ASCII character (preceded by one blank) or as a two-character C language escape sequence for control characters, such as horizontal tab (\t) and newline (\n). If ASCII interpretation is not possible, the byte is shown in two-character hexadecimal representation. The first 12 bytes of the I/O buffer for each traced read are shown, even in the absence of the -r flag. The default is -r!all .
-s [!] <i>Signal</i>	Permits listing <i>Signals</i> to trace or exclude. Those signals specified in a list (separated by a comma) are traced. The trace output reports the receipt of each specified signal even if the signal is being ignored, but not blocked, by the process. Blocked signals are not received until the process releases them. Signals may be specified by name or number (see sys/signal.h). If the list begins with the "!" symbol, the listed signals are excluded from being displayed with the trace output. The default is -s all .
-t [!] <i>Syscall</i>	Includes or excludes system calls from the trace process. System calls to be traced must be specified in a list and separated by commas. If the list begins with an "!" symbol, the specified system calls are excluded from the trace output. The default is -tall .

Item

Description

-u [!] [*LibraryName* [...]::[!]*FunctionName* [...]]

Traces dynamically loaded user level function calls from user libraries. The *LibraryName* is a comma-separated list of library names. The *FunctionName* is a comma-separated list of function names. In both cases the names can include name-matching metacharacters *****, **?**, **[]** with the same meanings as interpreted by the shell but as applied to the library/function name spaces, and not to files.

A leading **!** on either list specifies an exclusion list of names of libraries or functions not to be traced. Excluding a library excludes all functions in that library. Any function list following a library exclusion list is ignored. Multiple **-u** options may be specified and they are honored left-to-right. By default no library/function calls are traced.

-w [!] *FileDescriptor*

Displays the contents of the I/O buffer for each write on any of the listed file descriptors (see **-r**). The default is **-w!all**.

-x [!] *Syscall*

Displays data from the specified parameters of traced system calls in raw format, usually hexadecimal, rather than symbolically. The default is **-x!all**.

-X

Displays data from the specified parameters of traced system calls in human-readable format. The supported system calls are **bind**, **connect**, **socketpair**, **lseek**, **creat**, **access**, **accept**, **socket**, and **statx**.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To produce a trace of the **find** command on the terminal, type:

```
truss find . -print >find.out
```

2. To trace the **lseek**, **close**, **statx**, and open system calls, type:

```
truss -t lseek,close,statx,open find . -print > find.out
```

3. To display thread id along with regular output for **find** command, enter:

```
truss -l find . -print >find.out
```

4. To display timestamps along with regular output for **find** command, enter:

```
truss -d find . -print >find.out
```

5. To display delta times along with regular output for **find** command, enter:

```
truss -D find . -print >find.out
```

6. To trace the **malloc()** function call and exclude the **strlen()** function call in the **libc.a** library while running the **ls** command, enter:

```
truss -u libc.a::malloc,!strlen ls
```

7. To trace all function calls in the **libc.a** library with names starting with "m", and exclude the **strlen()** function call in the **libc.a** library while running the **ls** command, enter:

```
truss -u libc.a::m*,!strlen ls
```

8. To trace all function calls in the **libc.a** library with names starting with "m" while running the **ls** command, enter:

```
truss -u libc.a::m* ls
```

9. To trace all function calls from the library **libcurses.a** and exclude calls from **libc.a** while running executable **foo**, enter:

```
truss -u libcurses.a,!libc.a::* foo
```

10. To trace the **refresh()** function call from **libcurses.a** and the **malloc()** function call from **libc.a** while running the executable **foo**, enter:

```
truss -u libc.a::malloc -u libcurses.a::refresh foo
```

11. To trace the system calls arguments in human-readable format, enter:

```
truss -X -t lseek,bind,statx,creat find . -print > find.out
```

Files

/proc filesystem

trustchk Command

Purpose

Administration of Trusted Signature Database (TSD) and Trusted Execution function.

Syntax

Add Files to TSD

```
trustchk [ -R module name ] -s <private key file> -v <certificate file> [ -P ] -a [tree] { filename  
[ size=VOLATILE ] [ hardlinks=value ] [ symlinks=value ]...| -f filename }
```

Delete Files from TSD

```
trustchk -d { filename...| ALL | -f filename }
```

Query TSD

```
trustchk -q { filename...| ALL | -f filename }
```

Switch to New Hashing Algorithm

```
trustchk -g [ SHA1 | SHA256 | SHA512 ]
```

System Scan

```
trustchk [-i] [ -x ] { -n | -t | -y } tree [dirpath.....]
```

Configure Policies

```
trustchk [-@ { WparName | ALL } ] -p { [ TE [ = ON | OFF ] ] [ CHKEXEC [ = ON | OFF ] ] [ CHKSHLIB [ = ON | OFF ] ] [ CHKSCRIPT [ = ON | OFF ] ] [ CHKKERNEXT [ = ON | OFF ] ] [ SIG_VER [ = ON | OFF ] ] [ STOP_UNTRUSTD [ = ON | OFF | TROJAN ] ] [ STOP_ON_CHKFAIL [ = ON | OFF ] ] [ LOCK_KERN_POLICIES [ = ON | OFF ] ] [ TEP [ = ON | OFF | PathList ] ] [ TLP [ = ON | OFF | PathList ] [ TSD_FILES_LOCK [ = ON | OFF | EXVOL ] ] [ TSD_LOCK [ = ON | OFF ] ] }
```

System Audit

```
trustchk [-r ] { -n | -t | -y } { filename... | ALL }
```

Using Alternate TSD File

```
trustchk -F TSDFile { -a | -d | -g | -q | -y | -n | -t }
```

Update TSD trustchk

```
trustchk -u <filename> [<attr>=value]
```

```
trustchk -k -s <private key file> -v certificate file [ -N ] { [ -D ] "OU = distinguished name" }
```

Note: The plus sign (+) is a special character that can be used only with a distinguished name for the **-D** option.

The following example shows how to use the plus sign as a special character in a distinguished name:

```
trustchk -k -s sign-key -v verify-key -N -D
"OU=IT + OU=jj, OU=zlab037.austin.ibm.com"
```

You cannot use the plus sign in any other format.

Description

The **trustchk** command is used in the following situations:

- [Managing the Trusted Signature Database](#)
- [Auditing the security state of the system](#)
- [Enabling the Trusted Execution Mechanism](#)
- [Configuring different policies for Trusted Execution](#)
- [Scanning the system for TROJAN detection](#)
- [Installing the software or interim fixes with Trusted Execution \(TE\) policies](#)

Managing the Trusted Signature Database

Privileged users use the **trustchk** command to add, delete, or list entries to the Trusted Signature Database (TSD). The TSD is a database of security attributes of the trusted files that are present on the system. The TSD is in the **/etc/security/tsd/tsd.dat** file. This database gets populated at installation time. It stores the security attributes of the trusted files that are present on the system. The following attribute list forms a part of a file definition (stanza):

Attributes	Usage
Owner	Name of the owner of the file. The owner ID cannot be used.
Group	Name of the group of the file. The group ID cannot be used.
Type	Type of the definition. Specifies if the definition belongs to a file, directory, first-in-first-out special files (FIFO), character device, block device, or a multiplexed device .
Mode	Permission bits, along with additional parameters specifying whether SETUID, SETGID, TCB, or SVTX bits are set in the file.
hardlink	Colon-separated list of hard links pointing to the file.

Attributes	Usage
symlink	Colon-separated list of symbolic links pointing to the file.
size	Size of the file in bytes.
cert_tag	ID of the digital certificate that was used to calculate the signature of this file.
signature	Digital signature of the file calculated using RSA algorithm.
hash_value	Cryptographic hash value of the file. By default, the SHA256 value is used to calculate the hash value.
accessauths	Access authorization on the object.
innateprivs	Innate privileges for the file.
inheritprivs	Inheritable privileges for the file.
authprivs	Privileges that will be assigned to users if they have the given authorizations.
secflags	File security flags associated with the object.

Note: You must include a blank line between stanzas when you specify multiple stanzas in an external file with the **-f** flag.

Auditing the security state of the system

To audit the security state of the system, you must check the security parameters stored in the TSD against the parameters of the actual files present on the system. Use the **trustchk** command to do so. Any discrepancy in the values is pointed to the user based on the input flags specified. To check all of the files that are listed in the TSD, use the **ALL** parameter in place of *filename*. You can specify a list of files separated by spaces on the command line.

Enabling the Trusted Execution function

To enable or disable the runtime integrity-verification function that is responsible for verifying of a file's cryptographic hash or signature before being started, use the **trustchk** command. To turn the Trusted Execution function on or off, use the **TE -p** flag.

Configuring different policies for Trusted Execution

To enable or disable different security policies that are used with the Trusted Execution mechanism, use the **trustchk** command. You can specify the following different policies:

Item	Description
CHKEXEC	Checks the integrity of executable file that belongs to the TSD before starting it.
CHKKERNEXT	Checks the integrity of the kernel extensions that belong to the TSD before loading them.
CHKSHLIB	Checks the integrity of shared libraries that belong to the TSD before loading them.
CHKSCRIPT	Checks the integrity of shell scripts that belong to the TSD before starting them.
LOCK_KERN_POLICIES	If this policy is disabled, then any policies can be enabled or disabled at any time. If this policy is enabled, then all of the other policies will be locked. To enable or disable a policy in such condition, disable the LOCK_KERN_POLICIES policy and then restart the system.

Item	Description
SIG_VER	Enables or disables the Runtime Signature Verification policy. When both the Runtime Signature Verification policy and the TE policy are enabled, the signature is used to verify the integrity of files instead of the hash value. Note: The SIG_VER policy is only applicable to files that belong to AIX system software.
STOP_ON_CHKFAIL	Stops the loading of files whose integrity check fails.
STOP_UNTRUSTD	Stops the loading of files that do not belong to the TSD.
TROJAN	Stops the loading of files that do not belong to the TSD and have one of the following security settings: • Have suid/sgid bit set • Linked to a file in the TSD • Have entry in the privcmds Database • Be linked to a file in the privcmds database
TE	Enables or disables Trusted Execution. Policies can only be activated when the TE option is set to ON.
TEP	Sets the value of Trusted Execution path, and enables or disables it. The Trusted Execution path consists of a list of colon-separated absolute paths, for example, the /usr/bin:/usr/sbin . When this policy is enabled, the files belonging to only these directory paths are allowed to be started. If an executable program that does not belong to the TEP is to be loaded, the program is blocked.
TLP	Sets the value of Trusted Library path, and enables or disables it. The Trusted Library Path consists of a list of colon-separated absolute paths, for example, the /usr/lib:/usr/ccs/lib . When this policy is enabled, the libraries belonging to only these directory paths can be loaded. If a program tries to load a library that does not belong to the TLP, the program is blocked.
TSD_FILES_LOCK	Disables opening of files belonging to the TSD in write mode.
EXVOL	Disables the opening of only the nonvolatile files that belong to the TSD in write mode. The volatile files can be changed.
TSD_LOCK	Disallows opening of a TSD file (/etc/security/tsd/tsd.dat) in write mode to disable editing of the TSD.

By default, the TSD defines all the files and programs that are part of the trusted computing base, but the privileged user or a member of the security group can choose to define only those files considered to be security-relevant.

The TE policies are stored in the **/etc/security/tsd/tepolicies.dat** file.

This command writes messages to the standard error log (**stderr**).

Scanning the system for TROJAN detection

Trustchk has the capability to detect the system for TROJAN, if any executable is present on the system and you do not have the entry in TSD and have one of the following security settings:

- have suid/sgid bit set
- linked to a file in the TSD
- have entry in the **privcmds** database
- be linked to a file in the **privcmds** database

Installing the software or interim fixes with Trusted Execution (TE) policies

If the Trusted Execution (TE) policy is turned on along with the TSD_LOCK policy or the TSD_FILE_LOCK policy, the **installp** and **emgr** commands fail. To continue with the installation, manually turn off the TSD_LOCK policy or the TSD_FILE_LOCK policy. The **emgr** and **installp** commands run successfully with TE policies if the TSD_LOCK policy or the TSD_FILE_LOCK policy is not turned on.

Flags

Item	Description
-a <i>filename</i>	Adds file definitions in the TSD. The definitions are read from a file (the -f option) or are calculated by the command if you specify the absolute file name. The following parameters can be specified by the user with the file name: size=VOLATILE Specifies the size of a file. This attribute can only use the VOLATILE value. The VOLATILE value indicates that the file that this definition belongs to is volatile in nature. The contents of the file change frequently, so during audits, the size, hash value and the signature of this file should not be checked. hardlinks=value Supplies the hard links to a file that cannot be computed independently by the trustchk command. symlinks=value Supplies the symbolic links to a file. -tree This tree parameter is used along with the -a flag. It supports adding of stanzas to the trustchk database recursively when the directory name is provided along with the -a flag. If the file name is mentioned, the stanza for the file name is added. To add a regular file to the TSD, you must specify the private key, or specify the signing key with the -s flag in ASN.1/DER in PKCS#8 format without pass phrase (that is, password) protection. You must also specify the associated certificate with the -v flag in ASN.1/DER. The associated certificate contains the public key that will be used to verify the signature of the file. The digital certificate that you specified is copied to a certificate store in the /etc/security/certificates file so that it can be used during system audits to verify the signatures of the file. To add non-regular files, such as devices, directories and FIFO (that is, the first-in-first-out file), the private key and certificate is not required.
-d	Deletes file definitions from the TSD. The name of the file whose stanza needs to be deleted from the TSD is specified at command line, or is placed in a file that can be specified with the -f flag.
-D	This flag is used along with the -k flag when you want to enter the issuer DN and the Subject DN from the command-line interface.

Item	Description
-f <i>filename</i>	Specifies that file definitions are to be read from the file specified with the <i>filename</i> parameter. The file (or stanza) name must end with a colon. There must be a blank line between each file name entry in the external file.
-F	Specifies that a different the TSD file be used as a reference. This flag can be used with the -a , -d , -g , -q , -n , -t , or -y flags.
-g [SHA1 SHA256 SHA512]	Migrates the TSD to a new hashing algorithm. All of the hash_value fields in the file definitions are recomputed and updated in the TSD. The following algorithms are supported: SHA1, SHA256 and SHA512. To see the currently active algorithm, specify the -g flag without any algorithm names.
-i	Only used with -n , -t , -y options and long with tree parameter. It will ignore the scanning of NFS mounted filesystem.
-k	Generates the certificate and the private key files by using the trustchk command. The key file name and certificate file names must be specified by -s and -v flag. The generated keys are saved in the files that are specified files by the -s and -v flags.
-n	Specifies the auditing mode, and indicates that the errors are to be reported. Any discrepancy between the attributes in the TSD and the actual file parameters are printed to the stderr . error file. To check all of the entries in the TSD, use the ALL parameter. To scan the entire system or directories for TROJAN detection, use with tree parameter.
-p	Configures Trusted Execution policies. You can turn on the policy configuration from command line, for example, policyA=ON. Specify a policy name to retrieve its current state (for example, trustchk -p CHKEXEC). The TE=ON option enables policies except the TEP and TLP policies that are not related to TE The TEP and TLP policies can be automatically turned ON or turn OFF. The TEP=ON option enables the TEP , and the TLP=ON option enables the TLP function.
-P	Prompts you to enter the password. This password is used to encrypt or decrypt the private-key file. This option can be used along with -a flag. When this flag is used with the trustchk -a command, it prompts you to enter the password which is used to decrypt the private-key file.
-q	Queries the TSD for a file name. Prints the entire list of security attributes, for example, stanza for the specified file name. To retrieve all of the entries of the TSD, use the ALL parameter instead of listing file path names.
-r	Specifies check that only the authorizations and privileges are to be checked. This flag is valid only on Enhanced RBAC system. To check all of the entries in the TSD, use the ALL flag.
-R <i>module_name</i>	Specifies that the values for the TSD policy and TE policy to be taken from the module specified instead of the local copy.
-s	Specifies the signing key used for signature calculation of a file while adding it to the TSD. The signing key is an RSA private key in ASN.1/DER in PKCS#8 format without pass phrase (that is, password) protection.

Item	Description
-t	Specifies the auditing mode and indicates that errors are to be reported with a prompt asking whether the error should be fixed. To check all of the entries in TSD, use the ALL option. To scan the entire system or directories for TROJAN detection, use with tree parameter.
-u	Updates the value of the specified attribute in TSD. If any of the rbac attributes are changed using the trustchk -u command, you must run the setkst explicitly. This updates the kernel table. Note: This flag supports the following attributes: Owner, group, mode, Hardlinks, symlinks, accessauths, innateprivs, inheritprivs, authprivs, secflags, t_innateprivs, t_inheritprivs, t_secflags, t_authprivs, t_accessauths, and type.
-v	Specifies the verification certificate that is associated with the signing key (using the -s flag). This certificate is copied into a certificate store in the /etc/security/certificate file, and is used to verify the file signature during auditing. If a certificate with the same certificate ID already exists in the store, then it is overwritten with a new certificate. The verification certificate is in ASN.1/DER format.
-x	Only used with -n , -t , -y options and long with tree parameter. Do not follow the symbolic link.
-y	Specifies the auditing mode, and indicates that errors are to be fixed and reported. To check all of the entries in the TSD, use the ALL parameter. To scan the entire system or directories for TROJAN detection, use with tree parameter.  Attention: Use the -y option with care. It might make a file unusable if the trustchk command encounters a discrepancy.
-@ WparName	Lists the TE polices of a system WPAR.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error has occurred.

Examples

1. To add a new file definition for **/usr/bin/ls** using private key located at **/home/guest/privkey.der** and an associated certificate at **/home/guest/certificate.der**, enter the following command:

```
trustchk -s /home/guest/privkey.der -v /home/guest/certificate.der
-a /usr/bin/ls
```

2. To add a file as a volatile file to the TSD using same pair of private key and certificate in the previous example, enter the following command:

```
trustchk -s /home/guest/privkey.der -v /home/guest/certificate.der
-a /usr/bin/passwd size=VOLATILE
```

3. To add a file **/usr/bin/ls** with a **/usr/local/bin/ls** hardlink to TSD using same pair of private key and certificate in the first example, enter the following command:

```
trustchk -s /home/guest/privkey.der -v /home/guest/certificate.der  
-a /usr/bin/ls hardlinks=/usr/local/bin/ls
```

4. To delete a file **/usr/bin/logname**, enter the following command:

```
trustchk -d /usr/bin/logname
```

5. To add file definitions stored in a file **/home/guest/filedef.in**, enter the following command:

```
trustchk -s /home/guest/privkey.der  
-v /home/guest/certificate.der  
-a -f /home/guest/filedef.in
```

6. To enable policy for checking executable file listed in the TSD on every load, follow these steps:

- a. Configure the policy by entering the following command:

```
trustchk -p CHKEEXEC=ON
```

- b. Activate the policy by entering the following command:

```
trustchk -p TE=ON
```

7. To check the integrity of all of the files belonging to the TSD, enter the following command:

```
trustchk -n ALL
```

8. To print the value of the currently active hash algorithm for TSD, enter the following command:

```
trustchk -g
```

9. To list all the policies of a WPAR, enter the following command:

```
trustchk -@ <wpar> -p
```

10. To list all the policies of all WPARs, enter the following command:

```
trustchk -@ ALL -p
```

11. To scan the whole system for a TROJAN detection report only, enter the following command:

```
trustchk -n tree
```

12. To scan only **dir /usr** for TROJAN detection and automatically fix them, enter the following command:

```
trustchk -y /usr
```

13. To scan the entire system for TROJAN detection, except NFS mounts filesystem, and fixes them interactively, enter the following command:

```
trustchk -i -t tree
```

14. To take the values from the LDAP server instead of the local copy, enter the following command:

```
trustchk -R LDAP -p
```

tset Command

Purpose

Initializes terminals.

Syntax

tset [**-e** *C*] [**-k** *C*] [**-i** *C*] [**-**] [**-s**] [**-I**] [**-Q**] [**-m** [*Identifier*] [*TestBaudRate*] :*Type*] ... [*Type*]

Description

The **tset** command enables you to set the characteristics of your terminal. It performs terminal-dependent processing, such as setting erase and kill characters, setting or resetting delays, and sending any sequences needed to properly initialize the terminal.

The **tset** command first determines the type of terminal involved (specified by the *Type* parameter). It then performs necessary initializations and mode settings. The type of terminal attached to each port is specified in the Object Data Manager (ODM) database. The terminfo database contains possible type names for terminals. If a port is not wired permanently to a specific terminal (that is, it is not hardwired), the **tset** command gives it an appropriate generic identifier, such as `dialup`.

When no flags are specified, the **tset** command reads the terminal type out of the **TERM** environment variable and re-initializes the terminal.

When the **tset** command is used in a startup script (the **.profile** file for **sh** users or the **.login** file for **csh** users), the script should include information about the type of terminal you will usually use on ports that are not hardwired. These ports are identified in the ODM database as `dialup`, `plugboard`, or `ARPANET`, among others. To specify which terminal type you usually use on these ports, use the **-m** flag (followed by the appropriate port type identifier), an optional baud rate specification, and the terminal type. If more than one mapping is specified, the first applicable mapping prevails. A missing port type identifier matches all identifiers. Any of the alternate generic names given in the **terminfo** database can be used as the identifier.

You can specify the baud rate in the **tset** command as you would with the **stty** command. The baud rate is compared with the speed of the diagnostic output (which should be the control terminal). The baud rate test can be any combination of the following characters:

- . (period)
- @ (at sign)
- < (less than sign)
- ! (exclamation point)

The @ (at sign) stands for the preposition at, and the ! (exclamation point) inverts the sense of the test. To avoid problems with metacharacters, place the **-m** flag argument inside ' ' (single quotes). Users of the **csh** command must also put a \ (backslash) before any ! (exclamation point).

The following example sets the terminal type to `adm3a` if the port in use is a `dialup` at a speed greater than 300 baud. It sets the terminal type to `dw2` if the port is a `dialup` port at a speed of 300 baud or less:

```
tset -m 'dialup>300:adm3a' -m dialup:dw2 -m 'plugboard:?adm3a'
```

If the *Type* parameter begins with a ? (question mark), you are prompted to verify the type. To use the specified type, press Enter. To use a different type, enter the type you want. In the example given, you are prompted to verify the `adm3` `plugboard` port type.

If no mapping applies and a final type option (not preceded by an **-m** flag) is given on the command line, that type is used. Otherwise, the default terminal type is the one identified in the ODM database. Hardwired ports should always be identified in the ODM database.

When the terminal type is known, the **tset** command engages in terminal driver mode setting. This usually involves setting:

- An initialization sequence to the terminal
- The single character erase and optionally the line-kill (full-line erase) characters
- Special character delays

Tab and new-line expansion are turned off during transmission of the terminal initialization sequence.

On terminals that can backspace but not overstrike (such as a CRT), and when the erase character is the default erase character (# on standard systems), the erase character is changed to Backspace (Ctrl-H).

Flags

Item	Description
-e <i>C</i>	Sets the erase character to the character specified by the <i>C</i> parameter. The default is the backspace character.
-I	Suppresses transmission of terminal initialization strings.
-i <i>C</i>	Sets the interrupt character to the character specified by the <i>C</i> parameter. The <i>C</i> parameter defaults to ^C (caret C). The ^ (caret) character can also be used for this option.
-k <i>C</i>	Sets the line-kill character to the character specified by the <i>C</i> parameter. The <i>C</i> parameter defaults to ^X (caret X). The ^ (caret) character can also be used for this option.
-m <i>IdentifierTestbaudRate:Type</i>	Specifies which terminal type (in the <i>Type</i> parameter) is usually used on the port identified in the <i>Identifier</i> parameter. A missing identifier matches all identifiers. You can optionally specify the baud rate in the <i>TestBaudRate</i> parameter.
-Q	Suppresses printing of the <code>Erase set to</code> and <code>Kill set to</code> messages.
-s	Prints the sequence of cs h commands that initialize the TERM environment variable, based on the name of the terminal decided upon.
-	The name of the terminal decided upon is output to standard output. This is the TERM environment variable.

Examples

The following examples all assume the Bourne shell and usage of the **-** flag. If you use the **cs**h command, use the preceding variations. A typical use of the **tset** command in a **.profile** or **.login** file includes the **-e** and **-k** flags, and often the **-n** or **-Q** flags as well. To streamline the examples, these flags have not been included here.

Note: Make sure to enter the **tset** command all on one line regardless of the number of lines used in the example.

1. Now you are a 2621 terminal. Do not use the following example in your **.profile** file, unless you are always a 2621 terminal.

```
export TERM; TERM=\`tset \- 2621\`
```

2. You have an h19 terminal at home that you dial up on, but your office terminal is hardwired and specified in the ODM database.

```
export TERM; TERM=\`tset \- \-m dialup:h19"\`
```

3. You have a switch that connects everything to everything, making it nearly impossible to key on what port you are coming in. You use a vt100 in your office at 9600 baud and dial up from home on a 2621 to switch ports at 1200 baud. Sometimes, you use a different terminal at work. At high speeds, you want to verify your terminal type, but at 1200 baud, you are always on a 2621. Note how the quotation marks protect the greater-than sign and the question mark from interpretation by the shell.

```
export TERM; TERM=\`tset \- \-m 'switch>1200:?vt100' \-m 'switch<=1200:2621'\`
```

If none of the conditions hold, the terminal type specified in the ODM database is used.

4. The following entry is appropriate if you always dial up at the same baud rate on many different terminals. Your most common terminal is an adm3a. You are always prompted to verify the terminal type, which defaults to adm3a.

```
export TERM; TERM=\tset \- \?adm3a\'
```

5. If the ODM database is not properly installed and you want to key entirely on the baud rate, type:

```
export TERM; TERM=\tset \- \-m 'switch>1200:vt100' \-m  
'switch<=1200:2621'
```

6. You dial up at 1200 baud or less on a Concept100, sometimes over switch ports and sometimes over regular dialups. You use various terminals at speeds higher than 1200 over switch ports, most often the terminal in your office, which is a vt100. However, sometimes you log in from the university over the ARPANET; in this case, you are on an ALTO emulating a dm2500. You also often log in on various hardwired ports, such as the console, all of which are properly entered in the ODM database. To set your erase character to Ctrl-H and your kill character to Ctrl-U, type:

```
export TERM  
TERM=\tset \-e \-k(hat)U \-Q \- "-m 'switch<1200:concept100'  
"-m 'switch:vt100' \-m dialup:concept100 "1-m arpanet: dm2500"
```

This also prevents the **tset** command from printing the following line:

```
Erase set to Backspace, Kill set to Ctrl-U
```

7. To set the erase character to a control character, type:

```
tset -e ^Y
```

Files

Item	Description
/usr/share/lib/terminfo	Contains the terminal capability database.

tsh Command

Purpose

Invokes the trusted shell.

Syntax

Press in sequence: the Ctrl+X, Ctrl+R keys.

tsh Command

Description

The **tsh** command is a command interpreter that provides greater security than the Korn shell (the standard login shell). Generally, a user calls the **tsh** shell by pressing Ctrl+X, Ctrl+R, the secure attention key (SAK) sequence, after a login. The **tsh** shell also can be invoked by defining it as the login shell in the **/etc/passwd** file.

To use the SAK sequence to invoke the trusted shell, the terminal the user is using must have SAK enabled, and the user must be allowed to use the trusted path. See the **Trusted Computing Base** in *Operating system and device management* for information on enabling SAK on a terminal, and see

the **/etc/security/user** file and the **chuser** command for information on allowing a user to access the trusted path.

To exit from the **tsh** shell, use any of the following commands: the **logout** command, **shell** command, **su** command. The **logout** command ends the login session, while the other commands execute the user's initial program and continue the login session.

The trusted shell differs from the Korn shell in the following ways:

- The function and alias definitions are not supported. Alias definitions are only supported in the **/etc/tsh_profile** file.
- The **IFS** and **PATH** environment variables cannot be redefined.
- Only trusted programs can be run from the **tsh** shell.
- The history mechanism is not supported.
- The only profile used is the **/etc/tsh_profile** file.
- The trusted shell has the following built-in commands:

Item	Description
logout	Exits the login session and terminates all processes.
shell	Re-initializes the user's login session. The effect is the same as logging in to the system.
su	Resets the effective ID to the user's identity on the system and executes another trusted shell.

Security

Access Control: This command should be a standard user program and have the **trusted computing base** attribute.

Files Accessed:

Mode	File
r	/etc/tsh_profile

Examples

To invoke the trusted shell, press the Ctrl+X, Ctrl+R key sequence, the secure attention key (SAK).

Files

Item	Description
/usr/bin/tsh	Contains the tsh command.
/etc/tsh_profile	Contains initialization commands for the trusted shell.
/etc/passwd	Contains basic user attributes.
/etc/security/user	Contains the extended attributes of users.
/etc/security/login.cfg	Contains configuration information.

tsig-keygen Command

See the [ddns-confgen](#) command.

tsm Command

Purpose

Provides terminal state management.

Syntax

tsm *Port*

Description

The **tsm** command invokes the terminal state manager, which controls the ports used in the trusted path. The functions are:

- Establishing line communication modes and discipline - functions performed by the **getty** command.
- Verifying the user's account and identity, and setting the initial process credentials and environment - functions performed by the **login** command.
- Performing trusted path management if the secure attention key (SAK) is enabled for the port and the system login program is used.

Note: The **tsm** command is not entered on the command line.

Trusted path management occurs in two phases:

Item	Description
login	This phase is in effect if a user has not successfully logged in. If the secure attention key (SAK) signal is detected, the system restarts getty-login type processing. The next login puts the user into the trusted state, if the port and the user support the trusted state.
shell	This phase occurs after successful user authentication. The command functions according to the user's tpath attribute. The following values are valid: on Provides standard trusted path management. When the secure attention key (SAK) signal is detected, all processes that access the port, except the tsm process and its siblings (including the trusted shell), are terminated the next time an attempt is made to access the port. The port is reset to its initial state and is marked as trusted, and the trusted shell command (the tsh command) is executed. notsh The user session terminates when the secure attention key (SAK) signal is detected. always The user is not allowed off the trusted path. The user's shell will always be the trusted shell, tsh. nosak The secure attention key (SAK) is disabled for the terminal, and the user's initial program runs.

You can configure the **tsm** command to create your home directory at your login if you do not have a home directory already. The **tsm** command calls the **mkuser.sys** command to create the home directory and customize the account. To enable this capability, set the **mkhomeatlogin** attribute of the **usw** stanza in the **/etc/security/login.cfg** file to true.

Security

Access Control: This command should grant execute (x) permission to any user. The command should be setuid to the root user and have the **trusted computing base** attribute.

Files Accessed:	
Mode	File
r	/etc/objrepos/CuAt
r	/usr/lib/objrepos/PdAt
r	/etc/security/login.cfg
r	/etc/security/user

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

To provide terminal state management on `tty0`, add the following line to the `/etc/inittab` file:

```
tty0:2:respawn:/usr/sbin/tsm /dev/tty0
```

This initializes the port `/dev/tty0` and sets up the characteristics of the port.

Files

Item	Description
<code>/usr/sbin/tsm</code>	Contains the tsm command.
<code>/etc/security/login.cfg</code>	Contains configuration information.
<code>/etc/security/user</code>	Contains extended user attributes.

tsort Command

Purpose

Sorts an unordered list of ordered pairs (a topological sort).

Syntax

```
tsort [ - ] [ File ]
```

Description

The **tsort** command reads from *File* or standard input an unordered list of ordered pairs, builds a completely ordered list, and writes it to standard output.

The input *File* should contain pairs of non-empty strings separated by blanks. Pairs of different items indicate a relative order. Pairs of identical items indicate presence, but no relative order. You can use the **tsort** command to sort the output of the **lorder** command.

If *File* contains an odd number of fields, an appropriate error message is displayed.

Flag

Item	Description
—	(Double hyphen) Interprets all arguments following the — flag as file names. If the file is named —, use <code>tsort —</code> .

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Files

Item	Description
<code>/usr/ccs/bin/tsort</code>	Contains the tsort command.
<code>/usr/ccs/bin/tsort</code>	Contains symbolic link to the tsort command.

ttt Command

Purpose

Starts the tic-tac-toe game.

Syntax

ttt [-e] [i]

Description

The **ttt** command starts the tic-tac-toe game. This is a learning version but it learns slowly. It loses nearly 80 games before completely mastering the game. When you start the game you are prompted Accumulated knowledge? (Yes or No). Entering y provides the computer with knowledge gained from previous games.

You are always X and your opponent is always O. You can either make the first move or pass to your opponent. To pass, press the enter key when prompted Your move? at the beginning of the game. The first to get three in a row wins the game. For example:

```
new game
123
456
789
Your move?
1
X03
456
789
Your move?
9
X00
456
78X
Your move?
5
You win
```

In the example, your first move was to place an X where 1 was located. The computer placed an O where the 2 was located. The game progressed until you had three in a diagonal row (1,5, 9). The game repeats until you quit. To quit the game, press the Interrupt (Ctrl-C) or End Of File (Ctrl-D) key sequence.

Flags

Item	Description

-e	Increases the speed of the learning.
-i	Displays the instructions prior to the start of the game.

Files

Item	Description
\$HOME/ttt.a	Specifies the location of the learning file.
/usr/games	Specifies the location of the system's games.

tty Command

Purpose

Writes to standard output the full path name of your terminal.

Syntax

/usr/bin/tty [**-s**]

Description

The **tty** command writes the name of your terminal to standard output.

If your standard input is not a terminal and you do not specify the **-s** flag, you get the message `Standard input is not a tty`.

The following environment variables affect the execution of the **tty** command:

Item	Description
LANG	Determines the locale to use for the locale categories when neither the LC_ALL variable nor the corresponding environment variable beginning with LC_ specifies a locale.
LC_ALL	Determines the locale to be used. This variable overrides any values for locale categories that are specified by any other environment variable beginning with LC_ or by the LANG variable.
LC_CTYPE	Determines the locale for the interpretation of sequences of bytes of text data as characters. For example, this variable may specify multi-byte characters instead of single-byte characters.
LC_MESSAGES	Determines the language for messages.

Flags

Item	Description

-s	Suppresses reporting the path name.

Exit Status

This command returns the following exit values:

Item	Description

0	Standard input is a terminal.
1	Standard input is not a terminal.
>1	An error occurred.

Examples

1. To display the full path name of your display:

```
tty
```

2. To test whether or not the standard input is a terminal:

```
if tty -s
then
echo 'Enter the text to print:' >/dev/tty
qprt -
fi
```

If the standard input is a terminal, this displays the message "Enter the text to print:" as a prompt and prints the text that the user types. If the standard input is not a terminal, this displays nothing; it merely prints the text read from the standard input.

The echo . . . >/dev/tty displays the prompt on the screen even if you redirect the standard output of the shell procedure. This way the prompt is never written into an output file. The special file **/dev/tty** always refers to your terminal, although it also has another name such as **/dev/console** or **/dev/tty2**.

Files

Item	Description
/usr/bin/tty	Contains the tty command.
/dev/tty	Specifies the tty pseudo device.

tunchange Command

Purpose

Updates one or more tunable stanzas in a file.

Syntax

tunchange -f *Filename* (**-t** *Stanza* ({ **-o** *Parameter*[=*Value*] } | **-D**) | **-m** *Filename2*)

Description

The **tunchange** command unconditionally updates a tunable file. It can also merge a second file with the current file.

Note: No message will be displayed (even when a parameter of type bosboot is changed).

Flags

Item	Description
-f <i>Filename</i>	Name of the updated tunable file. If the name does not include the '/' (forward slash) character, it is considered to be relative to /etc/tunables .
-t <i>Stanza</i>	Name of the stanza to update. <i>Stanza</i> is either schedo , vmo , ioo , no , nfso , or raso . <i>Stanza</i> corresponds to the name of the command which can update the parameter or parameters specified by the -o flag.
-o <i>Parameter=Value</i>	Parameter to be set to <i>Value</i> . It must be valid in the <i>Stanza</i> specified by the -t flag and consistent with the other parameters of the file specified by the -f flag.
-D	Resets all parameters of the <i>Stanza</i> to their default value.
-m <i>Filename2</i>	Merges the <i>Filename2</i> file with the current <i>Filename</i> file.

Exit Status

Item	Description
0	Changes were correctly applied.
>0	One of the following conditions caused an error: • The specified <i>Filename</i>, <i>Filename2</i>, or <i>Stanza</i> was invalid. • <i>Parameter=Value</i> was invalid for the Parameter. • No message was provided.

Examples

1. To update the **pacefork** parameter in the **/etc/tunables/nextboot** file, type:

```
tunchange -f nextboot -t schedo -o pacefork=10
```

2. To update the **pacefork** parameter in the **/home/mine/mytunable** file, type:

```
tunchange -f /home/mine/mytunable -t schedo -o pacefork=10
```

3. To reset all **schedo** stanza parameters to their default value in the **/etc/tunables/nextboot** file, type:

```
tunchange -f nextboot -t schedo -D
```

4. To merge the **/home/mine/mytunable** file with the **/etc/tunables/nextboot** file, type:

```
tunchange -f nextboot -m /home/mine/mytunable
```

Files

Item	Description
/usr/sbin/tunchange	Contains the tunchange command.
/etc/tunables/	Contains the default tunable files.

tuncheck Command

Purpose

Validates the file that contains the tunable parameters.

Syntax

tuncheck [**-r** [**-K**] | **-p**] **-f** *Filename*

Description

The **tuncheck** command validates the file that contains the tunable parameters. All tunable parameters listed in the specified file are checked for range and dependencies. If a problem is detected, a warning is issued.

The **tuncheck** command validates the following items:

Validation against the current context

Checks whether the tunable parameters in the specified file can be applied immediately. Tunable parameters that are not listed in *Filename* are interpreted as current values. The checking fails if a tunable of type **Incremental** is listed with a smaller value than its current value; it also fails if a tunable of type **Bosboot** or **Reboot** is listed with a different value than its current value.

Validation against the next boot context

Checks whether the tunable parameters in the specified file can be applied during a reboot, that is, if it could be a valid nextboot file. The file validation does not fail if a tunable parameter of type **Incremental** is specified with a smaller value than its current value. If a tunable of type **Bosboot** or **Reboot** is listed with a different value than its current value, a warning is issued but the checking does not fail.

> Validation against the next Live Update context

Checks whether the tunable parameters in the specified file can be applied during the Live Update operation, that is, whether the file can be a valid nextliveupdate file. The validation does not fail if a tunable parameter of type **Incremental** is specified with a smaller value than its current value. If a tunable parameter of type **Bosboot** or **Reboot** is listed with a different value than its current value in the specified file, a warning message is displayed but the file validation does not fail. <

Additionally, warnings are issued if *Filename* contains unknown stanzas, or unknown tunables in a known stanza. However, that does not make the checking fail.

Upon success, the **AIX_level**, **Kernel_type** and **Last_validation** fields in the info stanza of the checked file are updated.

Flags

Item	Description
-f <i>Filename</i>	Specifies the name of the tunable file to be checked. If it does not contain the '/' (forward slash) character, the name is relative to /etc/tunables .
-K	Checks the specified file in a boot and Live Update context.
-p	Checks <i>Filename</i> in both current and boot contexts. This is equivalent to running tuncheck twice, one time without any flag and one time with the -r flag.
-r	Checks <i>Filename</i> in a boot context.

If **-p** or **-r** are not specified, *Filename* is checked according to the current context.

Tuning Parameter Types

Item	Description
Dynamic	Can be changed at any time.
Static	Can never be changed
Reboot	Can only be changed during the reboot sequence
Bosboot	Can only be changed by running bosboot and rebooting the machine
Mount	Changes made are only effective for future filesystems or directory mountings
Incremental	Can only be incremented, except at boot time.
Connect	Changes are only effective for future socket connections.

Exit Status

0

Filename is valid.

>0

Filename is invalid, message have been provided.

Examples

1. To check whether mytunable can be applied immediately, type:

```
tuncheck -f ./mytunable
```

2. To check whether /etc/tunables/nextboot can be applied during a reboot, type:

```
tuncheck -r -f nextboot
```

3. To check whether /etc/tunables/nextboot can be applied immediately and after a reboot, type:

```
tuncheck -p -f nextboot
```

4. **>** To check whether the tunable parameters in the mytunables file can be applied during a reboot and Live Update operation, enter the following command:

```
tuncheck -r -K -f ./mytunable
```

<

Files

Item	Description
/usr/sbin/tuncheck	Contains the tuncheck command.
/etc/tunables	Contains all the tunable files.

tunedefault Command

Purpose

Reset all tunable parameters to their default value.

Syntax

tunedefault [[-r](#) [[-K](#)] | [-p](#)]

Description

The **tunedefault** command starts all the tuning commands (**ioo**, **vmo**, **schedo**, **no**, **nfso**, and **raso**) with the **-D** flag. This resets all the AIX tunable parameters to their default value, except for parameters of type **Bosboot** and **Reboot**, and parameters of type **Incremental** set at values bigger than their default value, unless **-r** was specified. Error messages are displayed for any parameter change impossible to make.

Flags

Item	Description
> > -K	Defers the reset operation of tunable parameters to its default value until the next Live Update operation. When you specify the -r flag, the tunedefault command resets all tunable parameters in the specified nextboot and nextliveupdate files to its default value. The -r flag clears the stanzas in the /etc/tunables/nextboot file, and if necessary, prompts for the bosboot operation and displays a warning message that a reboot operation is required to apply the changes. The -K flag adds the stanzas with default value of the tunable parameters in the /etc/tunables/nextliveupdate file in the nextliveupdate file. This updated nextliveupdate file is used for the next Live Update operation. < <
-p	Makes the changes permanent: resets all the tunable parameters to their default values and updates the /etc/tunables/nextboot file.
-r	Defers the reset to their default value to next reboot. This clears stanzas in the /etc/tunables/nextboot file, and if necessary, proposes bosboot and warns that a reboot is needed

Tunable Parameter Types

Item	Description
Dynamic	Can be changed at any time.
Static	Can never be changed
Reboot	Can only be changed during the reboot sequence
Bosboot	Can only be changed by running bosboot and rebooting the machine
Mount	Changes made are only effective for future filesystems or directory mountings
Incremental	Can only be incremented, except at boot time.
Connect	Changes are only effective for future socket connections.

Examples

1. To permanently reset all tunable parameters to their default values, enter the following command:

```
tunedefault -p
```


All of the tuning commands are launched with the **-Dp** flags. This resets all the tunable parameters to their default value. This also updates the **/etc/tunables/nextboot** file. This command completely and permanently resets all tunable parameters to their default values.

2. To defer the setting of all tunable parameters until next reboot, enter the following command:

```
tundefault -r
```

This command calls all tuning commands with the **-Dr** flags. This clears all of the stanzas in the **/etc/tunables/nextboot** file, and if necessary, proposes **bosboot** and displays a message warning that a reboot is necessary to make the changes effective.

3. **>** To defer setting all tunable parameters until next reboot and next Live Update operation, enter the following command:

```
tundefault -r -K
```

This command calls all tuning commands with the **-DrK** flags. This command clears all of the stanzas in the **/etc/tunables/nextboot** file, if necessary, proposes a **bosboot** operation and displays a message warning that a reboot is necessary to make the changes effective. In addition, this command also adds stanzas with the default values in the **/etc/tunables/nextliveupdate** file if a non-default value was set for the tunable parameter for the next Live Update operation. **<**

Files

Item	Description
/usr/sbin/tundefault	Contains the tundefault command.
/etc/tunables/	Contains all the tunable files.

tunrestore Command

Purpose

Restores tunable parameter values from a file.

Syntax

tunrestore [**-r** [**-K**]] **-f** *Filename*

tunrestore -R

Restriction: **tunrestore -R** can only be called from **inittab**.

Description

The **tunrestore** command restores all tunable parameters values stored in a file.

The **tunrestore -f** *Filename* immediately applies *Filename*. All tunables listed in *Filename* are set to the value defined in this file. Tunables not listed in *Filename* are kept unchanged. Tunables explicitly set to **DEFAULT** are set to their default value.

The **tunrestore -r -f** *Filename* applies *Filename* for the next boot. This is achieved by checking the specified file for inconsistencies (the equivalent of running **tuncheck** on it) and copying it over to **/etc/tunables/nextboot**. If **bosboot** is necessary, the user will be offered to run it.

> The **tunrestore -r -K -f** *Filename* command restores all the tunable parameter in the specified **nextboot** and **nextliveupdate** files. This command checks the specified file for inconsistencies (similar to running the **tuncheck** command) and copies the file to **/etc/tunables/nextboot** and **/etc/tunables/nextliveupdate** files. If required, this command prompts you to run the **bosboot** operation. **<**

The **tunrestore -R** is only used during reboot. All of the tunable parameters that are not yet set to the value defined in the **nextboot** file are modified. Tunables that are not listed in the **nextboot** file are forced to their default value. All actions, warnings and errors are logged into the **/etc/tunables/lastboot.log** file. Note that when modification is made to restricted tunables, a system **errlog** entry is added, including the list of all tunable commands controlling the modified restricted tunables and a reference to the **/etc/tunables/lastboot.log** file.

Additionally, a new tunable file called **/etc/tunables/lastboot** is automatically generated. That file has all of the tunables listed with numerical values. The values representing default values are marked with the comment **DEFAULT VALUE**. The values that are different from the default values for restricted tunables are marked with the comment **# RESTRICTED not at default value**. The info stanza of the new tunable file includes the checksum of the **/etc/tunables/lastboot.log** file to make sure pairs of the **lastboot/lastboot.log** files can be identified.

Flags

Item	Description
-f <i>Filename</i>	Specifies the name of the tunable file to apply. If it does not contain the '/' (forward slash) character, the name is relative to /etc/tunables .
> >-K	Updates the /etc/tunables/nextboot and /etc/tunables/nextliveupdate files with the tunable parameter values that are listed in the specified file. <<
-r	Updates the /etc/tunables/nextboot file with the tunable parameter values that are listed in the specified file.
-R	Restores /etc/tunables/nextboot during boot process.

Tunable Parameter Types

Item	Description
Dynamic	Can be changed at any time.
Static	Can never be changed
Reboot	Can only be changed during the reboot sequence
Bosboot	Can only be changed by running bosboot and rebooting the machine
Mount	Changes made are only effective for future filesystems or directory mountings
Incremental	Can only be incremented, except at boot time.
Connect	Changes are only effective for future socket connections.

Examples

1. To restore all tunable values stored in **/etc/tunables/mytunable**, enter:

```
tunrestore -f mytunable
```

2. To validate **/etc/tunables/mytunable** and make it the new **nextboot** file, enter:

```
tunrestore -r -f mytunable
```

3. **>|**To validate the **/etc/tunables/mytunable** file and make it new **nextboot** and **nextliveupdate** file, enter the following command:

```
tunrestore -r -K -f mytunable
```

Files

Item	Description
<code>/usr/sbin/tunrestore</code>	Contains the tunrestore command.
<code>/etc/tunables</code>	Contains tunable files.
<code>/etc/tunables/nextboot</code>	Contains the values to be applied during the next boot.
> > <code>/etc/tunables/nextliveupdate</code>	Contains the values to be applied during the next Live Update operation. < <
<code>/etc/tunables/lastboot</code>	Contains the values of all tunable parameters after the last boot.
<code>/etc/tunables/lastboot.log</code>	Contains messages, warnings and errors emitted by tunrestore during the last boot.

tunsave Command

Purpose

Saves current tunable parameter values to a file.

Syntax

```
tunsave [ -a | -A ] -f -F Filename [ -d Description ]
```

Description

The **tunsave** command saves the current state of tunable parameters in a file.

If *Filename* does not already exist, a new file is created. If it already exists, an error message prints unless the **-F** flag is specified, in which case, the existing file is overwritten.

Note that the saved restricted tunables that have been modified to a value different from the default value, are flagged with a comment `# RESTRICTED not at default value`, appended to the line.

Flags

Item	Description
-a	Saves all tunable parameters, including those who are currently set to their default value. These parameters are saved with the special value <code>DEFAULT</code> .
-A	Saves all tunable parameters, including those who are currently set to their default value. These parameters are saved numerically, and a comment, <code># DEFAULT VALUE</code> , is appended to the line to flag them.
-d <i>Description</i>	Specifies the text to use for the <i>Description</i> field. Special characters must be escaped or quoted inside the <i>Description</i> field.
-f <i>Filename</i>	Specifies the name of the tunable file where the tunable parameters are saved. If <i>Filename</i> already exists, an error message prints. If it does not contain the <code>'/'</code> (forward slash) character, the <i>Filename</i> is relative to /etc/tunables .

Item	Description
-F <i>Filename</i>	Specifies the name of the tunable file where the tunable parameters are saved. If <i>Filename</i> already exists, the existing file is overwritten. If it does not contain the '/' (forward slash) character, the <i>Filename</i> is relative to /etc/tunables .

Examples

1. To save all tunables different from their default value into **/etc/tunables/mytunable**, enter:

```
tunsave -f mytunable
```

2. To save all tunables, including those who are currently set to their default value, but replace the default values with the special value DEFAULT, enter:

```
tunsave -a -f /home/admin/mytunable
```

3. To save all tunables, including those who are currently set to their default value using all numerical values, but flag the default values with the comment DEFAULT VALUE, enter:

```
tunsave -A -f mytunable
```

Files

Item	Description
/usr/bin/tunsave	Contains the tunsave command.
/etc/tunables	Contains all the saved files.

turnacct Command

Purpose

Provides an interface to the **accton** command to turn process accounting on or off.

Syntax

/usr/sbin/acct/turnacct on | off | switch

Description

The **turnacct** command provides an interface to the **accton** command to turn process accounting on or off. You must specify whether you want process accounting on or off, because there is no default.

The **switch** flag turns off accounting and moves the current active data file (**/var/adm/pacct**) to the next free name in the **/var/adm/pacctincr** file, where *incr* is a number starting at 1 and increased by one for each additional **pacct** file. After moving the **pacct** file, the **turnacct** command again turns on accounting.

The **turnacct switch** command is usually called by the **ckpacct** command, running under the **cron** daemon, to keep the active **pacct** data file a manageable size.

Security

Access Control: This command should grant execute (x) access only to members of the adm group.

Files

Item	Description
/usr/sbin/acct	Contains the path to the accounting commands.
/var/adm/pacct	Contains the current file for process accounting.
/var/adm/pacct*	Used if the pacct file gets too large.

turnoff Command

Purpose

Sets the permission codes off for files in the **/usr/games** directory.

Syntax

turnoff

Description

The **turnoff** command sets the permission codes of files in the **/usr/games** directory. Root user authority is required to run this command.

The **turnoff** command looks for files in **/usr/games** whose permissions are set to 111 and sets these permissions to 000. If you install any new games in the **/usr/games** directory, set these permissions to 111.

Files

Item	Description
/usr/games	Contains the location of the system's games.

turnon Command

Purpose

Sets permission codes on for files in the games directory.

Syntax

turnon

Description

The **turnon** command sets the permission codes of files in the **/usr/games** directory. Root user authority is required to run this command.

The **turnon** command looks for files with permissions set to 000 and sets them to 111 (execute permission for all users). If you install any new games in the **/usr/games** directory, set these permissions to 111.

File

Item	Description
/usr/games	Contains the location of the system's games.

tvf Command

Purpose

Provides a trusted editor with a full screen display.

Syntax

```
tvf [ -l ] [ -R ] [ -w Number ] [ -c [ Subcommand ] ] [ File ... ]
```

Description

The **tvf** command calls the **tvf** editor, a trusted version of the **vi** editor, to edit the file or files specified by the *File* parameter. Files are edited in the order specified. If you do not provide a file name, the command opens a new file in which you can create text, but if you try to save the text to a file, you are prompted to add a file name to the save command, such as **:w File**. See the [Examples](#) section for more information.

You enter and leave the **tvf** editor in command mode, but to add or change text, you must enter text input mode. See the description of text input mode for information about the subcommands that initiate text input mode. To leave text input mode, press the **Esc** key. This returns you to command mode where you can [save the text](#) to a file with one of the **:w** commands, and [exit](#) the **tvf** editor, for example, with the **:q** command.

Because the full-screen display editor started by the **tvf** command is based on the **ex** editor, you can use the **ex** subcommands within the **tvf** editor. Subcommands function at the cursor position on the display screen.

The **tvf** editor makes a copy of the file you are editing in an edit buffer. The contents of the file are not changed until you save the changes.

Note: Several functions of the **vi** editor are not supported by the **tvf** editor. If you refer to information on the **vi** editor, be aware that the **-r** flag, the **-t** flag, shell escapes, user-defined macros, key mapping, and setting **vi** options permanently are not supported by the **tvf** editor.

tvf Editor Limitations

The maximum limits of the **tvf** editor assume single-byte characters. The limits are as follows:

- 256 characters per global command list
- 2048 characters in a shell escape command
- 128 characters in a string-valued option
- 30 characters in a tag name
- 524,230 lines silently enforced
- 128 map macros with 2048 characters total

Editing Modes

The **tvf** editor operates in the following modes:

Item	Description
command mode	The tvf editor starts in command mode. Any subcommand can be called except those that only correct text during text input mode. To see a description of the subcommands, refer to the topics in Subcommands for the tvf Editor . To identify the subcommands that cannot be called from command mode, refer to Changing Text While in Input Mode . The tvf editor returns to command mode when subcommands and other modes end. Press the Esc key to cancel a partial subcommand.

Item	Description
text input mode	The tvi editor enters text input mode when you use a permitted command that adds or changes text. To see a list of subcommands that initiate text input mode, refer to Adding Text to a File and the subcommands that change text from command mode, the C subcommand and the cx subcommands. After entering one of these subcommands, you can edit text with any of the subcommands that function in text input mode. To see a description of the subcommands, refer to the topics in "Subcommands for the tvi Editor". To return to command mode from text input mode, press the Esc key for a typical exit or press the Ctrl+C keys to create an INTERRUPT signal.
last line mode	Some subcommands read input on a line displayed at the bottom of the screen. These subcommands include those with the prefix : (colon), / (slash), and ? (question mark). When you enter the initial character, the tvi editor places the cursor at the bottom of the screen so you can enter the remaining command characters. To run the subcommand, press the Enter key. To cancel the subcommand, press the Ctrl+C keys to create an INTERRUPT signal. When you use the : (colon) to enter last line mode, the following characters have special meaning when used before commands that specify counts: % All lines regardless of cursor position \$ Last line · Current line

Customizing the tvi Editor

You can customize the **tvi** editor on a temporary basis by following the directions in ["Setting vi Editor Options"](#). The section on "Setting vi Options Permanently" is *not* applicable to the **tvi** editor.

Subcommands for the tvi Editor

Information on **vi** editor subcommands that are applicable to the **tvi** editor is summarized in the following list:

- [vi General Subcommand Syntax](#).
- [vi Subcommands for Adjusting the Screen](#).
- [Editing Text with the vi Editor](#).
- [Entering Shell Commands within the vi Editor](#) is *not* supported by the **tvi** editor.
- [Manipulating Files with the vi Editor](#).
- [Subcommands for Interrupting and Ending the vi Editor](#).

Flags

Item	Description
-c [<i>Subcommand</i>]	Carries out the ex editor subcommand before editing begins. This provides a line-oriented text editor. When a null operand is entered for the <i>Subcommand</i> parameter, as in -c ' ', the editor places the cursor on the last line of the file.

Item	Description
-l	Enters the editor in LISP mode. In this mode, the editor indents appropriately for LISP code, and the () , { } , [] , and]] subcommands are modified to act appropriately for LISP. These subcommands place the cursor at the specified LISP function. For more information on the LISP subcommands, refer to Moving to Sentences, Paragraphs, and Sections .
-R	Sets the readonly option to protect the file against overwriting.
-w Number	Sets the default window size to the value specified by the <i>Number</i> parameter. This is useful when you use the editor over a low-speed line.
+ [Subcommand]	Same as the -c Subcommand.

Security

Access Control: This command should grant execute (x) access to all users and have the **trusted computing base** attribute.

Auditing Events:

Event	Information
TVI	filename

Examples

1. To call a trusted editor to edit the plans file, type:

```
tvi plans
```

This command puts the **tvi** editor into command mode. To add or change text, you must enter text input mode or use a command accepted in command mode. For more information, refer to the description of [text input mode](#).

2. To save the text you create with the **tvi** editor, leave text input mode by pressing the Esc key, and then enter one of the save commands **:w**, **:w File**, or **:w! File**, for example:

```
:w plans
```

In this example, a file name, such as plans, is needed if you gave the **tvi** command without specifying a file name. If the file is already named, the **:w** command would not need the *File* parameter. If you want to overwrite an existing file, use the **:w! File** command, specifying the file you want to overwrite with the *File* parameter.

If you try to save an unnamed file without supplying a file name, the following message appears:

```
No current filename
```

If this happens, repeat the **:w** command with a file name.

3. To exit the **tvi** editor from text input mode, press the Esc key to type command mode, and then type:

```
:q!
```

If the editor already is in command mode, you do not need to press the Esc key before giving the quit (**q!**) command.

Files

Item	Description
<code>/usr/bin/tvi</code>	Contains the tvi command.

twconvdict Command

Purpose

Converts other user dictionary to the operating system user dictionary.

Syntax

```
twconvdict [ -i Type ] [ -v CodePage ] [ -f Source ] [ -t Target ]
```

Description

The **twconvdict** command converts a dictionary to an operating system user dictionary. The supported code pages are SOPS, PS55 and ET. The dictionary type include both Tseng_Jye and Phonetic user dictionaries.

Flags

Item	Description
-f <i>Source</i>	Specifies the name of font file to convert.
-i <i>Type</i>	Specifies the type of dictionary to convert to. <i>Type</i> can be: TJ Tseng_Jye, or PH Phonetic.
-t <i>Target</i>	Specifies the name of the converted font file.
-v <i>CodePage</i>	Specifies the type of code page to convert to. <i>CodePage</i> can be: SOPS PS55 , or ET .

Exit Status

This command returns the following exit values:

Ite m	Description
0	Successful completion.
>0	An error occurred.

Security

Access Control: You must have root authority to run this command.

Auditing Events: N/A

Examples

To convert the dictionary USRFONT.C12 to an operating system dictionary of code page of type SOPS and dictionary type of Tseng_Jye with the name aix, enter:

```
twconvdict -i TJ -v SOPS -f USRFONT.C12 -t aix
```

Files

Item	Description
/usr/lpp/tls/bin/twconvdict	Contains the twconvdict command.

twconvfont Command

Purpose

Converts other font files to a BDF font file.

Syntax

twconvfont [**-v** *CodePage*] [**-f** *Source*] [**-t** *Target*]

Description

The twconvfont command converts one font file type to the BDF font file. The supported code pages are SOPS, PS55 and ET.

Flags

Item	Description
-f <i>Source</i>	Specifies the name of font file to convert.
-t <i>Target</i>	Specifies the name of the converted font file.
-v <i>CodePage</i>	Specifies the type of code page to convert to. <i>CodePage</i> can be: SOPS PS55 , or ET .

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Security

Access Control: You must have root authority to run this command.

Auditing Events: N/A

Examples

To convert the font file USRFONT.C12 to a BDF font file of code page of type SOPS with the name user.bdf, enter:

```
twconvfont -v SOPS -f USRFONT.C12 -t user.bdf
```

Files

Item	Description
/usr/lpp/tls/bin/twconvfont	Contains the twconvfont command.

type Command

Purpose

Writes a description of the command type.

Syntax

type *CommandName* ...

Description

The standard output of the **type** command contains information about the specified command and identifies whether this is a shell built-in command, subroutine, alias, or keyword. The **type** command indicates how the specified command would be interpreted if used. Where applicable, the **type** command displays the related path name.

Because the **type** command must know the contents of the current shell environment, it is provided as a Korn shell or POSIX shell regular built-in command. If the **type** command is called in a separate command execution environment, the command may not produce accurate results. This would be the case in the following examples:

```
nohup type writer
```

```
find . -type f | xargs type
```

Exit Status

The following exit values are returned:

Item	Description
0	Successful completion.
>0	An error occurred.

Examples

1. To learn whether the **cd** command is a base command or an alias or some other command type, enter:

```
type cd
```

The screen displays the following information:

```
cd is a shell builtin
```

2. To see the location of the **find** command, enter:

```
type find
```

The screen displays the following information:

```
find is /usr/bin/find
```

Files

Item	Description
/usr/bin/ksh	Contains the Korn shell type built-in command.

u

The following AIX commands begin with the letter *u*.

ucfgif Method

Purpose

Unloads an interface instance from the kernel.

Syntax

ucfgif [**-l** *InterfaceInstance*]

Description

The **ucfgif** method removes an interface instance from the kernel. To remove the interface instance, the **ucfgif** method does the following:

1. Unloads the interface software by calling the **/usr/sbin/ifconfig** interface detach.
2. Sets the status flag of the interface instance to **defined**.

Note: The **ucfgif** method is a programming tool and should not be executed from the command line.

Flags

Item	Description
-l <i>InterfaceInstance</i>	Specifies the interface instance to be unconfigured. If no interface name is specified, all configured interface instances are unconfigured.

Example

To remove an interface instance from the kernel, enter the method in the following format:

```
ucfgif -l tr0
```

In this example, the name of the interface instance is **tr0**.

ucfginet Method

Purpose

Unloads the Internet instance and all related interface instances from the kernel.

Syntax

ucfginet

Description

The **ucfginet** method unloads the Internet instance from the kernel. This subroutine also deletes the appropriate entries in the Address Family Domain switch table and in the Network Input Interface switch

table. The **ucfginet** method also sets the status flag of the instance to **defined**. The **ucfginet** method is called by the **rmdev** high-level command.

Note: The **ucfginet** method is a programming tool and should not be executed from the command line.

ucfgqos Method

Purpose

Unconfigures and unloads the Quality of Service (QoS) instance from the kernel.

Syntax

ucfgqos

Description

The **ucfgqos** method disables Quality of Service (QoS) for the TCP/IP protocol suite on a host. This method detaches the QoS instance from the TCP/IP instance and unloads it from the kernel.

Note: The **ucfgqos** method is a programming tool and is not intended to be invoked from the command line.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Example

To configure QoS on a host, use the following format:

```
ucfgqos
```

ucfgvds Command

Purpose

ucfgvds – Unconfigures a virtual shared disk.

Syntax

ucfgvds {-a | *vds_name* ...}

Description

The **ucfgvds** command unconfigures the specified virtual shared disks. The specified virtual shared disks must be in the stopped state to be unconfigured. This command does not change any virtual shared disk definitions. It moves virtual shared disks from the stopped state to the defined state.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit vds_mgmt
```

and select the **Unconfigure a Virtual Shared Disk** option.

Under normal circumstances, you should not issue this command. The Recoverable virtual shared disk subsystem uses this command to manage shared disks in a controlled manner. If you issue this command, the results may be unpredictable.

Flags

-a

Specifies that all virtual shared disks in the stopped state are to be unconfigured.

Parameters

vsd_name

Specifies a virtual shared disk. The disk specified must be in the stopped state. If all disks have been unconfigured, and you specify VSD0, this command will attempt to unload the device driver from the kernel.

Security

You must have root authority to run this command.

Exit Status

0

Indicates the successful completion of the command.

nonzero

Indicates that an error occurred.

Restrictions

Under normal circumstances, you should not issue this command. The Recoverable virtual shared disk subsystem uses this command to manage shared disks in a controlled manner. If you issue this command, the results may be unpredictable.

Standard Output

Current RVSD subsystem run level.

Examples

To unconfigure the virtual shared disk **vsd1vg1n1** in the stopped state, enter:

```
ucfgvsd vsd1vg1n1
```

Location

/opt/rsct/vsd/bin/ucfgvsd

uconvdef Command

Purpose

Compiles or generates a UCS-2 (Unicode) conversion table for use by the **iconv** library.

Syntax

uconvdef [**-f** *SrcFile*] [**-v**] *UconvTable*

Description

The **uconvdef** command reads *SrcFile* and creates a compiled conversion table in *UconvTable*. The *SrcFile* defines a mapping between UCS-2 and multibyte code sets (one or more bytes per character). The *UconvTable* is in a format that can be loaded by the UCSTBL conversion method located in the **/usr/lib/nls/loc/uconv** directory. This method uses the table to support UCS-2 conversions in both directions.

Flags

Item	Description
-f <i>SrcFile</i>	Specifies the conversion table source file. If this flag is not used, standard input is read.
-v	Causes output of the processed file statements.
<i>UconvTable</i>	Specifies the path name of the compiled table created by the uconvdef command. This should be the name of the code set that defines conversions into and out of UCS-2.

Exit Status

The following exit values are returned:

Item	Description
0	Successful completion.
>0	An error occurred.

Examples

To access the compiled UCS-2 conversion table:

1. Create the compiled *UconvTable* using the name of the multibyte code set. For example, the conversion table between IBM-850 and UCS-2 can be compiled as follows:

```
uconvdef -f IBM-850.ucmap IBM-850
```

2. Place the table in a directory called **uconvTable**. The default system directory is **/usr/lib/nls/loc/uconvTable**. If another directory is used, the **LOCPATH** environment variable needs to be set to include the parent directory (for example, **/usr/lib/nls/loc**).

```
mv IBM-850 /usr/lib/nls/loc/uconvTable
```

3. Create symbolic links for conversions in each direction in a directory called **iconv**. The names for these links should be formed by concatenating the "From" code set and the "To" code set, separated by an underscore. The links should be set to point to the **/usr/lib/nls/loc/uconv/UCSTBL** conversion method. The default directory for these links is **/usr/lib/nls/loc/iconv**. If another directory is used, the **LOCPATH** environment variable needs to be set to include the parent directory (for example, **/usr/lib/nls/loc**).

```
ln -s /usr/lib/nls/loc/uconv/UCSTBL \  
/usr/lib/nls/loc/iconv/IBM-850_UCS-2
```

```
ln -s /usr/lib/nls/loc/uconv/UCSTBL \  
/usr/lib/nls/loc/iconv/UCS-2_IBM-850
```

Note: The \ (backslash) is a line continuation character that is only needed if the command is broken into two lines.

undefif Method

Purpose

Removes an interface object from the system configuration database.

Syntax

undefif [**-l** *InterfaceInstance*]

Description

The **undefif** method deletes the specified interface instance from the system configuration database by:

1. Removing the database object associated with the interface instance.
2. Removing the connection and attribute information associated with the interface instance.

Flags

Item	Description
-l <i>InterfaceInstance</i>	Specifies the interface instance to be undefined. If no interface instances are specified, the undefif method undefines all defined interface instances.

Example

To remove an interface instance from the database, enter a method similar to the following:

```
undefif -l tr0
```

In this example, the interface instance to be removed is `tr0`.

undefinet Method

Purpose

Undefines the Internet instance in the configuration database.

Syntax

undefinet

Description

The **undefinet** method removes the database information associated with the Internet instance, including attribute information associated with the Internet instance.

Note: The **undefinet** method is a programming tool and should not be executed from the command line.

udfcheck Command

Purpose

Performs a file system check on a UDF file system.

Syntax

udfcheck **-d** *device* [**-t** *tempfile*]

Description

The **udfcheck** command checks and repairs the UDF volume on a specified device.

Flags

Item	Description
-d <i>device</i>	Specifies the device on which udfcheck checks and repairs the UDF volume.
-t <i>tempfile</i>	Specifies a file where the udfcheck command stores information needed to perform a file system check.

Examples

1. To check the content of the UDF file system on device **/dev/cd1**, enter the following:

```
udfcheck -d /dev/cd1
```

Files

Item	Description
/usr/sbin/udfcheck	Contains the udfcheck command
/usr/lib/libudf.a	Contains the library routines called by the udfcheck command

udfcreate Command

Purpose

Creates the user defined functions (UDF) file systems.

Syntax

udfcreate **-d** *device* [**-b** *bitmap_location*] [**-f** *formatType*]

Description

The **udfcreate** command creates a UDF file system on the specified device and labels it with the generic set ID (*setID*) and volume name (*volName*).

Flags

Item	Description
-b <i>bitmap_location</i>	Specifies the location of the bitmap. It can be one of the following, b , e , or m . b indicates that the bitmap will be placed at the beginning of the partition. e indicates that the bitmap will be placed at the end of the partition. m indicates that the bitmap will be placed at the middle of the partition. The default location of the bitmap is the beginning of the partition.
-d <i>device</i>	Specifies the device on which to create the UDF volume.
-f <i>formatType</i>	Indicates the version of UDF to be present on the media. The format type of 1 represents UDF 1.5 version, 2 represents UDF 2.0 version, and 3 represents UDF 2.01 version. The default version is UDF 1.5.
-s 2048	Forces the newly created UDF filesystem to use 2048 byte logical blocks.

Examples

1. To create a new UDF file system on device **/dev/cd1**, enter the following command:

```
udfcreate -d /dev/cd1
```

Files

Item	Description
/usr/sbin/udfcreate	Contains the udfcreate command
/usr/lib/libudf.a	Contains the library routines called by the udfcreate command

udflabel Command

Purpose

Fetches and changes the label on a UDF file system.

Syntax

udflabel **-d** *device* [**-l** *label*]

Description

The **udflabel** command displays or changes a UDF volume name. If there is no label provided, it displays the current UDF volume name on the device. If there is a label provided, it sets the current UDF volume name on the device to the new label.

Flags

Item	Description
-d <i>device</i>	Specifies the device containing the UDF volume.

Item	Description
-l <i>label</i>	Sets the label on the current UDF volume.

Examples

1. To change the current label on device **/dev/cd1** to hello, enter the following command:

```
udflabel -d /dev/cd1 -l hello
```

2. To display the current label on device **/dev/cd1**, enter the following command:

```
udflabel -d /dev/cd1
```

Files

Item	Description
/usr/sbin/udflabel	Contains the udflabel command
/usr/lib/libudf.a	Contains the library routines called by the udflabel command

>lugrep Command

Purpose

Searches for a Unicode pattern in a file.

Syntax

```
ugrep unicode_hex_notation_pattern [ -i loose_match_unicode_hex_notation_pattern ] File..
```

Description

The **ugrep** command searches an input file for characters that match the specified hexadecimal representation of the Unicode-defined code point of a character.

The regular expression writer might not use a Unicode character set to specify the pattern that needs to be searched. Also, it might not be possible to input Unicode-defined code points for every character of the major written languages by using a keyboard. Therefore, the Unicode pattern that is specified to search must be the hexadecimal representation of the Unicode-defined code point of a character.

Note: The function of the **ugrep** command is the same as the function of the **grep** command with the **-U** flag.

Flags

Table 27. Flags

Item	Description
<i>unicode_hex_notation_pattern</i>	Specifies the hexadecimal representation of the Unicode-defined code point of a character. For example, to represent the □ character, whose Unicode-defined code point is U+1D11E, the value of the <i>unicode_hex_notation_pattern</i> pattern can be the hexadecimal representation as \U0001D11E, \x{1D11E}, or \u{1D11E}.

Table 27. Flags (continued)

Item	Description
-i <i>unicode_hex_notation_pattern</i>	Specifies that the search is based on a loose match of the specified Unicode hex notation pattern. Most of the regular expression engines offer case-insensitive matching as the only loose matching. If the expression engine offers case-insensitive matching as the only loose matching, then the expression engine must account for the large range of cased Unicode characters outside of the ASCII characters.

Exit Status

This command returns the following exit values:

Table 28. Exit status

Ite m	Description
0	A match was found.
1	No match was found.
>1	A syntax error was found or a file was inaccessible (even if matches were found).

Examples

1. To search the `regex_test.txt` file for the character 我, whose Unicode-defined code point is U+6211 and the hexadecimal representation is `\u6211`, enter the following command:

```
ugrep "\u6211" regex_test.txt
```

To search multiple characters, you can add a list of hexadecimal representations of the Unicode-defined code points without any space. For example, to search the characters ऋ and ॠ in the `regex_test.txt` file, enter the following command:

```
ugrep "\u0918\u0930" regex_test.txt
```

2. To specify a range of characters between the code points U+6200 and U+6300 to search in the `regex_test.txt` file, enter the following command:

```
ugrep "[\u6200-\u6300]" regex_test.txt
```

To specify a range of characters between the code points U+6200 and U+6300 that are also uppercase to search in the `regex_test.txt` file, enter the following command:

```
ugrep "[\u0000-\u0010FFFF- -\p{Lu}]" regex_test.txt
```

3. To do a loose match search of the character □, whose Unicode-defined code point is U+10425 and the hexadecimal representation is `\U00010425`, enter the following command:

```
ugrep -i "\U00010425" regex_test.txt
```

4. To search the `regex_test.txt` file for a number with decimal digits, enter the following command:

```
ugrep "\p{Nd}" regex_test.txt
```

where `Nd` is a Unicode character property for numbers with decimal digits.

5. To search the `regex_test.txt` file for Hiragana characters in the Japanese language, enter the following code:

```
ugrep "\p{Hiragana}" regex_test.txt
```

6. To search the `regex_test.txt` file for uppercase letters, lowercase letters, or numbers by using Unicode properties, enter the following commands:

- `ugrep "\p{Ll}" regex_test.txt`

where the property `Ll` matches lowercase letters in Unicode and includes lowercase letters from all languages.

- `ugrep "\p{Lu}" regex_test.txt`

where the property `Lu` matches uppercase letters in Unicode and includes uppercase letters from all languages.

- `ugrep "\p{L}" regex_test.txt`

Or

```
ugrep "\p{letter}" regex_test.txt
```

where the properties `L` and `letter` matches all letters in Unicode. The search by using the `Lu` property includes uppercase letters, lowercase letters, and connector characters. However, the search by using the `letter` property includes only the uppercase and lowercase letters.

- `ugrep "[\p{L}|\p{Nd}]" regex_test.txt`

where the property `Nd` matches numeric digits in Unicode and includes numeric digits from all languages.

7. To search for characters of the Latin language by using the `script` property, enter the following command:

```
ugrep "\p{script=Latin}" regex_test.txt
```

You can search for characters in any language by setting the value of the `script` property to the specific language.

Files

Item	Description
<code>/usr/bin/ugrep</code>	Contains the ugrep command.



uil Command

Purpose

Starts the User Interface Language (UIL) compiler for the AIXwindows system.

Syntax

```
uil [ -IPathName ] InputFile [ -m ] [ -o FileName ] [ -s ] [ -v FileName ] [ -w ] [ -wmd FileName ]
```

Description

The **uil** command calls the UIL compiler. The UIL is a specification language for describing the initial state of a user interface for an AIXwindows application. The specification describes the objects (menus, dialog boxes, labels, push buttons, and so on) used in the interface and specifies the functions to be called when the interface changes state as a result of user interaction.

Flags

Item	Description
-I <i>PathName</i>	Specifies I nclude <i>PathName</i> with no spaces. Causes the compiler to look for include files in a specified directory if include files were not found in the default paths. (uppercase i)
-m	Specifies that machine code is listed. This directs the compiler to place a description of the records that it added to the User Interface Definition (UID) in the listing file. This helps you isolate errors. The default is no machine code.
-o <i>FileName</i>	Directs the compiler to produce a UID. By default, UIL creates a UID with the name a.uid . The file specifies the file name for the UID. No UID is produced if the compiler issues any diagnostics categorized as error or severe.
-s	Directs the compiler to set the locale before compiling any files. The locale is set in an implementation-dependent manner. On ANSI C-based systems, the locale is usually set by calling the setlocale (LC_ALL, "") function. If this option is not specified, the compiler does not set the locale.
-v <i>FileName</i>	Directs the compiler to generate a listing. The file specifies the file name for the listing. If the -v option is not present, no listing is generated by the compiler. The default is no listing.
-w	Specifies that the compiler suppress all warning and informational messages. If this option is not present, all messages are generated, regardless of the severity.
-wmd <i>FileName</i>	Specifies a binary widget meta-language (WML) description file to be used instead of the default WML description.

Example

To start the UIL compiler, enter:

```
uil -I. -o ex.uid ex.uil
```

Exit Status

This command returns the following exit values:

Item	Description
0	Indicates successful completion.
>0	Indicates an error occurred.

uimx Command

Purpose

Starts the UIM/X user-interface management system for the X Window System.

Syntax

uimx [**-dir** *Path*] [**-file** *FileName*] [**-workspace** *Name*] [**-xrm** *Options*]

Description

The **uimx** command starts the UIM/X user-interface management system for the X Window System. It supports Motif 1.2 and provides a complete programming environment for developing graphical user interfaces (GUIs). UIM/X supports object-oriented programming in both C and C++.

UIM/X saves and loads text files that use the Xt resource syntax to describe interfaces and projects. It can also load UIL files. It generates C, C++, and UIL code. It can also generate makefiles, message catalogs, and resource files for an application.

UIM/X includes a built-in C Interpreter and the following tools and editors:

- Palette of Motif widgets
- Widget Browser for browsing complex widget hierarchies
- WYSIWYG layout editor for drawing interfaces
- Property Editor for setting initial values of widget properties; initial values can be literal values or C expressions
- Callback Editors for entering callback code
- Event, Action, and Translation Editors
- Menu and Main Window Editors
- Declarations Editor for editing the generated code for an interface
- Program Layout Editor for editing the generated main program and makefile; this editor gives you direct access to the main event loop

UIM/X supports two operating modes: Design and Test. In Test mode, the built-in C Interpreter allows you to test the behavior of your application. In Design mode, the C Interpreter validates the code you enter into the various UIM/X editors.

UIM/X provides a convenience library of functions that simplify the task of programming with X and Motif.

Flags

Item	Description
dir <i>Path</i>	Sets UIM/X's current directory to path.
file <i>FileName</i>	Loads an existing project, interface, or palette file called <i>FileName</i> . <i>FileName</i> can include an absolute path name, a path name relative to the current directory, or a path name relative to the -dir value.
workspace <i>Name</i>	Loads UIM/X into the corresponding CDE workspace called <i>name</i> .
xrm <i>Options</i>	Enables you to enter any resource specifications (<i>options</i>) that you would otherwise put in a resource file.

Security

Access Control: Any User

Files Accessed: None

Example

To start UIM/X, enter:

```
uimx
```


Files

Item	Description
<code>/usr/uimx2.8/bin/uimx</code>	Contains the uimx command.

ul Command

Purpose

Performs underlining.

Syntax

ul [**-i**] [**-t** *Terminal*] [*File ...*]

Description

The **ul** command reads the named files specified by the *File* parameter (or standard input if no file is given) and translates occurrences of underscores to the sequence that indicates underlining for the terminal in use, as specified by the **TERM** environment variable.

Flags

Item	Description
-i	Causes the ul command to indicate underlining by a separate line containing appropriate <code>_</code> (underline characters). Use this to see the underlining present in an nroff command output stream on a CRT terminal.
-t <i>Terminal</i>	Overrides the terminal type specified in the environment. The terminfo file is read to determine the appropriate sequences for underlining. If the terminal is incapable of underlining, but is capable of a standout mode, then that mode is used instead. If the terminal can overstrike or automatically underline, the ul command acts like the cat command and displays on the screen. If the terminal cannot underline and no alternatives are available, underlining is ignored. If the -t flag is not specified, the ul command translates for the terminal type specified by the TERM environment variable. If the value of the <i>Terminal</i> variable is not a valid terminal type, the ul command translates for a dumb terminal.

Files

Item	Description
<code>/usr/share/lib/terminfo/*</code>	Contains the terminal capabilities database.

ulimit Command

Purpose

Sets or reports user resource limits.

Syntax

ulimit [**-H**] [**-S**] [**-a**] [**-c**] [**-d**] [**-f**] [**-m**] [**-n**] [**-r**] [**-s**] [**-t**] [**-u**] [*Limit*]

Description

The **ulimit** command sets or reports user process resource limits, as defined in the **/etc/security/limits** file. This file contains these default limits:

```
fsize = 2097151
core = 2097151
cpu = -1
data = 262144
rss = 65536
stack = 65536
nofiles = 2000
threads = -1
nproc = -1
```

These values are used as default settings when a new user is added to the system. The values are set with the **mkuser** command when the user is added to the system, or changed with the **chuser** command.

Limits are categorized as either soft or hard. With the **ulimit** command, you can change your soft limits, up to the maximum set by the hard limits. You must have root user authority to change resource hard limits.

Many systems do not contain one or more of these limits. The limit for a specified resource is set when the *Limit* parameter is specified. The value of the *Limit* parameter can be a number in the unit specified with each resource, or the value *unlimited*. To set the specific ulimit to unlimited, use the word *unlimited*.

Note: Setting the default limits in the **/etc/security/limits** file sets system wide limits, not just limits taken on by a user when that user is created.

The current resource limit is printed when you omit the *Limit* parameter. The soft limit is printed unless you specify the **-H** flag. When you specify more than one resource, the limit name and unit is printed before the value. If no option is given, the **-f** flag is assumed.

Since the **ulimit** command affects the current shell environment, it is provided as a shell regular built-in command. If this command is called in a separate command execution environment, it does not affect the file size limit of the caller's environment. This would be the case in the following examples:

```
nohup ulimit -f 10000
env ulimit 10000
```

Once a hard limit has been decreased by a process, it cannot be increased without root privilege, even to revert to the original limit.

Flags

Ite m	Description

-a	Lists all of the current resource limits.
-c	Specifies the size of core dumps, in number of 512-byte blocks.
-d	Specifies the size of the data area, in number of K bytes.
-f	Sets the file size limit in blocks when the <i>Limit</i> parameter is used, or reports the file size limit if no parameter is specified. The -f flag is the default.
-H	Specifies that the hard limit for the given resource is set. If you have root user authority, you can increase the hard limit. Anyone can decrease it.
-m	Specifies the size of physical memory (resident set size), in number of K bytes. This limit is not enforced by the system.
-n	Specifies the limit on the number of file descriptors a process may have.
-t	Specifies the limit on the number of threads a process can have.
-s	Specifies the stack size, in number of K bytes.

Item	Description

-S	Specifies that the soft limit for the given resource is set. A soft limit can be increased up to the value of the hard limit. If neither the -H nor -S flags are specified, the limit applies to both.
-t	Specifies the number of seconds to be used by each process.
-u	Specifies the limit on the number of a process a user can create.

Exit Status

The following exit values are returned:

Item	Description

0	Successful completion.
>0	A request for a higher limit was rejected or an error occurred.

Example

To set the file size limit to 51,200 bytes, enter:

```
ulimit -f 100
```

To list all the current resource limits, enter:

```
ulimit -a
time(seconds)      unlimited
file(blocks)       2097151
data(kbytes)       131072
stack(kbytes)      32768
memory(kbytes)     65536
coredump(blocks)   2097151
nofiles(descriptors) 2000
threads(per process) unlimited
processes(per user) unlimited
```

Files

Item	Description
/usr/bin/ksh	Contains the ulimit built-in command.

umask Command

Purpose

Displays or sets the file mode creation mask.

Syntax

umask [**-S**] [*Mask*]

Description

If the *Mask* parameter is not specified, the **umask** command displays to standard output the file mode creation mask of the current shell environment. If you specify the *Mask* parameter using a three-digit octal number or symbolic code, the **umask** command sets the file creation mask of the current shell

execution environment. The bits set in the file creation mask are used to clear the corresponding bits requested by an application or command when creating a file.

The **chmod** command describes how to use the symbolic and numeric codes to set permissions.

The **-S** flag produces symbolic output. If the flag is not specified, the default output format is octal.

If the **/usr/bin/umask** command is called in a subshell or separate command execution environment, it does not affect the file mode creation mask of the caller's environment. This would be the case in the following example:

```
(umask 002)
```

```
nohup umask ...
```

```
find . -exec umask ... \;
```

Flags

Item	Description

-S	Produces symbolic output.

Exit Status

The following exit values are returned:

Item	Description

0	The file mode creation mask was successfully changed, or no <i>Mask</i> parameter was supplied.
>0	An error occurred.

Examples

1. To set the mode mask so that subsequently created files have their **S_IWOTH** bit cleared, enter either:

```
umask a=rx,ug+w
```

OR

```
umask 002
```

After setting the mode mask, display the current values of the mode mask by entering:

```
umask
```

The screen displays the following value:

```
02
```

2. To produce symbolic output, enter:

```
umask -S
```

The screen displays the following values:

```
u=rx,g=rx,o=rx
```

3. Either numeric or symbol output can be used as the *Mask* parameter to a subsequent invocation of the **umask** command. Assume the mode mask is set as shown in example 2. To set the mode mask so that subsequently created files have their **S_IWGRP** and **S_IWOTH** bits cleared, enter:

```
umask g-w
```

4. To set the mode mask so that subsequently created files have all their write bits cleared, enter:

```
umask -- -w
```

Note: The **-r**, **-w**, and **-x** *Mask* parameter values (or anything beginning with a hyphen) must be preceded by **--** (double hyphen, no space between) to keep it from being interpreted as an option.

Files

Item	Description
/usr/bin/ksh	Contains the Korn shell umask built-in command.
/usr/bin/umask	Contains the umask command.

umcode_latest Command

Purpose

Identifies system resources with firmware or microcode that can be updated from a specified source of image files.

Syntax

```
umcode_latest [-s source] [-l] [-A] | [-a [-q][-r] | -i] | -h
```

Description

The `umcode_latest` command lists or downloads the system resources that have an older firmware or microcode level than the firmware or microcode level that was found on the specified source for those system resources.

Note: System Firmware images of system types 8842/8844/7047/7013/7015/7017 and 7025-F50 are not supported by this command. For systems with temporary and permanent system firmware images, the `umcode_latest` command uses the temporary system firmware image for comparisons with the images on the specified source. System firmware image file names must end with `.img`.

Flags

Item	Description
-a	Updates all system resources that have newer microcode on the source.
-A	Lists or updates resource when any of the images on the source is different from the image currently listed or updated. The default is to list or update whenever the source has a newer image.
-h	Provides extended usage help.
-i	Provides an interactive mode so that each resource that needs an update is prompted.

Item	Description
-l	Lists the system resources that need updates. This is the default.
-q	Refrains from asking whether to proceed with the update all.
-i	Refrains from asking whether to proceed with the update requiring a system IPL.
-s <i>source</i>	Points to the source of the microcode image. The default is /etc/microcode.

Exit Status

Item	Description
0	The command completed successfully.
>0	An error occurred.

Examples

1. To list all system resources with firmware or microcode that can be updated from the images in /etc/microcode, enter:

```
/usr/lpp/diagnostics/bin/umcode_latest
```

2. To list all system resources with firmware or microcode that can be updated from the images that are in the /tmp/fwupdate directory, enter:

```
/usr/lpp/diagnostics/bin/umcode_latest -s /tmp/fwupdate
```

3. To list all system resources with firmware or microcode that can be updated from the images that are in the /tmp/fwupdate directory, and for each resource ask whether the resource should be updated at this time, enter:

```
/usr/lpp/diagnostics/bin/umcode_latest -s /fwupdate -i
```

4. To automatically update all of the system resources with firmware or microcode that have newer images on the ISO 9660 format CD-ROM, which has already been inserted into the cd1 drive, enter:

```
/usr/lpp/diagnostics/bin/umcode_latest -s cd1 -a -q
```

Restrictions

System Firmware images of system types 8842/8844/7047/7013/7015/7017 and 7025-F50 are not supported by this command. For systems with temporary and permanent system firmware images, the umcode_latest command uses the temporary system firmware image for comparisons with the images on the specified source. System firmware image file names must end with .img.

Location

/usr/lpp/diagnostics/bin/umcode_latest

umount or unmount Command

Purpose

Unmounts a previously mounted file system, directory, or file.

Syntax

{ **umount** | **umount** } [**-f**] [**-a**] | [**all** | **allr** | *Device* | *Directory* | *File* | *FileSystem* | **-n** *Node* | **-t** *Type*]

Description

Another name for the **umount** command is the **unmount** command. Either name can be used. You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit umount
```

The **umount** command unmounts a previously mounted device, directory, file, or file system. Processing on the file system, directory, or file completes and it is unmounted. Members of the system group and users operating with root user authority can issue any **umount** command. Only users with root authority or are members of the system group can unmount a directory or file.

Note: SMIT will not unmount the **/usr/lpp/info/\$LANG** directory, the directory on which SMIT helps are located. Typically, this is the CD-ROM.

To unmount local mounts you can specify the device, directory, file, or file system on which it is mounted.

If the file system being unmounted is a JFS2 snapshot, the **umount** command will unmount the snapshot, though the snapshot will still be active. The **snapshot** command must be used to delete the snapshot.

If the file system being unmounted is a snapped file system with mounted snapshots, the **umount** command displays a warning that there are mounted snapshots and exits without unmounting the file system. The snapshots must be unmounted first.

Note: If the **cdromd** CD and DVD automount daemon is enabled, then those devices will be automatically mounted as specified in the **/etc/cdromd.conf** file. Use the **cdumount** or **cdeject** command to unmount an automounted CD or DVD. Use "**stopsrc -s cdromd**" to disable the CD/DVD automount daemon.

Flags

Item	Description
-a	Unmounts all mounted file systems.
all	Unmounts all mounted file systems.
allr	Unmounts all remotely mounted file systems.

Note: For remote mounts, specify the device, directory, file, or file system parameters. If you specify the **allr** flag, the **umount** command unmounts all remote mounts.

Item	Description
-f	For remote mounted file systems, the -f flag forces an unmount to free a client when the server is down and server path names cannot be resolved, or when a file system must be unmounted while it is still in use. Note: For remote file systems, using this flag causes all file operations on the file system except close() and unmap() to fail. Any file data that has been written by an application but has not yet transferred to the server will be lost. A forced unmount of an NFS version 4 file system can cause open file state for other file systems mounted from the same server to be lost as well. For local JFS2 file systems, the -f flag forces an unmount when a file system must be unmounted while it is still in use. Note: You can use the -f flag only in JFS2 file systems, not in other journaled file systems. The following restrictions are applied on a forced unmount of a JFS2 file system: • The -f flag cannot force an unmount of a file system if a subdirectory or a file is overmounted on the file system. • The -f flag cannot force an unmount of a file system with mounted or open external snapshots until those snapshots are forced unmounted.
-n Node	Specifies the node holding the mounted directory you want to unmount. The umount -n Node command unmounts all remote mounts made from the <i>Node</i> parameter.
-t Type	Unmounts all stanzas in the /etc/filesystems file that contain the type=Type flag and are mounted. The <i>Type</i> parameter is a string value, such as the remote value that specifies the name of the group.

Note: You cannot use the **umount** command on a device in use. A device is in use if any file is open for any reason or if a user's current directory is on that device.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To unmount all mounts from remote node Node A, enter:

```
umount -n nodeA
```

2. To unmount files and directories of a specific type, enter:

```
umount -t test
```

This unmounts all files or directories that have a stanza in the **/etc/filesystems** file that contains the **type=test** attribute.

Files

Item	Description
/etc/filesystems	Lists the known file systems and defines their characteristics.

umountall Command

Purpose

Unmounts groups of dismountable devices or filesystems.

Syntax

umountall [**-k**] [**-s**] [**-F** *FileSystemType*] [**-l** | **-r**]

umountall [**-k**] [**-s**] [**-h** *Host*]

Description

The **umountall** command by default unmounts all dismountable file systems or devices except root, /proc, /var and /usr. If the *FileSystemType* is specified, **umountall** limits its actions to the file system type specified. There is no guarantee that **umountall** will unmount busy file systems, even if the **-k** option is specified.

Flags

Item	Description
-F <i>FileSystemType</i>	Specifies the type of file systems to be dismounted. <i>FileSystemType</i> corresponds to the vfs column printed out by the mount command. All dismountable file systems of the given type will be unmounted. This flag cannot be used in combination with the -h flag.
-h <i>Host</i>	Specifies the host node. All file systems remotely mounted from this host will be unmounted.
-k	Sends a SIGKILL to each process on the mount point before unmounting. This option internally uses the fuser -k command to kill all the processes running on the mount point. As this option causes each process on the mount point to be killed, the unmount of the mount point does not happen immediately. There is no guarantee that umountall will unmount busy file systems, even if the -k option is specified. An attempt to unmount the mount point will be made only after all the processes using the mount point are killed.
-l	Limits the action to local filesystems.
-r	Limits the action to remote filesystems.
-s	This is a no-operation flag provided for System V compatibility on serializing the unmounts . The serialization of the unmount command is done using -k option by terminating all the associated processes on the mount point.

Exit Status

- 0
The command completed successfully.

>0

An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To unmount all dismountable file systems, enter:

```
umountall
```

2. To unmount all dismountable filesystems of type **jfs**, enter:

```
umountall -F jfs
```

3. To unmount all dismountable filesystems mounted from host.domain, enter:

```
umountall -h host.domain
```

4. To unmount all remotely mounted filesystems, enter:

```
umountall -r
```

Files

Item	Description
/usr/sbin/umountall	Contains the umountall command.

unalias Command

Purpose

Removes alias definitions.

Syntax

unalias -a

unalias *AliasName* ...

Description

The **unalias** command removes the definition for each alias name specified, or removes all alias definitions if the **-a** flag is used. Alias definitions are removed from the current shell environment.

Since the **unalias** command affects the current shell execution environment, it is provided as a Korn shell or POSIX shell regular built-in command.

Flags

Item	Description

-a	Removes all alias definitions from the current shell environment.

Exit Status

The following exit values are returned:

Item	Description
0	Successful completion.
>0	One of the alias names specified did not represent a valid alias definition, or an error occurred.

Files

Item	Description
/usr/bin/ksh	Contains the Korn shell unalias built-in command.
/usr/bin/unalias	Contains the unalias command.

uname Command

Purpose

Displays the name of the current operating system.

Syntax

```
uname [ -a | -x | -S Name ] [ -F ] [ -f ] [ -l ] [ -L ] [ -m ] [ -M ] [ -n ] [ -p ] [ -r ] [ -s | V ] [ -T Name ] [ -u ] [ -v ] [ -W ]
```

Description

The **uname** command writes to standard output the name of the operating system that you are using.

The machine ID number contains 12 characters in the following digit format: *xyyyyyymmss*. The *xx* positions indicate the system and is always 00. The *yyyyyy* positions contain the unique ID number for the entire system. The *mm* position represents the model ID. The *ss* position is the submodel number and is always 00. The model ID describes the ID of the CPU Planar, not the model of the System as a whole.

Most machines share a common model ID of 4C.

The machine identifier value returned by the **uname** command may change when new operating system software levels are installed. This change affects applications using this value to access licensed programs. To view this identifier, enter the **uname -m** command.

Contact the appropriate support organization if your application is affected.

Flags

Item	Description
-a	Displays all information specified with the -m , -n , -r , -s , and -v flags. Cannot be used with the -x or -SName flag. If the -x flag is specified with the -a flag, the -x flag overrides it.
-F	Displays a system identification string comprised of hexadecimal characters. This identification string is the same for all partitions on a particular system.
-f	Similar to the F flag, except that the partition number is also used in the calculation of this string. The resulting identification string is unique for each partition on a particular system.
-l	Displays the LAN network number.

Item	Description
-L	Displays LPAR number and LPAR name. If LPAR does not exist, -1 is displayed for LPAR number and NULL for LPAR name. If a system is capable of LPAR, but is currently running in Symmetric Multi Processing (SMP) mode, 1 is displayed for LPAR number and NULL for LPAR name.
-m	Displays the machine ID number of the hardware running the system. Note: The -m flag cannot be used to generate a unique machine identifier for partitions in an LPAR environment.
-M	Displays the system model name. If the model name attribute does not exist, a null string is displayed.
-n	Displays the name of the node. This may be a name the system is known by to a UUCP communications network.
-p	Displays the architecture of the system processor.
-r	Displays the release number of the operating system.
-s	Displays the system name. This flag is on by default. The -s option is mutually exclusive with the -V option.
-V	Displays the VIOS Complete version detail if ran in a LPAR that contains VIOS, else displays details of the AIX operating system. The -V option is mutually exclusive with the -s option.
-S Name	Sets the name of the node. This can be the UUCP communications network name for the system.
-T Name	Sets the system name. This can be the UUCP communications network name for the system.
-u	Displays the system ID number. If this attribute is not defined, the output is the same as the output displayed by uname -m .
-v	Displays the operating system version.
-W	Displays the static workload partition identification number. If the uname command runs in the Global environment, a value of zero is displayed.
-x	Displays the information specified with the -a flag as well as the LAN network number, as specified by the -l flag.

If you enter a flag that is not valid, the **uname** command exits with an error message, an error return status, and no output.

Note: The **uname** command does not preserve the new system name and node name values across system reboot.

Exit Status

This command returns the following exit values:

Item	Description
0	The requested information was successfully written.
>0	An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command](#)

Database in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Example

To display the complete system name and version banner, enter:

```
uname -a
```

Files

Item	Description
/usr/bin/uname	Contains the uname command.

uncompress Command

Purpose

Restores compressed files.

Syntax

```
uncompress [ -c ] [ -F ] [ -f ] [ -n ] [ -q ] [ -V ] [ File ... ]
```

Description

The **uncompress** command restores original files that were compressed by the **compress** command. Each compressed file specified by the *File* parameter is removed and replaced by an expanded copy. The expanded file has the same name as the compressed version, but without the **.Z** extension. If the user has root authority, the expanded file retains the same owner, group, modes, and modification time as the original file. If the user does not have root authority, the file retains the same modes and modification time, but acquires a new owner and group. If no files are specified, standard input is expanded to standard output.

Flags

Item	Description
-c	Write to standard output. No files are changed.
-f or -F	Forces expansion. The -f and -F flags are interchangeable. Overwrites the file if it already exists. The system does not prompt the user that an existing file will be overwritten. File size may not actually shrink.
-n	Omits the compressed file header from the compressed file. Note: Use this option if the file was compressed using the -n flag. Otherwise, uncompressing the file will not work.
-q	Suppresses the display of compression statistics generated by the -v flag. If several -v and -q flags are on the same command line, the last one specified controls the display of the statistics.
-V	Writes the current version and compile options to standard error.

Parameters

Item	Description
<i>File ...</i>	Specifies the compressed files to restore.

Return Values

The **uncompress** command detects an error and exit with a status of 1 if any of the following events occur:

- The input file was not produced by the **compress** command.
- An input file cannot be read or an output file cannot be written.

If no error occurs, the exit status is 0.

Exit Status

Item	Description
0	Successful completion.
>0	An error occurred.

Example

To uncompress the `foo.Z` file, enter:

```
uncompress foo.Z
```

The `foo.Z` file is uncompressed and renamed `foo`.

undefvsd Command

Purpose

undefvsd – Undefined a virtual shared disk.

Syntax

undefvsd *vsd_name* ...

Description

This command is used to remove a virtual shared disk definition and any special device files from **/dev** for the given *vsd_names* on all the virtual shared disk nodes. The virtual shared disks must be unconfigured and in the defined state on all the virtual shared disk nodes.

You can use the System Management Interface Tool (SMIT) to run the **undefvsd** command. To use SMIT, enter:

```
smit delete_vsd
```

and select the **Undefine a Virtual Shared Disk** option.

Flags

None.

Parameters

vsd_name

Specifies the virtual shared disk whose underlying logical volume you no longer want to be globally accessed by any virtual shared disk nodes.

Security

You must have root authority to run this command.

Exit Status

0

Indicates the successful completion of the command.

nonzero

Indicates that an error occurred.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **startdomain** command. To bring a particular node online in an existing peer domain, use the **startnode** command. For more information on creating and administering an RSCT peer domain, refer to the *RSCT: Administration Guide*.

Standard Output

Current RVSD subsystem run level.

Examples

To delete the information associated with the virtual shared disk **vsd1vg2n1**, enter:

```
undefvsd vsd1vg2n1
```

Location

/usr/lpp/vsd/bin/undefvsd

unexpand Command

Purpose

Writes to standard output with tabs restored.

Syntax

unexpand [**-a** | **-t** *TabList*] [*File ...*]

Description

The **unexpand** command puts tabs back into the data from the standard input or the named files and writes the result to standard output. By default, only leading spaces and tabs are reconverted to maximal strings of tabs.

Note: The *File* parameter must be a text file.

Flags

Item	Description
-a	Inserts tabs wherever their presence compresses the resultant file by replacing two or more characters.
-t <i>TabList</i>	Specifies the position of the tab stops. The default value of a tab stop is 8 column positions. The <i>TabList</i> variable must consist of a single positive-decimal integer or multiple positive-decimal integers. The multiple integers must be in ascending order and must be separated by commas or by blank characters with quotation marks around the integers. The single <i>TabList</i> variable sets the tab stops an equal number of column positions apart. The multiple <i>TabList</i> variable sets the tab stop at column positions that correspond to the integers in the <i>TabList</i> variable. A space-to-tab conversion does not occur for characters at positions beyond the last one specified in a multiple <i>TabList</i> variable.

Note: When the **-t** flag is specified, the **-a** flag is ignored and conversion is not limited to processing leading blank characters.

Exit Status

This command returns the following exit values:

Item	Description
0	The command ran successfully.
>0	An error occurred.

Example

To replace space characters with tab characters in the **xyz** file, enter:

```
unexpand xyz
```

Files

Item	Description
/usr/bin/unexpand	Contains the unexpand command.

unfencevsd Command

Purpose

unfencevsd – Gives applications running on a node or group of nodes access to a virtual shared disk or group of virtual shared disks that were previously fenced from applications running on those nodes.

Syntax

```
unfencevsd {-a | -v vsd_name_list} {-n node_list [-f]}
```


Description

Under some circumstances, the system may believe a node has become inoperable and may begin recovery procedures when the node is actually operational, but is cut off from communication with other nodes running the same application. In this case, the problem node must not be allowed to serve requests for the virtual shared disks it normally manages until recovery is complete and the other nodes running the application recognize the problem node as operational. The `fencevsd` command prevents the problem node from filling requests for its virtual shared disks. The `unfencevsd` command allows fenced nodes to regain access to the virtual shared disks.

You can issue this command from any node that is online in the peer domain.

Flags

-a

Specifies all virtual shared disks.

-f

Allows a fenced node to unfence itself.

-n *node_list*

Specifies one or more node numbers separated by commas.

-v *vsd_name_list*

Specifies one or more virtual shared disk names, separated by commas.

Parameters

None.

Security

You must have root authority to run this command.

Exit Status

0

Indicates the successful completion of the command.

nonzero

Indicates that an error occurred.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **starttrpdomain** command. To bring a particular node online in an existing peer domain, use the **starttrpnode** command. For more information on creating and administering an RSCT peer domain, refer to the *RSCT: Administration Guide*.

Standard Output

Current RVSD subsystem run level.

Examples

1. To unfence node 5 from the virtual shared disks vsd1 and vsd2, enter:

```
unfencevsd -v vsd1,vsd2 -n 5
```

2. To unfence node 7 from the virtual shared disks vsd1 and vsd2 when the unfencevsd command must be entered from node 7, enter:

```
unfencevsd -v vsd1,vsd2 -n 7 -f
```

Location

/opt/rsct/vsd/bin/unfencevsd

unget Command (SCCS)

Purpose

Cancels a previous **get** command.

Syntax

unget [-rSID] [-s] [-n] *File* ...

Description

The **unget** command allows you to restore a g-file created with **get -e** before the new delta is created. Any changes are therefore discarded. If you specify a - (dash) for the value of *File*, standard input is read, and each line of standard input is interpreted as the name of an SCCS file. An end-of-file character terminates input.

If you specify a directory for the *File* value, the **unget** command performs the requested actions on all SCCS files that are currently in the process of being edited (those files with the **s.** prefix).

Once you have run an **unget** command on a file, you must reissue a **get -e** command to make changes to the file. The **unget** command automatically deletes the g-file.

Flags

Each flag or group of flags applies independently to each named file.

Item	Description
-n	Prevents the automatic deletion of the g-file. This flag allows you to retain the edited version of the file without making a delta.
-rSID	Specifies the new delta that would have been created by the next use of the delta command. You must use this flag if you have two or more pending deltas to the file under the same login name. You can look at the p-file to see if you have more than one delta pending to a particular SID under the same login name. The <i>SID</i> specification must unambiguously specify only one SID to discard, or the unget command displays an error message and stops running.
-s	Suppresses displaying the deleted SID.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Example

To discard the changes you have made to an SCCS file after running a **get -e** command, enter:

```
unget s.prog.c
```

Files

Item	Description
<code>/usr/bin/unget</code>	Contains the path to the SCCS unget command.

unifdef Command

Purpose

Removes `ifdef` lines from a file.

Syntax

```
unifdef [ -t ] [ -l ] [ -c ] [ -DSymbol ] [ -USymbol ] [ -idSymbol ] [ -iuSymbol ] [ File ]
```

Description

The **unifdef** command is useful for removing `ifdef` lines from a file while otherwise leaving the file alone. The **unifdef** command recognizes nested `ifdefs`, comments, and single and double quotes of C syntax in order to function correctly, but does not include files or interpret macros. The **unifdef** command recognizes but does not remove comments.

The **unifdef** command takes its input from standard input if no *File* is specified and copies its output to standard output.

Once a *Symbol* is specified, the lines inside those `ifdefs` are copied to the output or removed, as appropriate. The `ifdef`, `ifndef`, `else`, `elif`, and `endif` lines associated with the symbol are also removed. `ifdefs` that involve unspecified symbols are untouched and copied out along with their associated `ifdef`, `else`, `elif`, and `endif` lines. If the same symbol appears in more than one argument, only the first occurrence is significant. For instance, if an `ifdef X` occurs nested inside another `ifdef X`, the inside `ifdef` is considered an unrecognized symbol.

When using `ifdefs` to delimit non-C lines such as comments or unfinished code, it is necessary to specify which symbols are to be used for that purpose. Otherwise, the **unifdef** command will try to parse for quotes and comments in those `ifdef` lines.

The **unifdef** command cannot process **cpp** constructs such as:

```
#if defined(X) || defined(Y)
```

OR

```
#elif X
```

OR

```
#elif defined(X) || defined(Y)
```

Keywords

The following keywords are recognized by the **unifdef** command:

- **ifdef**
- **ifndef**

- **else**
- **endif**
- **elif**

Flags

Item	Description
-c	Complements the operation of the unifdef command. That is, the lines which would have been removed are retained and vice versa.
-D Symbol	Specifies the symbol to be defined.
<i>File</i>	Specifies the input source.
-id Symbol	The unifdef command will not try to recognize comments, single quotes, or double quotes inside specified ifdefs , but these lines will be copied out.
-iu Symbol	The unifdef command will not try to recognize comments, single quotes, or double quotes inside specified ifdefs . These lines will not be copied out.
-l	Causes removed lines to be replaced with blank lines instead of being deleted.
-t	Allows the unifdef command to be used for plain text (instead of C code): the unifdef command will not try to recognize comments, single quotes and double quotes.
-U Symbol	Specifies the symbol to be undefined.

Exit Status

This command returns the following exit values:

Item	Description
0	The output is an exact copy of the input.
1	The output is not an exact copy of the input.
2	The command failed due to a premature EOF, or to an inappropriate else , elif , or endif .

Examples

1. The following example:

```
unifdef -DA original.c > modified.c
```

causes the **unifdef** command to read the file `original.c`, and remove the `#ifdef A` lines. It then removes everything following an `#elif/#else` associated with the `#ifdef A`, down to the `#endif`. The output is placed in the `modified.c` file.

2. The following example:

```
unifdef -UA original.c > modified.c
```

causes the **unifdef** command to read the file `original.c`, and remove the `#ifdef A` down to either its associated `#elif/#else`, or its associated `#endif`. In the case of the `#elif`, the `#elif` is replaced with `#if`. In the case of `#else`, the `#else` is deleted along with its associated `#endif`. The output is placed in the `modified.c` file.

Files

Item	Description
<code>/usr/bin/unifdef</code>	Contains the unifdef command.

uniq Command

Purpose

Reports or deletes repeated lines in a file.

Syntax

```
uniq [ -c | -d | -u ] [ -f Fields ] [ -s Characters ] [ -Fields ] [ +Characters ] [ InFile [ OutFile ] ]
```

Description

The **uniq** command deletes repeated lines in a file. The **uniq** command reads either standard input or a file specified by the *InFile* parameter. The command first compares adjacent lines and then removes the second and succeeding duplications of a line. Duplicated lines must be adjacent. (Before issuing the **uniq** command, use the **sort** command to make all duplicate lines adjacent.) Finally, the **uniq** command writes the resultant unique lines either to standard output or to the file specified by the *OutFile* parameter. The *InFile* and *OutFile* parameters must specify different files.

The input file must be a text file. A *text* file is a file that contains characters organized into one or more lines. The lines can neither exceed 2048 bytes in length (including any newline characters) nor contain null characters.

The **uniq** command compares entire lines by default. If the **-f** *Fields* or **-Fields** flag is specified, the **uniq** command ignores the number of fields specified by the *Fields* variable. A *field* is a string of characters separated from other character strings by one or more <blank> characters. If the **-s** *Characters* or **-Characters** flag is specified, the **uniq** command ignores the number of characters specified by the *Characters* variable. Values specified for the *Fields* and *Characters* variables must be positive decimal integers.

The current national language environment determines the <blank> characters used by the **-f** flag as well as how the **-s** flag interprets bytes as a character.

The **uniq** command exits with a value of 0 if successful. Otherwise, it exits with a value greater than 0.

Flags

Item	Description
-c	Precedes each output line with a count of the number of times each line appeared in the input file.
-d	Displays only the repeated lines.
-f <i>Fields</i>	Ignores the number of fields specified by the <i>Fields</i> variable. If the value of the <i>Fields</i> variable exceeds the number of fields on a line of input, the uniq command uses a null string for comparison. This flag is equivalent to the -Fields flag.
-u	Displays only the unrepeated lines.

Item	Description
-s Characters	Ignores the number of characters specified by the <i>Characters</i> variable. If the value of the <i>Characters</i> variable exceeds the number of characters on a line of input, the uniq command uses a null string for comparison. If both the -f and -s flags are specified, the uniq command ignores the number of characters specified by the -s Characters flag starting in the field following the fields specified by the -f Fields flag. This flag is equivalent to the +Characters flag.
-Fields	Ignores the number of fields specified by the <i>Fields</i> variable. This flag is equivalent to the -f Fields flag.
+Characters	Ignores the number of characters specified by the <i>Characters</i> variable. If both the -Fields and +Characters flags are specified, the uniq command ignores the number of characters specified by the +Characters flag starting in the field following the fields specified by the -Fields flag. This flag is equivalent to the -s Characters flag.

Exit Status

This command returns the following exit values:

Item	Description
0	The command ran successfully.
>0	An error occurred.

Example

To delete repeated lines in a file named `fruit` and save it to a file named `newfruit`, enter:

```
uniq fruit newfruit
```

If the `fruit` file contains the following lines:

```
apples
apples
peaches
pears
bananas
cherries
cherries
```

then the `newfruit` file will contain the following lines after you run the **uniq** command:

```
apples
peaches
pears
bananas
cherries
```

Files

Item	Description
/usr/bin/uniq	Contains the uniq command.

units Command

Purpose

Converts units in one measure to equivalent units in another measure.

Syntax

units [**-**] [*File*]

Description

The **units** command converts quantities expressed in one measurement to their equivalents in another. The **units** command is an interactive command. It prompts you for the unit you want to convert *from* and the unit you want to convert *to*. This command only does multiplicative scale changes. That is, it can convert from one value to another only when the conversion involves a multiplication. For example, it cannot convert between degrees Fahrenheit and degrees Celsius because the value of 32 must be added or subtracted in the conversion.

You can specify a quantity as a multiplicative combination of units, optionally preceded by a numeric multiplier.

Indicate powers by entering suffixed positive integers, and indicate division with a / (slash).

The **units** command recognizes lb as a unit of mass, but considers pound to be the British pound sterling. Compound names are run together (such as lightyear). Prefix British units differing from their American counterparts with br (brgallon, for instance).

The **/usr/share/lib/unittab** file contains a complete list of the units that the **units** command uses. You can also define new units in this file. The *File* parameter may be used to override the values of the standard conversion factors listed in the **/usr/share/lib/unittab** file. The specified file must follow the same format as the **unittab** file.

Most familiar units, abbreviations, and metric prefixes are recognized by the **units** command, as well as the following:

Item	Description
pi	Ratio of circumference to diameter
c	Speed of light
e	Charge on an electron
g	Acceleration of gravity
force	Same as g
mole	Avogadro's number
water	Pressure head per unit height of water
au	Astronomical unit

Flags

Item	Description
m	
-	Lists the conversion factors contained in the /usr/share/lib/unittab file before you are prompted to enter your conversion.

Examples

1. To display conversion factors for inches to centimeters, enter:

```
units
you have: in
you want: cm
```

The **units** command returns the following values:

```
* 2.540000e+00
/ 3.937008e-01
```

The output tells you to multiply the number of inches by 2.540000e+00 to get centimeters, and to multiply the number of centimeters by 3.937008e-01 to get inches.

These numbers are in standard exponential notation, so 3.937008e-01 means 3.937008×10^{-1} , which is the same as 0.3937008.

Note: The second number is always the reciprocal of the first; for example, 2.54 equals $1/0.3937008$.

2. To convert a measurement to different units, enter:

```
units
you have: 5 years
you want: microsec
```

The **units** command returns the following values:

```
* 1.577846e+14
/ 6.337753e-15
```

The output shows that 5 years equals 1.577846×10^{14} microseconds, and that one microsecond equals 6.337753×10^{-15} years.

3. To give fractions in measurements, enter:

```
units
you have: 1|3 mi
you want: km
```

The **units** command returns the following values:

```
* 5.364480e-01
/ 1.864114e+00
```

The | (vertical bar) indicates division, so 1|3 means one-third. This shows that one-third mile is the same as 0.536448 kilometers.

4. To include exponents in measurements, enter:

```
units
you have: 1.2-5 gal
you want: floz
```

The **units** command returns the following values:

```
* 1.536000e-03
/ 6.510417e+02
```

The expression 1.2-5 gal is the equivalent of 1.2×10^{-5} . Do *not* type an e before the exponent (that is, 1.2e-5 gal is not valid). This example shows that 1.2×10^{-5} (0.000012) gallons equal 1.536×10^{-3} (0.001536) fluid ounces.

5. To specify complex units, enter:


```
units
you have: gram centimeter/second2
you want: kg-m/sec2
```

The **units** command returns the following values:

```
* 1.000000e-05
/ 1.000000e+05
```

The units `gram centimeter/second2` mean "grams x centimeters/second2." Similarly, `kg-m/sec2` means "kilograms x meters/sec2," which is often read as "kilogram-meters per seconds squared."

6. If the units you specify after you `have :` and you `want :` are incompatible:

```
you have: ft
you want: lb
```

The **units** command returns the following message and values:

```
conformability
3.048000e-01 m
4.535924e-01 kg
```

The `conformability` message means the units you specified cannot be converted. Feet measure length, and pounds measure mass, so converting from one to the other does not make sense. Therefore, the **units** command displays the equivalent of each value in standard units.

In other words, this example shows that one foot equals 0.3048 meters and that one pound equals 0.4535924 kilograms. The **units** command shows the equivalents in meters and kilograms because the command considers these units to be the standard measures of length and mass.

Files

Item	Description
<code>/usr/bin/units</code>	Contains the units command.
<code>/usr/share/lib/unittab</code>	Lists units that the units command creates as well as units defined by the user.

unlink Command

Purpose

Performs an **unlink** subroutine.

Syntax

unlink *File*

Description

The **unlink** command performs the **unlink** subroutine on a specified file.

The **unlink** command does not issue error messages when the associated subroutine is unsuccessful; you must check the exit value to determine if the command completed normally. It returns a value of 0 if it succeeds, a value of 1 if too few or too many parameters are specified, and a value of 2 if its system call is unsuccessful.

Attention: The **unlink** command allows a user with root user authority to deal with unusual problems, such as moving an entire directory to a different part of the directory tree. It also permits you to create directories that cannot be reached or escaped from. Be careful to preserve the directory structure by observing the following rules:

- Be certain every directory has a . (dot) link to itself.
- Be certain every directory has a .. (dot dot) link to its parent directory.
- Be certain every directory has no more than one link to itself or its parent directory.
- Be certain every directory is accessible from the root of its file system.

An attempt to remove a file or directory that has been exported for use by the NFS version 4 server will fail with a message saying that the resource is busy. The file or directory must be unexported for NFS version 4 use before it can be removed.

Example

To remove a directory entry pointed by `file2`, enter:

```
unlink file2
```

Files

Item	Description
<code>/usr/sbin/unlink</code>	Contains the unlink command.

unloadipsec Command

Purpose

Unloads a crypto module from the IP Security subsystem.

Syntax

unloadipsec -c *crypto_mod_name*

Description

The **unloadipsec** command unloads a crypto module from the IP Security subsystem. The **unloadipsec** command can be used when a crypto module is no longer being used or when a crypto module is to be replaced with a newer version.

A crypto module can only be unloaded after the IP Security device is stopped. The steps for replacing a crypto module are: change the IP Security device to the defined state; unload the old crypto module using this command; uninstall the old module and install the new module, and bring the IP Security device back to the available state.

Flags

Item	Description
-c <i>crypto_mod_name</i>	Specifies the name of the crypto module to be unloaded. When used without any flag, the command lists all the crypto modules installed (but not necessarily loaded).

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

unmirrorvg Command

Purpose

Removes the mirrors that exist on volume groups or specified disks.

Syntax

unmirrorvg [**-c** *Copies*] [**-p** *mirrorpool*] *VolumeGroup* [*PhysicalVolume...*]

Description

The **unmirrorvg** command unmirrors all the logical volumes detected on a given volume group. This same functionality may also be accomplished manually if you execute the **rm_lvcopy** command for each individual logical volume in a volume group.

By default, **unmirrorvg** will pick the set of mirrors to remove from a mirrored volume group. If you wish to control which drives no longer are to contain mirrors, you must include the list of disks in the input parameters, *PhysicalVolume*.

When the *PhysicalVolume* parameter is listed in the command, this indicates that only logical volumes with copies that exist on this *PhysicalVolume* should be unmirrored. Logical volumes that exist solely on the other drives in the volume group are unaffected and remain mirrored.

Note:

1. If LVM has not recognized that a disk has failed it is possible that LVM will remove a different mirror. Therefore if you know that a disk has failed and LVM does not show those disks as missing you should specify the failed disks on the command line or you should use **replacepv** to replace the disk or **reducevg** to remove the disk.
2. If a logical volume copy spans more than one disk, the portion of the logical volume copy that resides on a disk not listed by the user is also removed.
3. The **unmirrorvg** command is not allowed on a snapshot volume group.
4. Using a *PhysicalVolume* list with the **-c 1** option (the default) will cause affected triply-mirrored logical volumes to have two copies removed. Only one of these copies will be related to the listed physical volumes. This is because the physical volume list is used to determine affected logical volumes, which are then reduced to the specified number of copies. In this case, the second copy to remove is selected by **unmirrorvg**.
5. When a corresponding hard disk and **/dev/ipldevice** are removed then a reboot is required.
6. If you are removing the first mirror pool copy by specifying the disks in the first copy to remove, you might also want to move your logical volumes mirror pool assignments by running the **chlv** command. For example:

```
chlv -m copy1=poolb -M 2 lv00
```

When **unmirrorvg** is executed, the default COPIES value for each logical volume becomes 1. If you wish to convert your volume group from triply mirrored to doubly mirrored, use the **-c** option.

Note: To use this command, you must either have root user authority or be a member of the **system** group.



Attention: The **unmirrorvg** command may take a significant amount of time to complete because of complex error checking and the number of logical volumes to unmirror in a volume group.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit unmirrorvg
```

Flags

Item	Description
-c <i>Copies</i>	Specifies the minimum number of copies that each logical volume must have after the unmirrorvg command has finished executing. If you do not want all logical volumes to have the same number of copies, then reduce the mirrors manually with the rmlvcopy command. If this option is not used, the copies will default to 1.
-p <i>mirrorpool</i>	Removes the copy that exists on specified mirror pools. To remove more than one copy, provide multiple [-p mirrorpool] flags.

The following is a description of **rootvg**:

Item	Description
rootvg unmirroring	When the rootvg unmirroring has completed, you must perform two additional tasks: bosboot and bootlist. The bosboot command is required to reinitialize the boot record on the remaining disk. The bootlist command needs to be performed so that the system will only boot to the disk left in rootvg.

Examples

1. To unmirror a triply mirrored volume group and leave two copies, enter:

```
unmirrorvg -c 2 workvg
```

The logical partitions in the logical volumes held on **workvg** now have 2 copies.

2. To get default unmirroring of **rootvg**, enter:

```
unmirrorvg rootvg
```

rootvg now has only 1 copy.

3. To replace a bad disk drive in a mirrored volume group, enter:

```
unmirrorvg workvg hdisk7
reducevg workvg hdisk7
rmdev -l hdisk7 -d
replace the disk drive, let the drive be renamed hdisk7
extendvg workvg hdisk7
mirrorvg workvg
```

Note: By default in this example, **mirrorvg** will try to create 2 copies for logical volumes in **workvg**. It will try to create the new mirrors onto the replaced disk drive. However, if the original system had been triply mirrored, there may be no new mirrors created onto **hdisk7**, as other copies may already exist for the logical volumes. This follows the default behavior of **unmirrorvg** to reduce the mirror copy count to 1.

Note: When **unmirrorvg workvg hdisk7** is run, **hdisk7** will be the remaining drive in the volume group. This drive is not actually removed from the volume group. You must run the **migratepv** command to move the data from the disk that is to be removed from the system to disk **hdisk7**.

Files

Item	Description
/usr/sbin	Directory where the unmirrorvg command resides.

unpack Command

Purpose

Expands files.

Syntax

unpack *File ...*

Description

The **unpack** command expands files created by the **pack** command. For each file specified, the **unpack** command searches for a file called *File.z*. If this file is a packed file, the **unpack** command replaces it by its expanded version. The **unpack** command names the new file name by removing the **.z** suffix from *File*. If the user has root authority, the new file has the same access modes, access and modification times, owner, and group as the original file. If the user does not have root authority, the file retains the same access modes, access time, and modification time, but acquires a new owner and group.

The **unpack** command operates only on files ending in **.z**. As a result, when you specify a file name that does not end in **.z**, the **unpack** command adds the suffix and searches the directory for a file name with that suffix.

The exit value is the number of files the **unpack** command was unable to unpack. A file cannot be unpacked if any of the following occurs:

- The file name (exclusive of **.z**) has more than 253 bytes.
- The file cannot be opened.
- The file is not a packed file.
- A file with the unpacked file name already exists.
- The unpacked file cannot be created.

Note: The **unpack** command writes a warning to standard error if the file it is unpacking has links. The new unpacked file has a different i-node than the packed file from which it was created. However, any other files linked to the original i-node of the packed file still exist and are still packed.

Exit Status

This command returns the following exit values:

Item	Description
0	The command ran successfully.
>0	An error occurred.

Example

To unpack packed files:

```
unpack chap1.z chap2
```

This expands the packed files `chap1.z` and `chap2.z`, and replaces them with files named `chap1` and `chap2`. Note that you can give the **unpack** command file names either with or without the **.z** suffix.

Files

Item	Description
<code>/usr/bin/unpack</code>	Contains the unpack command.

untab Command

Purpose

Changes tabs into spaces.

Syntax

untab [*FileName ...*]

Description

The **untab** command reads the file specified by the *FileName* parameter or standard input, and replaces tabs in the input with space characters. If you specify a file with the *FileName* parameter, the **untab** command writes the resulting file back to the original file. If the input is standard input, the **untab** command writes to standard output. The **untab** command assumes that tab stops are set every eight columns, starting with column nine. The file name specified for the *FileName* parameter cannot exceed **PATH_MAX**-9 bytes in length.

Example

To replace tab characters in the `File` file with space characters, enter:

```
untab File
```

Files

Item	Description
<code>/usr/bin/untab</code>	Contains the untab command.

update Command

Purpose

Periodically updates the super block.

Syntax

update

Description

The **update** command executes a **sync** subroutine every 30 seconds. This action ensures the file system is up-to-date in the event of a system crash.

Files

Item	Description
<code>/usr/sbin/update</code>	Contains the update command.

update_iscsi Command

Purpose

Lists and updates the configurations of devices for the iSCSI software initiator that is accessed through the iSCSI software initiator or the iSCSI TOE adapter.

Syntax

update_iscsi [**-l** *name*]

Description

The **update_iscsi** command lists and updates the devices for which configuration attributes are related to iSCSI and must be migrated to the Object Data Manager (ODM) of the **rootvg** image.

You can run the **update_iscsi** command in maintenance mode after all of the file systems that contain the base operating system in the **rootvg** image are mounted. Note that only the devices that are causing iSCSI boot problems should be updated.

To list the devices for which the iSCSI configuration attributes are changed, run the **update_iscsi** command without any argument.

To migrate the configuration of a listed device to the ODM of the **rootvg** image, run the **update_iscsi** command with the **-l** *name* flags. The *name* parameter represents the ODM name of a device in the RAM file system.

The **update_iscsi** command displays the devices that are listed in the **iscsi_devlist** file, which is located in the **/etc/objrepos** directory. The command lists these devices after matching them to the corresponding **rootvg** entries. If the **iscsi_devlist** file is missing, or if the file lists no devices, a message will be printed indicating that you did not set the ODM for the RAM file system.

Flags

Item	Description
-l	Specifies the ODM name of a device in the RAM file system. This flag is optional.

Parameters

Item	Description
<i>name</i>	The ODM name of a device in the RAM file system.

Sample Output

The following sample shows the output of the **update_iscsi** command with no flag specified:

RAM FS DEVICE NAME	ROOTVG DEVICE NAME	DESCRIPTION
inet0	inet0	Internet Network Extension
en0	en1	Standard Ethernet Network Interface
iscsi0	iscsi0	iSCSI Protocol Device

Exit Status

If the **update_iscsi** command cannot find the ODM name that the *name* parameter specifies, the value of the **ROOTVG DEVICE NAME** is set to *New Device*.

If the **iscsi_devlist** file is missing or empty, an error message is printed.

Location

/usr/sbin/

Files

Item	Description
iscsi_devlist	Contains a list of the devices with attributes that are set through the Network Disk Install menu.

updatevsdnode Command

Purpose

Modifies virtual shared disk subsystem options.

Syntax

updatevsdnode

-n {**ALL** | *node_number* [,*node_number* ...]}

{**-a** *VSD_adapter*}

-b *min_buddy_buffer_size*

-x *max_buddy_buffer_size*

-s *max_buddy_buffers*

-M *vsd_max_ip_packet_size*}

-f [**-c** *cluster_name* | NONE]

Description

Use **updatevsdnode** to modify virtual shared disk subsystem options.

Note: This command only modifies the subsystem options. To effectively configure the virtual shared disks, you must first unconfigure all the virtual shared disks, unload the device driver, and then reconfigure the shared disks.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit vsd_mgmt
```

and select the **Set/Show Virtual Shared Disk Device Driver Operational Parameters** option or the **Update virtual shared disk Device Driver Node Parameters** option.

Flags

-n

Specifies the node numbers of the nodes whose information you want this command to update, or **ALL** nodes in the RSCT peer domain. You can issue the command **/usr/bin/lscslcfg** to find out the node number of the node you are running on.

-a

Specifies the adapter name to be used for virtual shared disk communications with this node or nodes. You must specify **ml0** as the adapter name.

-b

Specifies the smallest buddy buffer a server uses to satisfy a remote request to a virtual shared disk. This value must be a power of 2 and greater than or equal to 4096. The suggested value to use is 4096 (4 KB).

-x

The largest buddy buffer a server will use to satisfy a remote request. This value must be a power of 2 and greater than or equal to the *min_buddy_buffer_size*. The suggested value to use is 262144 (256 KB). This value must be the same on all nodes in the RSCT peer domain.

-s

This is the number of *max_buddy_buffer_size* buffers to allocate. The virtual shared disk device driver will have an initial size when first loaded, and then will dynamically allocate and reclaim additional space as needed. The suggested starting value for a 32-bit kernel is 128 256 KB buffers. The suggested value is 2000 256KB buffers.

Buddy buffers are only used on the servers. On client nodes you may want to set *max_buddy_buffers* to 1.

Note: The `statvsd` command will indicate if remote requests are queueing waiting for buddy buffers.

-M

Specifies the maximum message size in bytes for virtual shared disks. This value must not be greater than the maximum transmission unit (MTU) size of the network. The recommended values are:

- 61440 (60 KB) for a switch
- 8192 (8 KB) for jumbo frame Ethernet
- 1024 (1 KB) for 1500-byte MTU Ethernet

-f

Specifies that this command will force updates to virtual shared disk subsystem options by reconfiguring one or more virtual shared disks on all nodes in the RSCT peer domain on which virtual shared disks are currently configured.

-c *cluster_name* | NONE

Changes the cluster the node belongs to. NONE removes the node from the cluster.

Note: The *cluster_name* is required only for SSA (Serial Storage Architecture) disks.

Parameters

vsd_name

Specifies the virtual shared disk whose underlying logical volume you no longer want to be globally accessed by any virtual shared disk nodes.

Security

You must have `root` authority to run this command.

Exit Status

0

Indicates the successful completion of the command.

nonzero

Indicates that an error occurred.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **startpdomain** command. To bring a particular node online in an existing peer domain, use the **startpnode** command. For more information on creating and administering an RSCT peer domain, refer to *RSCT Administration Guide*.

Examples

To increase the buddy buffer size to 48 maximum sized buddy buffers on node 3, enter:

```
updatevsdnode -n 3 -s 48
```

Note: The device driver must be unconfigured from the kernel and reloaded to have this change go into effect.

Location

/opt/lpp/vsd/bin/updatevsdnode

updatevsdtab Command

Purpose

updatevsdtab – Changes the Virtual shared disk subsystem attributes.

Syntax

updatevsdtab {-v *vsd_names* | -a} [{-s }] [-f]

Description

Use this command to update the virtual shared disk size. When you change the virtual shared disk size using the **updatevsdtab** command, the change will not take effect until the virtual shared disk is unconfigured and configured again.

If the **-f** flag is specified, the virtual shared disks involved will be reconfigured on all nodes that are up and initially had these virtual shared disks configured.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit vsd_mgmt
```

and select the Set/Show virtual shared disk Device Driver Operational Parameters option or the Update virtual shared disk Options option.

Flags

-v *vsd_names*

Specifies a list of virtual shared disk names to be updated.

-a

Specifies that the option is to be changed on all nodes of the system or system partition.

-s

Updates the virtual shared disk size after the associated logical volume size is changed.

-f

Forces changes by reconfiguring a virtual shared disk on all nodes in the current system partition on which the virtual shared disk is configured.

Parameters

None.

Security

You must have root authority to run this command.

Exit Status

0

Indicates the successful completion of the command.

nonzero

Indicates that an error occurred.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **starttrpdomain** command. To bring a particular node online in an existing peer domain, use the **starttrpnode** command. For more information on creating and administering an RSCT peer domain, refer to the *RSCT: Administration Guide*.

Standard Output

Current RVSD subsystem run level.

Examples

1. To reset the size of the virtual shared disk named USER1n3, enter:

```
updatevsdtab -v USER1n3 -s
```

Location

/usr/lpp/csd/bin/updatevsdtab

updatevsdvg Command

Purpose

Changes virtual shared disk global volume group characteristics.

Syntax

```
updatevsdvg { -a | -g global_volgrp { -k VSD -p primary_node -b secondary_node | -k CVSD -l server_list [-c cluster_name] } }
```

Description

The `updatevsdvg` command changes virtual shared disk global volume group characteristics. This command allows you to change global volume groups from concurrent virtual shared disk volume groups to serial-access (or nonconcurrent) virtual shared disk volume groups, and the other way around. This command can be used whenever server node numbers change, such as replacing or re-cabling servers where the new server numbers are different, or when you need to delete a server.

This command performs the following operations:

1. Suspends all virtual shared disks that are part of this volume group

2. Stops all virtual shared disks that are part of this volume group
3. Issues the `varyoffvg` command for the volume group
4. Verifies that the volume group exists on the new servers and tries to import the volume group if it does not exist
5. Updates the global volume group characteristics
6. Issues the `varyonvg` command for the volume group to the appropriate servers
7. Starts all virtual shared disks that are part of this volume group

Note:

1. If you issue this command with the `-a` flag, the recoverable virtual shared disk subsystem should not be active. Otherwise, this command can be run while the recoverable virtual shared disk subsystem is active, as long as no application is using the virtual shared disks that are part of the volume group being updated.
2. Concurrent virtual shared disks are supported for disks that have implemented the SCSI-3 persistent reserve model of the AIX SCSI device drivers, and for SSA (Serial Storage Architecture) disks.

Flags

-a

Specifies that persistent reserve information should be reestablished in the object data manager (ODM) for all VSD volume groups served by this node. This flag is intended for the initial setup phase of allowing multiple clusters to access the same virtual shared disks. It is also useful for recovery after the device ODM entries have been removed inadvertently.

This flag causes all of the volume groups served by the node to be varied offline. The volume groups will be varied offline on this node and on all other servers for the volume groups. For this reason, you should stop the recoverable virtual shared disk subsystem before issuing the `updatevsdvg` command with this flag.

-b *secondary_node*

Specifies the secondary node.

-c *cluster_name*

Specifies the cluster name for the server nodes that will be serving concurrently accessed shared disks. This flag is applicable only for SSA (Serial Storage Architecture) disks, and a *cluster_name* must be specified for SSA.

-g *global_volgrp*

Specifies an existing global volume group name.

-k VSD | CVSD

Specifies whether the volume group will be of type concurrent virtual shared disk or serial-access (nonconcurrent) virtual shared disk.

-l *server_list*

Specifies a colon-separated list of servers for concurrent virtual shared disks.

-p *primary_node*

Specifies the primary node.

Parameters

vsd_name

Specifies the virtual shared disk whose underlying logical volume you no longer want to be globally accessed by any virtual shared disk nodes.

Security

You must have `root` authority to run this command.

Exit Status

0

Indicates the successful completion of the command.

nonzero

Indicates that an error occurred.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **startdomain** command. To bring a particular node online in an existing peer domain, use the **startnode** command. For more information on creating and administering an RSCT peer domain, refer to *RSCT Administration Guide*.

Standard Output

Current RVSD subsystem run level.

Examples

1. To change the global volume group named `ess_gvg` from a virtual shared disk global volume group to a concurrent global volume group with three servers, assuming that the disks are cabled correctly and that the disk subsystem supports persistent preserve such as ESS disks, enter:

```
updatevsdvg -g ess_gvg -k CVSD -l 9:17:21
```

2. To remove a server from an SSA global volume group named `ssa_gvg`, where the original server list is `9:10` and belongs to an SSA cluster named `cluster9_10`, (that is, the command `vsdata1st -c` shows SSA cluster information), enter:

```
updatevsdvg -g ssa_gvg -k CVSD -l 9 -c cluster9_10
```

3. To change a concurrent global volume group named `ess_gvg` back to a virtual shared disk global volume group, where the original server list is `9:17:21`, the new primary node number is `9`, and the new secondary node number is `21`, enter:

```
updatevsdvg -g ess_gvg -k VSD -p 9 -b 21
```

Location

`/opt/rsct/vsd/bin/updatevsdvg`

uprintfd Daemon

Purpose

Constructs and writes kernel messages.

Syntax

uprintfd

Description

The **uprintfd** daemon retrieves, converts, formats, and writes kernel messages to processes' controlling terminals. Kernel messages are submitted through the **NLuprintf** and **uprintf** kernel services. Because the **uprintfd** daemon never exits, it should be run only once.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

uptime Command

Purpose

Shows how long the system has been up.

Syntax

uptime

Description

The **uptime** command prints the current time, the length of time the system has been up, the number of users online, and the load average. The load average is the number of runnable processes over the preceding 1-, 5-, 15-minute intervals. The output of the **uptime** command is, essentially, the heading line provided by the **w** command.

useradd Command

Purpose

Creates a new user account.

Syntax

```
useradd [ -c comment ] [ -d dir ] [ -e expire ] [ -g group ] [ -G group1,group2 ... ] [ -m [ -k skel_dir ] ] [ -u uid ] [ -s shell ] [ -r role1,role2 ... ] login
```

Description

The **useradd** command creates a new user account. The *login* parameter must be a unique string (its length is can be configured by administrators using the **chdev** command). You cannot use the **ALL** or **default** keywords in the user name.

The **useradd** command does not create password information for a user. It initializes the password field with an asterisk (*). Later, this field is set with the **passwd** or **pwdadm** command. New accounts are disabled until the **passwd** or **pwdadm** commands are used to add authentication information to the **/etc/security/passwd** file.

The **useradd** command always checks the target user registry to make sure the ID for the new account is unique to the target registry. The **useradd** command can also be configured to check all user registries of the system using the **dist_uniqid** system attribute. The **dist_uniqid** system attribute is an attribute of the **usw** stanza of the **/etc/security/login.cfg** file, and can be managed using the **chsec** command.

The **dist_uniqid** system attribute has the following values:

never

Does not check for ID collision against the nontarget registries. This is the default setting.

always

Checks for ID collision against all other registries. If collision is detected between the target registry and any other registry, account creation or modification fails.

uniqbyname

Checks for ID collision against all other registries. Collision between registries is allowed only if the account to be created has the same name as the existing account.

Note: ID collision detection in the target registry is always enforced regardless of the `dist_uniqid` system attribute.

The `uniqbyname` system attribute setting works well against two registries. With more than two registries, and with ID collision already existing between two registries, the behavior of the `useradd` command is unspecified when creating a new account in a third registry using colliding ID values. The new account creation might succeed or fail depending on the order in which the registries are checked.

The check for ID collision only enforces ID uniqueness between the local registry and remote registries, or between remote registries. There is no guarantee of ID uniqueness between the newly created account on the remote registry and existing local users on other systems that make use of the same remote registry. The `useradd` command bypasses a remote registry if the remote registry is not reachable at the time the command is run.

Flags

Item	Description
<code>-c comment</code>	Supplies general information about the user specified by the <i>login</i> parameter. The <i>comment</i> parameter is a string with no embedded colon (:) characters and cannot end with the characters '#! '.
<code>-d dir</code>	Identifies the home directory of the user specified by the <i>login</i> parameter. The <i>dir</i> parameter is a full path name.
<code>-e expire</code>	Identifies the expiration date of the account. The <i>expire</i> parameter is a 10-character string in the <i>MMDDhhmmyy</i> form, where <i>MM</i> is the month, <i>DD</i> is the day, <i>hh</i> is the hour, <i>mm</i> is the minute, and <i>yy</i> is the last 2 digits of the years 1939 through 2038. All characters are numeric. If the <i>expire</i> parameter is 0, the account does not expire. The default is 0. See the <code>date</code> command for more information.
<code>-g group</code>	Identifies the user's primary group. The <i>group</i> parameter must contain a valid group name and cannot be a null value.
<code>-G group1,group2,...</code>	Identifies the groups the user belongs to. The <i>group1,group2,...</i> parameter is a comma-separated list of group names.
<code>-k skel_dir</code>	Copies default files from <i>skel_dir</i> to user's home directory. Used only with <code>-m</code> flag.
<code>-m</code>	Makes user's home directory if it does not exist. The default is not to make the home directory.
<code>-r role1,role2,...</code>	Lists the administrative roles for this user. The <i>role1,role2,...</i> parameter is a list of role names, separated by commas.

Item	Description
-s <i>shell</i>	Defines the program run for the user at session initiation. The <i>shell</i> parameter is a full path name.
-u <i>uid</i>	Specifies the user ID. The <i>uid</i> parameter is a unique integer string. Avoid changing this attribute so that system security will not be compromised.

Exit Status

Item	Description
0	The command completed successfully.
>0	An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To create the `davis` user account with default values, enter:

```
useradd davis
```

Restrictions

To prevent login inconsistencies, avoid composing user names entirely of uppercase alphabetic characters. While the `useradd` command supports multibyte user names, restrict user names to characters with the POSIX-portable filename character set.

To ensure that your user database remains uncorrupted, you must be careful when naming users. User names must not begin with a hyphen (-), plus sign (+), at sign (@), or tilde (~). You cannot use the keywords `ALL` or `default` in a user name. Additionally, do not use any of the following characters within a user-name string:

Item	Description
:	Colon
"	Double quote
#	Pound sign
,	Comma
=	Equal sign
\	Back slash
/	Slash
?	Question mark
'	Single quote
`	Back quote

Finally, the *login* parameter cannot contain any space, tab, or newline characters.

Location

/usr/sbin/useradd

Files

The useradd command has read and write permissions to the following files.

Item	Description
/etc/passwd	Contains the basic attributes of users.
/etc/security/user	Contains the extended attributes of users.
/etc/security/user.roles	Contains the administrative role attributes of users.
/etc/security/limits	Defines resource quotas and limits for each user.
/etc/security/envIRON	Contains the environment attributes of users.
/etc/security/audit/config	Contains audit configuration information.
/etc/security/lastlog	Contains the last login attributes of users.
/etc/group	Contains the basic attributes of groups.
/etc/security/group	Contains the extended attributes of groups.

userdel Command

Purpose

Removes a user account.

Syntax

userdel [-r] *login*

Description

The **userdel** command removes the user account identified by the *login* parameter. The command removes a user's attributes without removing the user's home directory by default. The user name must already exist. If the -r flag is specified, the **userdel** command also removes the user's home directory.

If the **AIX_USERDEL_RECURSIVE_DEL** environment variable is set, the **userdel** command recursively deletes the directories and files that belong to the removed user. If another user uses the same home directory, the files and directories of the user is preserved. If the directory of the deleted user contains content owned by a different user, the directory ownership of the user is changed to the user **nobody** with a permission of **777** and a **sticky bit** set. This operation is performed for the continued access of the directory and its content for the affected users by using the same home space. It is very important to change the permission and ownership of the affected directories to a new user immediately after running the **userdel** command. The system administrator can change the permission and ownership setting of the affected directories to a new user to prevent illegal access.

Only the root user or users with UserAdmin authorization can remove administrative users. Administrative users are those users with admin=true set in the /etc/security/user file.

Flags

Item	Description
-r	Removes the home directory of the user. Files located in other file systems must be searched manually and deleted. Removing the home directory, which is shared by other users, might leave the system in an inconsistent state.

Exit Status

Item	Description
0	The command completed successfully.
>0	An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To remove the user davis account and its attributes from the local system, enter:

```
userdel davis
```

Location

/usr/sbin/userdel

Files

The userdel command has read and write permissions to the following files.

Item	Description
/etc/passwd	Contains the basic attributes of users.
/etc/security/user	Contains the extended attributes of users.
/etc/security/user.roles	Contains the administrative role attributes of users.
/etc/security/limits	Defines resource quotas and limits for each user.
/etc/security/envIRON	Contains the environment attributes of users.
/etc/security/audit/config	Contains audit configuration information.
/etc/security/lastlog	Contains the last login attributes of users.
/etc/group	Contains the basic attributes of groups.
/etc/security/group	Contains the extended attributes of groups.

usermod Command

Purpose

Changes user attributes.

Syntax

```
usermod [ -u uid ] [ -g pgroup ] [ -G group1,group2 ... ] [ -d dir [ -m ] ] [ -s shell ] [ -c comment ] [ -l new_name ] [ -e expire ] [ -r role1,role2 ... ] login
```

Description



Attention: Do not use the `usermod` command if you have a Network Information Service (NIS) database installed on your system.

The `usermod` command changes attributes for the user identified by the *login* parameter. The user name must already exist. To change an attribute, specify the flag and the new value. The following files contain local user attributes that are set by this command:

- `/etc/passwd`
- `/etc/security/environ`
- `/etc/security/limits`
- `/etc/security/user`
- `/etc/security/user.roles`
- `/etc/security/audit/config`
- `/etc/group`
- `/etc/security/group`

Avoid changing the ID for an account so that system security is not compromised. However, when the ID is changed using the `usermod` command, ID collision checking is also controlled by the `dist_uniqid` attribute in the `usw` stanza of the `/etc/security/login.cfg` file. The behavior of ID collision control is the same as that described for the `mkuser` command.

Flags

Item	Description
-c <i>comment</i>	Supplies general information about the user specified by the <i>login</i> parameter. The <i>comment</i> parameter is a string with no embedded colon (:) characters and cannot end with the characters '#! '.
-d <i>dir</i>	Changes the home directory to the directory specified by the <i>dir</i> parameter.
-g <i>pgroup</i>	Identifies the primary group. The <i>pgroup</i> parameter must be a valid group name or ID.
-e <i>expire</i>	Identifies the expiration date of the account. The <i>expire</i> parameter is a 10-character string in the <i>MMDDhhmm</i> yy form, where <i>MM</i> is the month, <i>DD</i> is the day, <i>hh</i> is the hour, <i>mm</i> is the minute, and <i>yy</i> is the last 2 digits of the years 1939 through 2038. All characters are numeric. If the <i>expire</i> parameter is 0, the account does not expire. The default is 0. See the <code>date</code> command for more information.
-G <i>group1,group2,...</i>	Identifies the groups the user belongs to. The <i>group1,group2,...</i> parameter is a comma-separated list of group names.
-l <i>new_name</i>	Specifies the new name of the user.

Item	Description
-m	Moves the contents of the user's current home directory to the new home directory. Only used with the -d flag.
-r <i>role1,role2,...</i>	Lists the administrative roles for this user. The <i>role1,role2,...</i> parameter is a list of role names, separated by commas.
-s <i>shell</i>	Defines the program run for the user at session initiation. The <i>shell</i> parameter is a full path name.
-u <i>uid</i>	Specifies the user ID. The <i>uid</i> parameter is a unique integer string. Avoid changing this attribute so that system security will not be compromised.

Exit Status

Item	Description
0	The command completed successfully.
>0	An error occurred.

Examples

1. To change the user `davis` to be a member of the `system` group, enter the following command:

```
usermod -G system davis
```

Restrictions

To ensure the integrity of user information, some restrictions apply when using the **usermod** command. Only the root user or users with `UserAdmin` authorization can use the `usermod` command to perform the following tasks:

- Make a user an administrative user by setting the `admin` attribute to `true`.
- Change any attributes of an administrative user.
- Add a user to an administrative group

An administrative group is a group with the `admin` attribute set to `True`. Members of the security group can change the attributes of non-administrative users and add users to non-administrative groups.

The `usermod` command manipulates local user data only. You cannot use it to change data in registry servers like NIS and DCE.

Location

`/usr/sbin/usermod`

Files

The `usermod` command has read and write permissions to the following files.

Item	Description
<code>/etc/passwd</code>	Contains the basic attributes of users.
<code>/etc/security/user</code>	Contains the extended attributes of users.
<code>/etc/security/user.roles</code>	Contains the administrative role attributes of users.
<code>/etc/security/limits</code>	Defines resource quotas and limits for each user.

Item	Description
/etc/security/environ	Contains the environment attributes of users.
/etc/security/audit/ config	Contains audit configuration information.
/etc/security/lastlog	Contains the last login attributes of users.
/etc/group	Contains the basic attributes of groups.
/etc/security/group	Contains the extended attributes of groups.

users Command

Purpose

Displays a compact list of the users currently logged on to the system.

Syntax

users [*FileName* | *WparName*]

Description

The **users** command lists the login names of the users that are currently logged on to the system to standard output (**stdout**) in a compact, one-line list format. If you specify absolute path name of a file, then it is used as an alternate file instead of **/etc/utmp**. If you do not specify an absolute path name, it is considered to be the name of a workload partition. If the name is "Global", it indicates the global environment.

Files

Item	Description
/etc/utmp	Contains list of current users.
/usr/bin/users	Contains the users command.

Note: The /etc/utmp file for a particular workload partition can be indicated by prefixing the root path for the workload partition.

usrck Command

Purpose

Verifies the correctness of a user definition.

Syntax

usrck { -l | -b | -n | -p | -t | -y } { **ALL** | *User ...* }

Description

The **usrck** command verifies the correctness of the user definitions in the user database files, by checking the definitions for **ALL** the users or for the users specified by the *User* parameter. If more than one user is specified, there must be a space between the names. You must select a flag to indicate whether the system should try to fix erroneous attributes.

The command first checks the entries in the **/etc/passwd** file. If you indicate that the system should fix errors, duplicate user names are reported and disabled. Duplicate IDs are reported only, because there is no system fix. If an entry has fewer than six colon-separated fields, the entry is reported, but not fixed. The **usrck** command next checks specific user attributes in other files.

The **usrck** command verifies that each user name listed in the **/etc/passwd** file has a stanza in the **/etc/security/user**, **/etc/security/limits** and **/etc/security/passwd** files. The **usrck** command also verifies that each group name listed in the **/etc/group** file has a stanza in the **/etc/security/group** file. The **usrck** command using the **-y** flag creates stanzas in the security files for the missing user and group names.

Note:

- This command writes its messages to **stderr**.
- If the *domainlessgroups* attribute is set, the **usrck** command will throw an error for the Lightweight Directory Access Protocol (LDAP) users.

A list of all the user attributes follows, with notations stating which attributes are checked:

Item	Description
account_locked	No check. The usrck command sets this attribute to True and disables accounts.
admgroups	Checks to see if the admgroups are defined in the user database and, if you indicate that the system should fix errors, the command removes any groups that are not in the database.
auditclasses	Checks to see if the auditclasses are defined for the user in the /etc/security/audit/config file. If you indicate that the system should fix errors, the command deletes all the auditclasses that are not defined in the /etc/security/audit/config file.
auth1	Checks the primary authentication method. Unless the method is NONE or SYSTEM, it must be defined in the /etc/security/login.cfg file and the program attribute must exist and be executable by the root user. If you indicate that the system should fix errors, it will disable the user account if an error is found. Note: The auth1 attribute is deprecated and should not be used.
auth2	Checks the secondary authentication method. Unless the method is NONE or SYSTEM, it must be defined in the /etc/security/login.cfg file and the program attribute must exist and be executable by the root user. There is no system fix. Note: The auth2 attribute is deprecated and should not be used.
core	Ensures that the values are sensible. If not, the command resets the values to 200 blocks, the minimum value.
core_hard	Ensures that the values are sensible. If not, the command resets the values to 200 blocks, the minimum value.
cpu	Ensures that the values are sensible. If not, the command resets the values to 120 seconds, the minimum value.
cpu_hard	Ensures that the values are sensible. If not, the command resets the values to 120 seconds, the minimum value.
data	Ensures that the values are sensible. If not, the command resets the values to 1272 blocks (636K), the minimum value.
data_hard	Ensures that the values are sensible. If not, the command resets the values to 1272 blocks (636K), the minimum value.

Item	Description
dictionlist	Checks the list of dictionary files. If you indicate that the system should fix errors, all dictionary files that do not exist are deleted from the user database.
expires	No check.
fsize	Ensures that the values are sensible. If not, the command resets the values to 200 blocks, the minimum value.
fsize_hard	Ensures that the values are sensible. If not, the command resets the values to 200 blocks, the minimum value.
gecos	No check.
histexpire	Ensures that the values are sensible. If you indicate that the system should fix errors, values that are too large are set to the largest possible value and values that are too small are set to the smallest possible value.
histsize	Ensures that the values are sensible. If you indicate that the system should fix errors, values that are too large are set to the largest possible value and values that are too small are set to the smallest possible value.
home	Checks the existence and accessibility of the home directory by read mode and search mode. If you indicate that the system should fix errors, it will disable the user account if an error is found.
id	Checks the uniqueness of the user ID. If you indicate that the system should fix errors, the command deletes any invalid entry in the /etc/passwd file.
login	No check.
loginretries	Checks if the user attempted unsuccessful logins more than the allowable amount. If so, the system disables the user account.
logintimes	Ensures that the string of time specifiers is valid. If you indicate that the system should fix errors, the system disables the user account if an error is found.

Item	Description
maxage	Ensures that the values are sensible. If you indicate that the system should fix errors, values that are too large are set to the largest possible value and values that are too small are set to the smallest possible value.
maxexpired	Ensures that the values are sensible. If you indicate that the system should fix errors, values that are too large are set to the largest possible value and values that are too small are set to the smallest possible value.
maxrepeats	Ensures that the values are sensible. If you indicate that the system should fix errors, values that are too large are set to the largest possible value and values that are too small are set to the smallest possible value.
minage	Ensures that the values are sensible. If you indicate that the system should fix errors, values that are too large are set to the largest possible value and values that are too small are set to the smallest possible value. The system also indicates if the minage attribute is larger than the maxage attribute.
minalpha	Ensures that the values are sensible. If you indicate that the system should fix errors, values that are too large are set to the largest possible value and values that are too small are set to the smallest possible value.
mindiff	Ensures that the values are sensible. If you indicate that the system should fix errors, values that are too large are set to the largest possible value and values that are too small are set to the smallest possible value.

Item	Description
minlen	Ensures that the values are sensible. If you indicate that the system should fix errors, values that are too large are set to the largest possible value and values that are too small are set to the smallest possible value.
minother	Ensures that the values are sensible. If you indicate that the system should fix errors, values that are too large are set to the largest possible value and values that are too small are set to the smallest possible value. The system also indicates if the minage attribute plus the maxage attribute is greater than the maximum password size.
name	Checks the uniqueness and composition of the user name. The name must be a unique string of eight bytes or less. It cannot begin with a + (plus sign), a : (colon), a - (minus sign), or a ~ (tilde). Names beginning with a + (plus sign) or with a - (minus sign) are assumed to be names in the NIS (Network Information Service) domain, and no further processing is performed. It cannot contain a colon (:) in the string and cannot be the ALL or default keywords. If you indicate that the system should fix errors, the command disables the user account if an error is found and deletes any invalid entry in the /etc/passwd file. The usrck command verifies that, for each user name listed in the /etc/passwd file, there is a stanza in the /etc/security/user, /etc/security/limits, and /etc/security/passwd files. The command adds stanzas for each one identified as missing. The usrck command additionally verifies that each group name listed in the /etc/group file has a stanza in the /etc/security/group file.
nofiles	Ensures that the value is sensible. If not, resets the value to 200, the minimum value.
nofiles_hard	Ensures that the value is sensible. If not, resets the value to 200, the minimum value.
pgrp	Checks for the existence of the primary group in the user database. If you indicate that the system should fix errors, it will disable the user account if an error is found.
pwdchecks	Checks the list of external password restriction methods. If you indicate that the system should fix errors, all methods that do not exist are deleted from the user database.
pwdwarntime	Ensures that the value is sensible. If not, the system resets the value to the difference between the maxage and minage values.
rlogin	No check.
rss	Checks to ensure that the values are sensible. If not, the command resets the values to 128 blocks (64KB), the minimum value. The value is not set by the system.
rss_hard	Checks to ensure that the values are sensible. If not, the command resets the values to 128 blocks (64KB), the minimum value. The value is not set by the system.
shell	Checks the existence and accessibility of the shell by execute mode. If you indicate that the system should fix errors, it will disable the user account if an error is found.
stack	Checks to ensure that the values are sensible. If not, the command resets the values to 128 blocks (64KB), the minimum value.

Item	Description
stack_hard	Checks to ensure that the values are sensible. If not, the command resets the values to 128 blocks (64KB), the minimum value.
su	No check.
sugroups	Checks for the existence of the sugroups in the user database files. If you indicate that the system should fix errors, it will delete all the groups that are not in the database.
sysenv	No check.
tpath	Checks to ensure that the shell attribute is tagged as a trusted process if tpath=always . If you indicate that the system should fix errors, it will disable the user account if an error is found.
ttys	Checks for the existence of the ttys in the user database files. If you indicate that the system should fix errors, it will delete all the ttys that do not exist from the user database.
usrenv	No check.

If the fix involves disabling a user account, use the **chuser** command to reset the value of the **account_locked** attribute to False. You can use the System Management Interface Tool (SMIT) to run the **chuser** command by entering:

```
smit chuser
```

The root user or a member of the security group can enable a user account again by removing the **account_locked** attribute or setting the **account_locked** attribute to False. The root user's account is not disabled by the **usrck** command.

Generally, the **sysck** command calls the **usrck** command as part of the verification of a trusted-system installation. If the **usrck** command finds any errors in the user database, the root user or a member of the security group should execute both the **grpck** command and the **pwdck** command.

The **usrck** command checks to see if the database management security files (**/etc/passwd.nm.idx**, **/etc/passwd.id.idx**, **/etc/security/passwd.idx**, and **/etc/security/lastlog.idx**) files are up-to-date or newer than the corresponding system security files. Please note, it is acceptable for the **/etc/security/lastlog.idx** to be not newer than **/etc/security/lastlog**. If the database management security files are out-of-date, a warning message appears indicating that the root user should run the **mkpasswd** command.

The **usrck** command checks if the specified user can log in. If the user cannot log in because of too many unsuccessful login attempts or because the password is expired, the **usrck** command issues a warning message indicating why the user cannot log in. If you indicate that the system should fix errors, the system disables the user account if the user cannot log in for the above reasons.

If the **-l** flag is specified, the **usrck** command scans all users or the users specified by the *User* parameter to determine if users can access the system. The criteria used to determine accessibility for a user are listed in the following table:

Table 29. User Accessibility Criteria		
Criterion	Description	Cause
1	User account is locked.	The user's account_locked attribute is set to true .
2	User account is expired.	The user's expires attribute is set to a value (expiration time) that is expired.

Table 29. User Accessibility Criteria (continued)		
Criterion	Description	Cause
3	User has too many consecutive failed login attempts.	The user's unsuccessful_login_count value is greater than the user's loginretries value.
4	User has no password.	The user's password field is '*' in /etc/passwd or /etc/security/password .
5	User is not allowed to log in for this date/time.	The current date/time is not within the allowed time as defined by the user's logintimes attribute.
6	The /etc/nologin file exists.	The /etc/nologin file prevents a non-root user from logging in.
7	User password is expired and only system administrator can change it.	The user's password is expired and the ADMIN password flag is set.
8	User is denied login to host.	The user's hostallowedlogin and hostsdeniedlogin attributes do not allow access to the current host.
9	User is denied access by applications.	The user's login , rlogin , and su attributes are set to false and the rcmds attribute is set to deny. If at least one but not all of these attribute values deny authorization, the system is considered partially accessible by the user.
10	User is denied login to terminal.	The user's ttys attribute does not allow access to the current terminal. The system is considered partially accessible for the user.

If the **-b** flag is also specified, the output consists of two fields, the user name and a 16-digit bit mask, separated by a tab. Each digit in the bit mask corresponds to a criteria in the User Accessibility Criteria table above, with criteria 1 represented by the rightmost digit. If the bit location for a criteria is set to 1, the check for this criteria failed for the user. Extra digits in the output are reserved for future use.

The following is an example of the **usrck** command with the **-l** flag:

```
# usrck -l testusr1 testusr2
3001-689 The system is inaccessible to testusr1, due to the following:
    User account is locked
    User denied login to terminal.

3001-689 The system is inaccessible to testusr2, due to the following:
    User account is expired.
    User has too many consecutive failed login attempts.
    User denied login to host.
```

The following is an example of the **usrck** command with the **-l** and **-b** flags:

```
# usrck -lb testusr1 testusr2
testusr1      0000000000000001
testusr2      0000000001000110
```

Flags

Item Description

-b Reports users who are not able to access the system and the reasons, with the reasons displayed in a bit-mask format. The **-l** flag must be specified if the **-b** flag is specified.

Note: The bit mask does not report [criteria 10](#) (user denied login to terminal), since this cannot be considered a complete scenario when determining if a system is inaccessible to a user. Likewise, the bit mask does not report [criteria 9](#) (User denied access by applications) if at least one but not all of the attributes' values deny authentication; this criteria is only reported when all four attribute values deny authentication.

-l Scans all users or the users specified by the *User* parameter to determine if the users can access the system.

-n Reports errors but does not fix them.

-p Fixes errors but does not report them.

-t Reports errors and asks if they should be fixed.

-y Fixes errors and reports them.

Exit Status

This command returns the following exit values:

Item Description

0 User definition files are appropriate.

>0 An error occurred or there is an error in one or more user definition files. The following error codes are returned:

EINVAL (22)

Invalid command-line arguments

ENOENT (2)

One or more user definition files do not exist

ENOTRUST (114)

Errors in user definitions in the database files or users unable to access the system (found by **-l** option)

Security

Access Control: This command should grant execute (x) access to the root user and members of the security group. The command should be **setuid** to the root user and have the **trusted computing base** attribute.

Files Accessed:

Mode	File
r	/etc/passwd
r	/etc/security/user

Mode	File
rw	/etc/security/group
rw	/etc/group
rw	/etc/security/lastlog
rw	/etc/security/limits
rw	/etc/security/audit/config
rw	/etc/security/login.cfg

Auditing Events:

Event	Information
USER_Check	user, attribute-error, status

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To verify that all the users exist in the user database, and have any errors reported (but not fixed), enter:

```
usrck -n ALL
```

2. To delete from the user definitions those users who are not in the user database files, and have any errors reported, enter:

```
usrck -y ALL
```

3. To display the list of users who are unable to access the system, enter:

```
usrck -l ALL
```

4. To display the list of users who are unable to access the system, in a bit mask format, enter:

```
usrck -l -b ALL
```

Files

Item	Description
/usr/bin/usrck	Specifies the path of the usrck command.
etc/passwd	Contains basic user attributes.
/etc/security/user	Contains the extended attributes of users.
/etc/group	Contains basic group attributes.
/etc/security/group	Contains the extended attributes of groups.
/etc/security/lastlog	Contains the last login attributes for users.
/etc/security/limits	Contains the process resource limits of users.
/etc/security/audit/config	Contains audit system configuration information.
/etc/security/login.cfg	Contains configuration information.

usrprt Command

Purpose

Reports the security capabilities of users.

Syntax

usrprt [**-R** <load_module>] [**-C**] [**-a** | **-c** | **-f**] user_list

Description

The **usrprt** command reports security capability information of users such as privileged commands executable by them, privileged files that can be accessed, and also the authorizations associated with the user.

Either of **-a**, **-c**, **-f** flags can be specified. When the **-a** option is specified, the list of authorizations associated with the user is displayed. When the **-c** option is specified, the privileged commands present in the `/etc/security/privcmds` database that can be executed by that user is listed. When the **-f** option is specified, the list of privileged files present in the `/etc/security/privfiles` database that can be accessed by the authorized user is listed.

The command takes a list of **comma** separated user names as input. When no option is specified, all the capability information such as authorizations, commands and privileged files information associated with the user is listed.

Flags

Item	Description
-a	Specify that a report of authorizations associated with the users is to be obtained.
-c	Specify that a report of privileged commands executable by the users is to be obtained.
-f	Specify that a report of privileged files accessible by the user is to be obtained.
-R	Specifies the loadable module to obtain the report of authorization capabilities from.
-C	Displays the authorization attributes in colon-separated records, as follows: #user:attribute1:attribute2: ... user1:value1:value2: ... user2:value1:value2: ...

Exit status

Item	Description
0	Successful completion.
>0	An error occurred.

Security

Access Control: This command should grant execute (x) access to the root user.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations associated with this command, see the [lssecattr](#) Command or the [getcmdattr](#) Subcommand.

Examples

1. To report the commands associated with user Bob:

```
usrprt -c Bob
```

2. To report all capabilities of user Simon:

```
usrprt Simon
```

3. To report all capabilities of user Simon in colon separated format

```
usrprt -C Simon
```

Information similar to the following appears:

```
#user:authorizations:commands:privfiles  
Simon:aix.security.user:/usr/bin/mkuser,/usr/bin/chuser:/etc/csh.cshrc,/etc/csh.login
```

Files

```
/etc/security/roles  
/etc/security/authorizations  
/etc/security/privcmds  
/etc/security/privfiles
```

utmpd Daemon

Purpose

Monitors and maintains **/etc/utmp** file.

Syntax

```
/usr/sbin/utmpd [ Interval ]
```

Description

The **utmpd** daemon monitors the **/etc/utmp** file for validity of the user process entries at regular intervals. An user process that has been terminated, but has not been cleaned up in the **/etc/utmp** file, is removed by cross checking the process id of the entry against the process table.

The Interval parameter specifies the amount of time in seconds between each scan of the **/etc/utmp** file. The default interval time would be 300 seconds.

Usage

To start **utmpd** from **/etc/inittab**, add the following entry to the file:

```
utmpd:2:respawn:/usr/sbin/utmpd
```

init starts the **utmpd** daemon during system startup. To have the changes take effects immediately without rebooting, type:

```
telinit q
```

Security

Only the root user can read and execute this command.

Files

Item	Description
<code>/etc/inittab</code>	Specifies stanzas read by the init command.
<code>/etc/utmp</code>	Contains a record of users logged into the system.

uucheck Command

Purpose

Checks for files and directories required by BNU.

Syntax

```
uucheck [ -v ] [ -x DebugLevel ]
```

Description

The **uucheck** command verifies the presence of the files and directories required by the Basic Networking Utilities (BNU) facility. The command also checks for some errors in the **/etc/uucp/Permissions** file.

Note: The **uucheck** command does not check for correct file and directory modes or for errors in the **/etc/uucp/Permissions** file, such as duplicate login or machine names.

Issue the **uucheck** command from the command line after installing the BNU program, configuring the BNU facility for your site, or making changes in part of the BNU facility, such as the **/etc/uucp/Permissions** file.

Note: Only someone with root user authority can use the **uucheck** command at the command line.

Flags

Item	Description
-v	Displays a detailed explanation of how BNU interprets the /etc/uucp/Permissions file.
-x <i>DebugLevel</i>	Displays debugging information. The valid range for the <i>DebugLevel</i> variable is 0 to 9, with a default of 5. The higher the number, the more detailed the information.

Examples

1. To find out how the BNU programs interpret the **/etc/uucp/Permissions** file, enter:

```
uucheck -v
```

The **-v** flag instructs the **uucheck** command to verify that the BNU files exist and displays a detailed explanation of how the BNU programs interpret the **/etc/uucp/Permissions** file. The output is similar to the following:

```
*** uucheck: Check Required Files and Directories
*** uucheck: Directories Check Complete

*** uucheck: Check /etc/uucp/Permissions file
** LOGNAME PHASE (when they call us)
```

```

When a system logs in as: (unostro)
  We DO allow them to request files.
  We WILL send files queued for them on this call.
  They can send files to
  /
  They can request files from
  /
  Myname for the conversation will be plague.austin..
  PUBDIR for the conversation will be
  /var/spool/uucppublic.

** MACHINE PHASE (when we call or execute their uux requests)

When we call system(s): (nostromo)
  We DO allow them to request files.
  They can send files to
  /
  They can request files from
  /
  Myname for the conversation will be plague.austin..
  PUBDIR for the conversation will be
  /var/spool/uucppublic.

Machine(s): (nostromo)
CAN execute the following commands:
command (ALL), fullname (ALL)

*** uuccheck: /etc/uucp/Permissions Check Complete

```

For an explanation of these permissions, see the **/etc/uucp/Permissions** file.

2. To debug with the **uuccheck** command, enter:

```
uuccheck -x8
```

The -x8 flag produces extensive debugging output.

Files

Item	Description
/etc/uucp/etc/uucp/Permissions	Describes access permissions for remote systems.
/etc/uucp/Systems	Describes accessible remote systems.

uucico Daemon

Purpose

Transfers Basic Networking Utilities (BNU) command, data, and execute files to remote systems.

Syntax

```
uucico [ -r RoleNumber ] [ -x DebugLevel ] -s SystemName
```

Description

The **uucico** daemon transfers Basic Networking Utilities (BNU) command (**C.***), data (**D.***), and execute (**E.***) files, created by the **uucp** and **uux** commands, to a specified remote system. Both the local and remote systems run the **uucico** daemon, and the two daemons communicate with each other to complete transfer requests.

The **uucico** daemon performs the following actions:

1. Scans the spooling directory (**/var/spool/uucp/SystemName**) on the local system for transfer requests.

2. Selects the device used for the communications connection after checking the /etc/uucp/Devices file and the lock files in the /etc/locks directory.
3. Places a call to the specified remote system using information in the Systems, Dialers, and Dialcodes files located in the /etc/uucp directory.
4. Performs the required login sequence specified in the Systems file.
5. Checks permissions listed in the /etc/uucp/Permissions file.
6. Checks scheduling limits in the Maxuuscheds and Maxuuxqts files located in the /etc/uucp directory.
7. Runs all transfer requests from both the local and the remote system, placing the transferred files in the public directories (/var/spool/uucppublic/*).
8. Logs transfer requests and completions in files in the /var/spool/uucp/.Log/uucico directory.
9. Notifies specified users of transfer requests.

Usually the **uucico** daemon is called by the **uucp** and **uux** commands when needed and is started periodically by the BNU scheduling daemon, **uusched**, which is started by the **cron** daemon.

The **uucico** daemon can be started from the command line for debugging. The BNU **uutry**, **Uutry**, and **uukick** commands also start the **uucico** daemon with debugging turned on.

Requirement: Either you must be in the /usr/sbin/uucp directory when you call the **uucico** daemon, or you must call the daemon with the full path name, /usr/sbin/uucp/uucico.

Tip: In the case of a **uux** command request for the execution of a command on a remote system, the **uucico** daemon transfers the files and the **uuxqt** daemon executes the command on the remote system.

Flags

Item	Description
-r <i>RoleNumber</i>	Specifies the server and client relationship. The role numbers are 1 for server mode and 0 for client mode. If the -r flag is not used, the uucico daemon is started in client mode (-r 0), because the uucico daemon is generally started automatically by a BNU command or daemon. When the uucico daemon is started manually, this flag should be set to 1.
-x <i>DebugLevel</i>	Displays debugging information on the screen of the local terminal. The valid range for the <i>DebugLevel</i> variable is 0 to 9, with a default of 5. Higher numbers cause the information to be more detailed. This flag is useful for diagnosing problems with the expect-send sequence in the <u>/etc/uucp/Systems</u> file.
-s <i>SystemName</i>	Specifies the name of the remote system. This flag is required when starting the uucico daemon from the command line. The <i>SystemName</i> variable is supplied internally when the uucico daemon is started automatically.

Note: System names must contain only ASCII characters.

Example

To call the **uucico** daemon from the command line, enter:

```
/usr/sbin/uucp/uucico -r 1 -s hera &
```

to start the daemon as a background process and contact remote system hera.

Files

Item	Description
<u>/etc/locks</u> /*	Contains lock files which prevent multiple uses of devices and multiple calls to systems.
<u>/usr/sbin/uucp</u> /*	Contains the uucico daemon and the configuration files for BNU.
<u>/etc/uucp/Devices</u>	Contains information about available devices.
<u>/etc/uucp/Dialcodes</u>	Contains dialing code abbreviations.
<u>/etc/uucp/Dialers</u>	Specifies initial handshaking on a connection.
<u>/etc/uucp/Maxuuscheds</u>	Limits scheduled jobs.
<u>/etc/uucp/Maxuuxqts</u>	Limits remote command executions.
<u>/etc/uucp/Permissions</u>	Describes access permissions for remote systems.
<u>/etc/uucp/Systems</u>	Describes accessible remote systems.
<u>/var/spool/uucp/.Admin/errors</u>	Lists uucico daemon errors that BNU cannot correct.
<u>/var/spool/uucp/.Log/uucico</u> /*	Contains uucico daemon log files.
<u>/var/spool/uucp/.Status/SystemName</u>	Lists the last time a remote system was contacted and the minimum time until the next retry.
<u>/var/spool/uucp/SystemName</u> /*	Contains C.* , D.* , and X.* files to be transferred by the uucico daemon.
<u>/var/spool/uucp/SystemName/C.*</u>	Contains command files.
<u>/var/spool/uucp/SystemName/D.*</u>	Contains data files.
<u>/var/spool/uucp/SystemName/X.*</u>	Contains execute files.
<u>/var/spool/uucppublic</u> /*	Contain files after transfer by the uucico daemon.

uuclean Command

Purpose

Removes files from the BNU spool directory.

Syntax

```
/usr/sbin/uucp/uuclean [ -m ] [ -nHours ] [ -pPrefix ] [ -dSubdirectory ]
```

Description

The **uuclean** command checks the Basic Networking Utilities (BNU) spool directory (/var/spool/uucp) for files with the specified prefixes and deletes those that are older than the given number of hours. If the **-nHours** flag is not included, the **uuclean** command deletes files that are older than 72 hours.

If the **-p** flag is not included, the **uuclean** command deletes all files in the specified subdirectories of the spool directory that meet the age requirement. If the **-d** flag is not included, the command deletes all the files (that meet the age and prefix requirements) in all the subdirectories of the spool directory. Thus if neither the **-d** or the **-p** flag is included, the **uuclean** command deletes *all* files in *all* subdirectories of the **/var/spool/uucp** directory that meet the age requirement.

If the **-m** flag is not specified, the **uuclean** command sends mail to owners of all command (**C.***) files that it deletes. If the **-m** flag is specified, the command sends mail to the owner of each file it deletes, including data (**D.***) and execute (**X.***) files. The mail message includes the name of the deleted file.

The **uuclean** command is usually run by the **cron** daemon.

Note: Only someone with root user authority or who is logged in as **uucp** can issue the **uuclean** command.

Flags

Item	Description
-dSubdirectory	Deletes files from the specified subdirectory of the /var/spool/uucp directory if they match specifications given with the -n and -p flags. If the -d flag is not specified, the uuclean command checks all subdirectories of the /var/spool/uucp directory. Up to 10 subdirectories can be specified with the -d flag.
-m	Instructs the uuclean command to send mail to the owner of each file when it is deleted.
-nHours	Deletes files whose ages are more than the number of hours specified by the <i>Hours</i> variable, if they match specifications given with the -d and -p flags. The default is 72 hours.
-pPrefix	Deletes files with the prefix given by the <i>Prefix</i> variable, if they match specifications given with the -n and -d flags. Up to 10 prefixes can be specified with the -p flag.

Examples

1. To delete all old command files, enter:

```
/usr/sbin/uucp/uuclean -pC
```

This command deletes all files in all subdirectories of the **/var/spool/uucp** directory whose names begin with C and that are older than 72 hours (the default). The system sends mail to the original owner of each file, stating that the file has been deleted.

2. To delete all old files from the spool directory for systems *venus* and *nostromo*, enter:

```
/usr/sbin/uucp/uuclean -n84 -dvenus -dnostromo
```

This command deletes all files in the **/var/spool/uucp/venus** and **/var/spool/uucp/nostromo** directories that are older than 84 hours. By default, the system notifies owners of **C.*** files that the files have been deleted; however, it does not notify owners of other files it deletes.

3. To delete all old files from all spool directories and notify users that they have been deleted, enter:

```
/usr/sbin/uucp/uuclean -m
```

This command deletes all files in all subdirectories of the spool directory, if the files are older than 72 hours (the default). It sends mail to the owner of each file it deletes.

4. To schedule the **uuclean** command to be started periodically by the **cron** daemon, add an entry similar to the following to your **/var/spool/cron/crontabs/uucp** file:

```
15 22 * * * /usr/sbin/uucp/uuclean -n96 -pC -pD -pX
```

This entry will cause the **cron** daemon to start the **uuclean** command at 22:15 (10:15 p.m.) daily. The **uuclean** command will delete all command (**C.***), data (**D.***), and execute (**X.***) files that are older than 96 hours from all subdirectories of the spool directory.

Files

Item	Description
/usr/sbin/uucp/uuclean	Contains the uuclean command.
/var/spool/uucp/*	Contains spooling files removed by the uuclean command.
/var/spool/cron/crontabs/uucp	Schedules uucp jobs for the cron daemon.

uucleanup Command

Purpose

Deletes selected files from the Basic Networking Utilities (BNU) spooling directory.

Syntax

uucleanup [**-C***Days*] [**-W***Days*] [**-m***String*] [**-D***Days*] [**-T***Days*] [**-X***Days*] [**-o** *Days*] [**-s***SystemName*]

Description

The Basic Networking Utilities (BNU) **uucleanup** command scans the spooling directory (**/var/spool/uucp**) for files that are older than a specified number of days and removes them. The **uucleanup** command performs the following tasks:

- Informs the requester of send and receive requests for systems that cannot be reached.
- Warns users about requests that have been waiting for a given number of days. The default is 1 day.
- Returns to the sender mail that cannot be delivered.
- Removes from the spool directory all other files older than a specified number of days.

Requirements:

- Only someone with root user privileges can issue the **uucleanup** command from the command line. The **uucleanup** command is not usually entered on the command line but is executed by the **uudemon.cleanu** command, a shell procedure.
- When BNU is installed, automatic cleanup is not enabled. Edit the **/var/spool/cron/crontabs/uucp** file and remove the comment character (**#**) from the beginning of the **uudemon.cleanu** line to instruct the **cron** daemon to start the **uudemon.cleanu** command.

Flags

Item	Description
-C <i>Days</i>	Removes C.* (command) files as old as, or older than, the number of days specified by the <i>Days</i> variable, and notifies the requester that the files have been deleted. The default time is 7 days.
-D <i>Days</i>	Removes D.* (data) files as old as, or older than, the number of days specified by the <i>Days</i> variable. Also attempts to deliver any remaining mail messages. The default time is 7 days.

Item	Description
-mString	Includes a specified line of text in the warning message generated by the -WDays option. The default line is See your local administrator to locate the problem.
-oDays	Removes other files as old as, or older than, the number of days specified by the <i>Days</i> variable. The default time is 2 days.
-sSystemName	Executes the uucleanup command only on the spooling directory specified by the <i>System</i> variable. The default is to clean up all BNU spooling directories. Restriction: System names can contain only ASCII characters.
-TDays	Removes TM.* (temporary) files as old as, or older than, the number of days specified by the <i>Days</i> variable. Also attempts to deliver any remaining mail messages. The default time is 7 days.
-WDays	Sends an electronic mail message to the requester warning that C.* (command) files as old as, or older than, the number of days specified by the <i>Days</i> variable are still in the spooling directory. The message includes the job ID and, if the request included mail, the mail message. The administrator can use the -m option to include a message line telling whom to call to check the problem. The default time is 1 day.
-XDays	Removes any X.* (execute) files as old as, or older than, the number of days specified by the <i>Days</i> variable. The default time is 2 days.

Examples

Warning Users That Their Command Files Have Not Been Sent

1. To send a warning for **C.*** (command) files 2 or more days old, enter:

```
uucleanup -W2
```

This warns the requester that the files have not been sent.

2. To send a message with the warning, enter:

```
uucleanup -m"Check these files waiting in the BNU job queue."
```

This locates **C.*** (command) files 1 or more days old (default), warns requesters that their files have not been sent, and gives the message: Check these files waiting in the BNU job queue.

Cleaning Up Command, Data, Execute, and Miscellaneous Files

1. To clean up command files 5 or more days old, enter:

```
uucleanup -C5
```

This removes all **C.*** (command) files 5 or more days old and sends an appropriate message to the requesters.

2. To clean up data and execute files 3 or more days old, enter:

```
uucleanup -D3 -X3
```

This removes all **D.*** (data) files and all **X.*** (execute) files 3 or more days old.

3. To clean up all files at once using defaults, enter:

```
uucleanup
```

This removes all **C.***, **D.***, **T.***, and **X.*** files, and all other files older than the default times.

Important: Whenever the **-C** and **-W** flags are used together, make sure the value specified for the **-W** flag is less than that for the **-C** flag. Otherwise, the **-C** flag will delete all the **C.*** (command) files before any warnings can be printed.

Cleaning Up Files for a Specific System

To delete files for one system, enter:

```
uucleanup -shera
```

This removes all files using defaults for system herera, but does not remove any files for any other systems.

Files

Item	Description
<code>/usr/sbin/uucp/*</code>	Contains the uudemon.cleanu shell procedure and all the configuration files for BNU.
<code>/var/spool/cron/crontabs/uucp</code>	Schedules BNU jobs for the cron daemon, including the uudemon.cleanu shell procedure.
<code>/var/spool/uucp/*</code>	Contain files removed by the uucleanup command.

uucp Command

Purpose

Copies files from one system to another.

Syntax

```
uucp [ -c | -C ] [ -d | -f ] [ -gGrade ] [ -j ] [ -m ] [ -nUser ] [ -r ] [ -sFile ] [ -xDebugLevel ] SourceFile ... DestinationFile ...
```

Description

The **uucp** command is a Basic Networking Utilities (BNU) command that copies one or more source files from one system to one or more destination files on another UNIX system. Files can be copied within a local system, between a local and a remote system, and between two remote systems.

The **uucp** command accomplishes the file transfer in two steps: first, by creating a command (**C.***) file in the spooling directory on the local computer and then by calling the **uucico** daemon to send the request to the specified computer. Command files include information such as the full path name of the source and destination files and the sender's login name. The full path name of a command file is a form of the following:

```
/var/spool/uucp/SystemName/C.SystemNameNxxxxx
```

where *N* is the grade of the request and *xxxx* is the hexadecimal sequence number used by BNU.

If the **uucp** command is used with the **-C** flag to copy the files to the spool directory for transfer, the **uucp** command creates not only a command file, but also a data (**D.***) file that contains the actual source file. The full path name of a data file is a form of the following:

```
/var/spool/uucp/SystemName/D.SystemNamexxxxx:###
```

Once the command files (and data files, if necessary) are created, the **uucp** command then calls the **uucico** daemon, which in turn attempts to contact the remote computer to deliver the files.

It is useful to issue the **uuname** command to determine the exact name of the remote system before issuing the **uucp** command. The **uulog** command provides information about **uucp** activities with another system.

Source and Destination File Names

File names and system names can contain only ASCII characters. Each can either be a path name on the local system or have the following form:

SystemName!PathName

where *SystemName* is taken from a list of system names that BNU knows about.

The destination *SystemName* can also be a list of names, such as the following:

SystemName!SystemName! . . . ! SystemName!PathName

In this case, an attempt is made to send the file using the specified route to the destination. Make sure that intermediate nodes in this route are willing to forward information, and that they actually talk to the next system.

The shell pattern-matching characters ? (question mark), * (asterisk), and [. . .] (brackets and ellipsis) can be used in the path names of the source file; the appropriate system expands them. The shell pattern-matching characters should not be used in the path name of the destination file.

If the *DestinationFile* is a directory rather than a file, the **uucp** command uses the last part of the *SourceFile* name to name the transferred file on the remote system.

Path Names

Path names for the *SourceFile* and *DestinationFile* parameters contain only ASCII characters. Paths for the source file can be one of the following:

- A full path name
- A relative path name

Paths for the *DestinationFile* parameter can be in the forms for the *SourceFile* parameter, or can be one of the following:

- A path name preceded by *~User* (for example, *~jkimble*) where *User* is a login name on the remote system. The specified user's login directory is then considered the destination of the transfer. If the user specifies an invalid login name, the files are transferred to the public directory, **/var/spool/uucppublic**, which is the default.
- A path name preceded by *~/Destination*, where *Destination* is appended to **/var/spool/uucppublic**. The destination is treated as a file name unless more than one file is being transferred by the request, the destination already exists as a directory on the remote system, or the destination is specified as a directory.

To specify the destination as a directory, follow the destination name with a / (slash). For example, *~/amy/* as the destination creates the directory **/var/spool/uucppublic/amy**, if it does not already exist, and puts the requested files in that directory.

Permissions

- The system administrator should restrict the access to local files by users on other systems.
- When transmitting files, the **uucp** command preserves execute permissions and grants read and write permissions to the owner, the group, and all others. (The **uucp** command owns the file.)
- Sending files to arbitrary *DestinationFile* path names on other systems or getting files from arbitrary *SourceFile* path names on other systems often fails because of security restrictions. The files specified in the path name must give read or write permission not only for the same group of users but also for any group.
- Protected files and files in protected directories owned by the requestor can be sent by the **uucp** command.

Flags

Item	Description
-c	Prevents files from being copied. This flag is the default and should not be used with the -C flag. If both flags are specified, the -c flag is overridden.
-C	Copies local files to the spool directory for transfer. Depending on the configuration of the Poll and Systems files and on how often the uusched daemon is run, the files may be transferred immediately on demand polling or in the future. Occasionally, problems occur while transferring a source file; for example, the remote computer may not be working or the login attempt may fail. In such a case, the file remains in the spool directory until it is either transferred successfully or removed by a cleanup command. This flag counteracts the -c flag.
-d	Creates any intermediate directories needed to copy the source files to the destination files on a remote system. Instead of first creating a directory and then copying files to it, the uucp command can be entered with the destination path name, and the BNU Program will create the required directory. This flag is the default and cannot be used with the -f flag.
-f	Does not create intermediate directories during the file transfer. This flag is used if the destination directory already exists and you do not want BNU to write over it. This command counteracts the -d flag.
-gGrade	Specifies when the files are to be transmitted during a particular connection. The <i>Grade</i> variable is a single number (0 to 9) or letter (A to Z, a to z); lower ASCII-sequence characters cause the files to be transmitted earlier than do higher sequence characters. The number 0 is the highest (earliest) grade; z is the lowest (latest) grade. The default is N .
-j	Displays the job identification number of the transfer operation on standard output. This job ID can be used with the uustat or uuq command to obtain the status of a particular job or with the uustat -k command or uuq -d command to terminate the transfer before it is completed.
-m	Sends a mail message to the requester when the source file is successfully copied to the destination file on a remote system. The message is sent to the requester's mailbox, /var/spool/mail/User. The mail command does not send a message for a local transfer. The -m flag works only when sending files or receiving a single file. It does not work when forwarding files.
-nUser	Notifies the recipient on the remote system identified by the <i>User</i> entry that a file has been sent. The mail system does not send a message for a local transfer. User names can contain only ASCII characters. Receiving multiple files specified by the shell pattern-matching characters ? (question mark), * (asterisk), and [. . .] (brackets and ellipses) does not activate the -n option.
-r	Prevents the starting of the uucico file transfer daemon, even if the command was issued at a time when calls to the remote system are permitted. (By default, a call to the remote system is attempted if the command is issued during a time period specified in the Poll and Systems files.) The -r option is useful for debugging.
-sFile	Reports the status of the transfer to the specified file. In this case, the <i>File</i> variable must designate a full path name.
-xDebugLevel	Displays debugging information on the screen of the local system. The <i>DebugLevel</i> variable is a number from 0 to 9. The higher the number, the more detailed the report.

Examples

1. To copy a file from the local system to a remote system, enter:

```
uucp /home/geo/f1 hera!/home/geo/f1
```

In this example, the `f1` file from the local system is copied to remote system `hera`.

2. To copy a file from the remote system and place it in the public directory, enter:

```
uucp hera!geo/f2 /var/spool/uucppublic/f2
```

In this example, the `f2` file from remote system `hera` is copied and placed in the public directory.

3. To copy a file from the remote system and place it in a directory other than the public directory, enter:

```
uucp hera!geo/f2 /home/geo/f2
```

In this example, the `f2` file from the remote system `hera` is copied to the `/home/geo/f2` directory. The `geo` login directory must allow write permission to members of the other group, for example, with mode `777`.

Files

Item

[/usr/bin/uucp](#)

[/etc/uucp/Poll](#)

[/etc/uucp/Systems](#)

[/etc/uucp/Sysfiles](#)

[/var/spool/uucp](#)

[/var/spool/uucppublic](#)

[/var/spool/uucppublic/SystemName/C.*](#)

[/var/spool/uucppublic/SystemName/D.*](#)

Description

Contains the **uucp** command.

File listing times when remote systems are automatically called (polled).

File describing accessible remote systems.

Specifies alternate files to be used as **Systems** files.

Spooling directory containing BNU status information.

Public directory containing files awaiting transfer by the **uucico** daemon.

Contains command files.

Contains data files.

uucpadm Command

Purpose

Enters basic BNU configuration information.

Syntax

uucpadm

Description

The **uucpadm** command provides interactive entry and modification of basic BNU configuration information in the **Devices**, **Systems**, **Permissions**, **Poll**, and **Dialcodes** files in the `/etc/uucp` directory. You can use the **uucpadm** command repeatedly to adjust the same file.

When you enter the **uucpadm** command at the command line, the command displays a list of the files you can change. After you choose a file to modify, the command displays a vertical list of the names of the fields in that file. You can enter the appropriate entry in each field. When you press the Enter key, the cursor moves to the next field in the list.

The command uses a copy of a file to record changes. The original file remains unchanged until you press the Ctrl+U or Ctrl+X key sequence at the appropriate menu. You can exit to the main **uucpadm** menu at any time, without saving your changes, by using the Ctrl+D key sequence.

The help routine provides instructions for each data field. Type a ? (question mark) in any menu field to access the help routine for that field.

Type a ~ (tilde) in any field to enter an ASCII editor and edit the appropriate file for that field. The **uucpadm** command invokes the editor designated by the **EDITOR** environment variable. If the **EDITOR** variable is not defined, the command invokes the **vi** editor.

If your entry for the first menu item matches an existing record, the **uucpadm** command retrieves that record for update. The command also tells you how many records have that first entry. If your entry for the first menu item does not match any existing record, the **uucpadm** command displays the word ADD at the top of the screen.

The **uucpadm** command checks the data as you enter it. If an inconsistency among the files is found, the command displays a warning message.

If the **uucpadm** command recognizes the entry you make for the first menu item, it fills in the default values for the remaining fields. For example, if you type TCP as the Type in the **Devices** file menu, the command places a - (hyphen) in each remaining field for you. It also checks for consistency with other files and for processes that should be running on the system. For example, when you type TCP as the Type in the **Devices** file menu, the **uucpadm** command checks to see if the **uucpd** daemon is running. If the daemon is not running, the command displays a note after the **Type** field, as follows:

```
Type: TCP
      <Note: Make certain uucpd is enabled.>
Line1: -
```

Note: The **uucpadm** command does not edit the **/etc/uucp/Dialers** file. Use an ASCII editor to edit this file.

Mode	File
rw	/etc/uucp/Devices
rw	/etc/uucp/Dialcodes
rw	/etc/uucp/Permissions
rw	/etc/uucp/Poll
rw	/etc/uucp/Systems

Examples

1. To start the **uucpadm** command, type the following:

```
/usr/sbin/uucp/uucpadm
```

A menu listing the files you can change is displayed.

2. To make an entry to the **/etc/uucp/Devices** file, choose the Add/Change Uucp Devices option at the **uucpadm** menu. The following is a sample **uucpadm** screen defining a direct 9600 baud connection to system merlin over the tty3 device:

```
Type: merlin
line1: tty3
line2: -
```

```
class: 9600
dialers: direct
```

3. To make an entry to the **/etc/uucp/Systems** file, choose the Add/Change Uucp Systems option at the **uucpadm** menu. The following is a sample **uucpadm** screen defining the `nostromo.aus.ibm.com` system connected to an ACU device in class 2400:

```
Name: nostromo.aus.ibm.com
Time: Any
Type: ACU
Class: 2400
Phone: 997-7942
Login: nuucp
Password: gotcha
```

4. To change the **/etc/uucp/Permissions** file, choose the Add/Change Uucp Permissions File option at the **uucpadm** menu.

- a. Following is a sample **uucpadm** screen defining a LOGNAME entry in the **Permissions** file:

```
L/M: LOGNAME=uucpz
Request: yes
Sendfiles: yes
Read: /
Write: NOWRITE=/etc
Callback:
Commands:
Validate: merlin:nostromo
```

If the remote machine is `merlin` or `nostromo`, the login ID must be `uucpz` (VALIDATE option). Remote hosts using this ID can request to send files, and the local host can sendfiles as requested. Users with this ID can read all files with permissions granted to the others group, and can write to all files, except those in the **/etc** directory, with permissions granted to the others group.

- b. Following is a sample **uucpadm** screen defining a MACHINE entry in the **Permissions** file:

```
L/M: MACHINE=merlin
Request: yes
Sendfiles:
Read: NOREAD=/etc
Write: NOWRITE=/etc
Callback:
Commands: ALL
Validate:
```

The machine ID is `merlin`. Requests for file transfers can be made. The user can read all files and can write to all files except those in the **/etc** directory. The execution of all commands is permitted.

5. To make an entry in the **/etc/uucp/Poll** file, choose the Add/Change Uucp Poll File option at the **uucpadm** menu. Following is a sample **uucpadm** screen defining an entry in the **Poll** file:

```
System: merlin
Hours: 0 7 13 19
```

This entry instructs BNU to poll the `merlin.aus.ibm.com` system at 2400 hours (midnight), 700 hours (7 a.m.), 1300 hours (1 p.m.), and 1900 hours (7 p.m.).

6. To make an entry in the **/etc/uucp/Dialcodes** file, choose the Add/Change Uucp Dialcodes option at the **uucpadm** menu. Following is a sample **uucpadm** screen defining an entry in the **Dialcodes** file:

```
Abr: LA
Dialcode: 1-213-
```

This entry assigns LA as the abbreviation for the Los Angeles area code.

Files

Item	Description
/usr/sbin/uucp/uucpadm	Contains the uucpadm command.
/etc/uucp/Devices	Contains information about available devices.
/etc/uucp/Dialcodes	Contains dialing code abbreviations.
/etc/uucp/Dialers	Specifies initial handshaking on a connection.
/etc/uucp/Permissions	Describes access permissions for remote systems.
/etc/uucp/Poll	Specifies when BNU polls remote systems to initiate tasks.
/etc/uucpSystems/	Describes accessible remote systems.

uucpd Daemon

Purpose

Handles communications between BNU and TCP/IP.

Syntax

The **uucpd** daemon cannot be started from the command line. It is started by the **inetd** daemon.

uucpd

Description

The **uucpd** daemon is an internal program that enables users of systems linked by the Basic Networking Utilities (BNU) program to establish a TCP/IP connection to other systems linked over a Token-Ring, Ethernet, or other network.

The **uucpd** daemon is a subserver of the **inetd** daemon. The **uucpd** daemon must be running as a background process on all the networked systems before the BNU program can use TCP/IP system to communicate. If the **uucpd** daemon is not running, [reconfigure the **inetd** daemon](#) to start the **uucpd** daemon. Use the **netstat** command to find out if the **uucpd** daemon is running.

Files

Item	Description
/etc/hosts	Contains the host name table used by TCP/IP.
/etc/inetd.conf	Contains the configuration of the inetd daemon.
/etc/services file	Defines socket assignments used by TCP/IP.
/usr/sbin /uucpd	Contains the uucpd daemon.
/etc/uucp/Devices	Contains information about available devices.
/etc/uucp/Permissions	Describes access permissions for remote systems.
/etc/uucp/Systems	Describes accessible remote systems.

uudecode Command

Purpose

Decodes a binary file that was used for transmission using electronic mail.

Syntax

uudecode [**-o** *OutputFile*] [*InFile*]

Description

The **uudecode** command reads an encoded file, strips off leading and trailing lines added by mailers, and recreates the original file with the specified mode and name. Decoding a file causes the result to be automatically saved to a file. The file name is identical to the remote file argument originally supplied to the **uuencode** command unless an output file name is specified with the **-o** flag.

Flags

Item	Description
-o <i>OutputFile</i>	Specifies the output file name that will be used instead of any pathname contained in the input data. You can direct the output of uudecode to standard output by specifying /dev/stdout as the <i>OutputFile</i> .

Parameters

Item	Description
<i>InFile</i>	Specifies the name of the file to decode.

Example

To decode the file `/tmp/con` on a local system that was encoded with the following command:

```
uuencode /usr/lib/boot/unix pigmy.goat > /tmp/con
```

enter:

```
uudecode /tmp/con
```

The file `pigmy.goat` will be identical to the originally encoded file `/usr/lib/boot/unix`.

Files

Item	Description
/usr/bin/uudecode	Contains the uudecode command.

uudemon.admin Command

Purpose

Provides periodic information on the status of BNU file transfers.

Syntax

uudemon.admin

Description

The **/usr/sbin/uucp/uudemon.admin** command is a shell procedure that mails status information about the Basic Networking Utilities (BNU) activities to the **uucp** login ID at intervals specified in the **/var/spool/cron/crontabs/uucp** file. The command executes both the **uustat -p** and the **uustat -q** commands:

- The **-p** flag instructs the **uustat** command to run the **ps -flp** command (process status, which generates a full, long list of specified process IDs) for all process ID (PID) numbers in the lock files.
- The **-q** flag lists the jobs currently queued to run on each system. These jobs either are waiting to execute or are in the process of executing. If a status file exists for the system, its date, time, and status information are reported.

Execute the **uudemon.admin** command at least once a day. The **uudemon.admin** command is not enabled when you install the BNU program. To run this command automatically, edit the **/var/spool/cron/crontabs/uucp** file, removing the comment character (#) from the beginning of the line that governs running the **uudemon.admin** command.

Examples

To run the **uudemon.admin** command automatically, edit the **/var/spool/cron/crontabs/uucp** file and remove the comment character (#) from the beginning of the **uudemon.admin** command line. Change:

```
#48 8,12,16 * * * /usr/bin/sh -c  
"/usr/sbin/uucp/uudemon.admin > /dev/null"
```

to:

```
48 8, 12, 16 * * * /usr/bin/sh -c "/usr/sbin/uucp/uudemon.admin > /dev/null"
```

The 48 notation represents minutes, the 8, 12, 16 notation represents hours based on the 24-hour clock, and the three asterisks (* * *) are placeholders representing the day of the month, the month of the year, and the day of the week, respectively. This line therefore instructs the **cron** daemon to run the **uudemon.admin** command daily at 48 minutes past the hours 0800, 1200, and 1600, that is, at 8:48 a.m., 12:48 p.m., and 4:48 p.m., respectively.

Note: These run intervals are defaults. By altering them, you can change the times at which the **cron** daemon executes the **uudemon.admin** command to fit the needs of your site.

Files

Item	Description
/usr/sbin/uucp/uudemon.admin	Contains the uudemon.admin command and the configuration files for BNU.
/etc/locks/*	Contains lock files which prevent multiple uses of devices and multiple calls to systems.
/var/spool/cron/crontabs/uucp	Schedules BNU jobs, including the uudemon.admin command, for the cron daemon.

uudemon.cleanu Command

Purpose

Cleans up BNU spooling directories and log files.

Syntax

uudemon.cleanu

Description

The **/usr/sbin/uucp/uudemon.cleanu** command is a shell script that cleans up the Basic Networking Utilities (BNU) spooling directories and log files. The command deletes files in the spooling directories that are as old as, or older than, a specified number of days, and then removes empty spooling directories.

The **uudemon.cleanu** command also updates archived log files by removing log information more than three days old. The command removes log files for individual computers from the **var/spool/uucp/.Log** directory, merges them, and places them in the **var/spool/uucp/.Old** directory, which contains old log information.

After performing the cleanup operations, the **uudemon.cleanu** command mails the **uucp** login ID a summary of the status information gathered during the current day.

Instruct the **cron** daemon to run the **uudemon.cleanu** command daily, weekly, or at longer intervals, depending on the amount of transactions the **uucico** and **uuxqt** daemons perform on the local system.

To run this command automatically, remove the comment character (#) at the beginning of the **uudemon.cleanu** command line in the **/var/spool/cron/crontabs/uucp** file.

Note: The **uudemon.cleanu** command is not usually entered on the command line but is instead executed by the **cron** daemon.

Example

To run the **uudemon.cleanu** procedure automatically, edit the **/var/spool/cron/crontabs/uucp** file and uncomment the **uudemon.cleanu** line. Change:

```
# 45 23 * * * /usr/bin/sh -c  
"/usr/sbin/uucp/uudemon.cleanu > /dev/null"
```

to:

```
45 23 * * * /usr/bin/sh -c "/usr/sbin/uucp/uudemon.cleanu > /dev/null"
```

The 45 notation represents minutes, the 23 notation represents hours based on the 24-hour clock, and the three asterisks (* * *) are placeholders representing the day of the month, the month of the year, and the day of the week, respectively. This line therefore instructs the **cron** daemon to run the **uudemon.cleanu** shell procedure at 45 minutes after hour 2300—that is, at 11:45 p.m.

Note:

1. These run intervals are defaults. By altering them, you can change the times at which the **cron** daemon executes the **uudemon.cleanu** command so that they fit the needs of your site.
2. The system allots the BNU program a specified amount of storage space for any one particular log file; the number of blocks is determined by the default **ulimit** value. If the **uudemon.cleanu** command fails to execute because the **ulimit** value is set too low for the requirements of the local system, delete the **uudemon.cleanu** command line (shown previously) from the **/var/spool/cron/crontabs/uucp** file and add the following entry to the **root** crontabs file, **/var/spool/cron/crontabs/root**:

```
45 23 * * * ulimit 5000; /usr/bin/su uucp  
-c "/usr/sbin/uucp/uudemon.cleanu > /dev/null"
```

Put the text on one line when entering it in the **root** crontabs file.

Files

Item	Description
/usr/sbin/uucp/uudemon.cleanu	Contains the uudemon.cleanu command.

Item	Description
<code>/var/spool/cron/crontabs/uucp</code>	Schedules BNU jobs, including the uudemon.cleanu command, for the cron daemon.
<code>/var/spool/cron/crontabs/root</code>	Schedules root user jobs for the cron daemon.
<code>/var/spool/uucp/.Log/*</code>	Contains the BNU program log files.

uudemon.hour Command

Purpose

Initiates file transport calls to remote systems using the BNU program.

Syntax

uudemon.hour

Description

The `/usr/sbin/uucp/uudemon.hour` command is a shell procedure used by the Basic Networking Utilities (BNU). In conjunction with the **Poll** file, the **uudemon.poll** command, and the `/var/spool/cron/crontabs/uucp` file, the **uudemon.hour** command initiates calls to remote systems.

The **uudemon.hour** command calls the following programs, which are involved in transferring files between systems at specified hourly intervals:

- The **uused** daemon first searches the spooling directory on the local system for command files that have not been transferred to the specified remote system, and then schedules the transfer of those files.
- The **uuxqt** daemon searches the spooling directory for execute files that have been transferred to the local system but have not yet been processed on that system.

Instruct the **cron** daemon to run the **uudemon.hour** command at specified hourly intervals. The frequency at which you run the **uudemon.hour** command depends on the amount of file-transfer activity originating from the local computer. If users on the local system initiate a large number of file transfers, you may need to specify that the **cron** daemon should start the **uudemon.hour** command several times an hour. If the number of file transfers originating from the local system is low, you can probably specify a start time once every 4 hours, for example.

To run the **uudemon.hour** command automatically, remove the comment character (**#**) from the beginning of the **uudemon.hour** command line in the `/var/spool/cron/crontabs/uucp` file.

Note: The **uudemon.hour** command is not usually entered on the command line, but is executed by the **cron** daemon.

Example

To run the **uudemon.hour** command automatically, edit the `/var/spool/cron/crontabs/uucp` file and remove the comment character (**#**) at beginning of the **uudemon.hour** command line. Change:

```
#25,55 * * * * /usr/bin/sh -c "/usr/sbin/uucp/uudemon.hour > /dev/null"
```

to:

```
25,55 * * * * /usr/bin/sh -c "/usr/sbin/uucp/uudemon.hour > /dev/null"
```

The 25,55 notation represents minutes, and the four asterisks (* * * *) are placeholders representing the hour of the day, the day of the month, the month of the year, and the day of the week, respectively. Therefore, this line instructs the **cron** daemon to run the **uudemon.hour** command at 25 minutes past the

hour and again at 55 minutes past the hour; for example, at 8:25 and 8:55 a.m., again at 9:25 and 9:55 a.m., and again every hour of every day.

Note:

1. These run intervals are defaults. By altering them, you can change the times at which the **cron** daemon executes the **uudemon.hour** command to fit the needs of your site. For example, to run the **uudemon.hour** command once every 4 hours, type the numeral 4 in the **time-interval** field.
2. If you change the run times for the **uudemon.hour** command, you should also change the run times for the **uudemon.poll** command so that it polls remote systems 5 to 10 minutes before the **uudemon.hour** command is run.

Files

Item	Description
/usr/sbin/uucp/uudemon.hour	Contains the uudemon.hour command.
/etc/uucp/Poll	Specifies when the BNU program should poll remote systems to initiate tasks.
/var/spool/cron/crontabs/uucp	Schedules BNU jobs, including the uudemon.hour and uudemon.poll commands, for the cron daemon.

uudemon.poll Command

Purpose

Polls the systems listed in the BNU **Poll** file.

Syntax

uudemon.poll

Description

The **/usr/sbin/uucp/uudemon.poll** command is a shell procedure used by the Basic Networking Utilities (BNU). In conjunction with the **/etc/uucp/Poll** file, the **uudemon.hour** command, and the **/var/spool/cron/crontabs/uucp** file, the **uudemon.poll** command initiates calls to remote systems.

The **uudemon.poll** command performs the following actions:

- Polls (contacts) the systems listed in the **Poll** file (**/etc/uucp/Poll**).
- Creates command (**C.***) files for the systems listed in the **Poll** file.

The time at which you run the **uudemon.poll** command depends on the time at which you run the **uudemon.hour** command. In general, schedule the polling shell procedure before the hourly procedure. This schedule enables the **uudemon.poll** command to create any required command files before the **cron** daemon runs the **uudemon.hour** command.

Instruct the **cron** daemon to run the **uudemon.poll** command about 5 to 10 minutes before running the **uudemon.hour** command. To run this procedure automatically, remove the comment character (#) from the beginning of the **uudemon.poll** command line in the **/var/spool/cron/crontabs/uucp** file.

Note: The **uudemon.poll** command is not usually entered on the command line, but is executed by the **cron** daemon.

Example

To run the **uudemon.poll** shell procedure automatically, edit the **/var/spool/cron/crontabs/uucp** file and remove the **#** (comment character) at the beginning of the line which starts the **uudemon.poll** command. Change:

```
#20,50 * * * * /usr/bin/sh -c "/usr/sbin/uucp/uudemon.poll > /dev/null"
```

to:

```
20,50 * * * * /usr/bin/sh -c "/usr/sbin/uucp/uudemon.poll > /dev/null"
```

The 20, 50 notation represents minutes, and the four asterisks (* * * *) are placeholders representing the hour of the day, the day of the month, the month of the year, and the day of the week, respectively. This line therefore instructs the **cron** daemon to run the **uudemon.poll** command at 20 minutes past the hour and again at 50 minutes past the hour—for example, at 8:20 and 8:50 a.m., and at 9:20 and 9:50 a.m.—every hour of every day.

Note: Change the times at which the **cron** daemon executes the **uudemon.poll** command to correspond to the times you set up for the **uudemon.hour** command. The defaults specified in the **/var/spool/cron/crontabs/uucp** file instruct the **cron** daemon to run the **uudemon.poll** command 5 minutes before running the **uudemon.hour** command.

Files

Item	Description
/usr/sbin/uucp/*	Contains the uudemon.poll and uudemon.hour commands and all the configuration files for BNU.
/etc/uucp/Poll	Specifies when the BNU program should poll remote systems to initiate tasks.
/var/spool/cron/crontabs/uucp	Schedules BNU jobs, including the uudemon.poll command, for the cron daemon.

uuencode Command

Purpose

Encodes a binary file for transmission using electronic mail.

Syntax

uuencode [**-m**] [*SourceFile*] *OutputFile*

Description

The **uuencode** command converts a binary file to ASCII data. This is useful before using BNU (or uucp) mail to send the file to a remote system. The **uudecode** command converts ASCII data created by the **uuencode** command back into its original binary form.

The **uuencode** command takes the named *SourceFile* (default standard input) and produces an encoded version on the standard output. The encoding uses only printable ASCII characters, and includes the mode of the file and the *OutputFile* filename used for recreation of the binary image on the remote system.

Use the **uudecode** command to decode the file.

Flags

Item	Description
-m	Encode the output using the MIME Base64 algorithm. If -m is not specified, the old uuencode algorithm will be used.

Parameters

Item	Description
<i>OutputFile</i>	Specifies the name of the decoded file. You can direct the output of the uuencode command to standard output by specifying /dev/stdout as the <i>OutputFile</i> .
<i>SourceFile</i>	Specifies the name of the binary file to convert. Default is standard input.

Examples

1. To encode the file `unix` on the local system and mail it to the user `jsmith` on another system called `mysys`, enter:

```
uuencode unix unix | mail jsmith@mysys
```

2. To encode the file `/usr/lib/boot/unix` on your local system with the name `pigmy.goat` in the file `/tmp/con`, enter:

```
uuencode /usr/lib/boot/unix pigmy.goat > /tmp/con
```

Files

Item	Description
/usr/bin/uuencode	Contains the uuencode command.

uuid_get command

Purpose

Generates Universal Unique Identifiers (UUIDs).

Syntax

```
uuid_get [ -n count ] [ -o outfile ] [ -c ]
```

Description

The **uuid_get** command generates UUIDs. By default, the **uuid_get** command generates a hexa-string representation of a UUID. You can use the **uuid_get** command along with the **-c** option to generate source-code representation of UUIDs.

Flags

- n count**
Generates number of UUIDs that is specified in the count parameter. The value of the **count** parameter must be greater than zero.

-o outfile

Redirects the generated UUID to an output file that is specified by using the **outfile** parameter.

-c

Generates a C programming source-code representation of a UUID.

Examples

1. To generate a hexa-string representation of a UUID, enter the following command:

```
#uuid_get
```

An output similar to the following example is displayed:

```
6ae84954-9ef6-11e6-8003-3a0ea8d2f402
```

2. To generate a C programming source-code representation of a UUID, enter the following command:

```
#uuid_get -c
```

An output similar to the following example is displayed:

```
{ 0xd966286a,
  0x9ef6,
  0x11e6,
  0x8004,
  {0x3a, 0x0e, 0xa8, 0xd2, 0xf4, 0x02} };
```

3. To generate 5 UUIDs by using a single command, enter the following command:

```
# ./uuid_gen -n 5
```

An output similar to the following example is displayed:

```
ba4dae20-f6d7-11e5-8007-3a0ea8d2f402
ba4daf56-f6d7-11e5-8007-3a0ea8d2f402
ba4dafa2-f6d7-11e5-8007-3a0ea8d2f402
ba4db06e-f6d7-11e5-8007-3a0ea8d2f402
ba4db0fa-f6d7-11e5-8007-3a0ea8d2f402
```

uukick Command

Purpose

Uses debugging mode to contact a specified remote system.

Syntax

```
uukick [ -xDebugLevel ] SystemName
```

Description

The **uukick** command contacts a remote system, named by the *SystemName* parameter, using debugging mode. The debugging mode provides a means of monitoring Basic Networking Utilities (BNU) file transfers and connections to remote computers.

The **uukick** command starts the **uucico** daemon, which actually contacts the specified remote system. The **uucico** daemon produces debugging output that enables you to monitor its progress as it establishes the connection to the remote system, performs the remote login, and transfers a file.

The debugging output is scrolled on the screen of the local system. Once the system has finished displaying this information, press the Interrupt key to return to the prompt.

Requirement: Either you must be in the `/usr/lib/uucp` directory when you issue the **uukick** command, or you must issue the command with the full path name, `/usr/sbin/uucp/uukick`.

Tip: The **uukick** command is a shell script stored in the `/usr/lib/uucp` directory.

Flags

Item	Description
<code>-xDebugLevel</code>	Overrides the default amount of detail in the debugging information the command displays on the screen. The valid range for the <code>DebugLevel</code> variable is 0 to 9, with a default of 5. Higher numbers cause the final report to be more detailed. If the <code>-x</code> flag is not used, the uucico daemon is started with the default level, which produces a moderate amount of information.

Example

To change the amount of detail in the information about the progress of the operation of the **uucico** daemon, use the `-x` flag to specify a higher or lower debugging level. For example, enter:

```
uukick -x9 hera
```

This instructs the **uukick** command to generate as much information as possible about the way in which the **uucico** daemon is working while trying to connect to system `hera`. Or, enter:

```
uukick -x3 hera
```

This instructs the command to generate less than the default amount of information about the connection.

Files

Item	Description
<code>/usr/sbin/uucp/uukick</code>	Contains the uukick shell script.
<code>/etc/uucp</code>	Contains the configuration files for BNU.
<code>/etc/uucp/Devices</code>	Contains information about available devices.
<code>/etc/uucp/Dialcodes</code>	Contains dialing code abbreviations.
<code>/etc/uucp/Dialers</code>	Specifies initial handshaking on a connection.
<code>/etc/uucp/Permissions</code>	Describes access permissions for remote systems.
<code>/etc/uucp/Systems</code>	Describes accessible remote systems.
<code>/var/spool/uucp/*</code>	Contain files to be transferred and files recording transfer statistics.
<code>/var/spool/uucppublic/*</code>	Contain files that have been transferred.

uulog Command

Purpose

Provides information about BNU file-transfer activities on a system.

Syntax

```
uulog [ -x ] [ -Number ] [ -fSystem | -sSystem ]
```

Description

The Basic Networking Utilities (BNU) **uulog** command displays the contents of the log files containing the activities of the **uucico** and **uuxqt** daemons. Individual log files are created for each remote system with which the local system uses the **uucp**, **uuto**, and **uux** commands to communicate.

Use the **uulog** command to display a summary of **uucp**, **uuto**, and **uux** command requests by the user or by the system. All of these transactions are logged in files in the **/var/spool/uucp/.Log** directory. The files are named *DaemonName/SystemName* where the *DaemonName* directory is named for the daemon involved and the *SystemName* file is named for the remote system the daemon is contacting.

The **uucp** and **uuto** commands call the **uucico** daemon. The **uucico** daemon's activities are logged in the *SystemName* file in the **/var/spool/uucp/.Log/uucico** directory.

The **uux** command calls the **uuxqt** daemon. The **uuxqt** activities are logged in the *SystemName* file in the **/var/spool/uucp/.Log/uuxqt** directory.

You can examine these individual log files by issuing the **uulog** command directly. However, you can also have the BNU program automatically append these temporary log files to a primary log file that you can then examine. This is called *compacting the log files* and is handled by the **uudemmon.cleanu** command, a shell script.

Flags

Item	Description
-fSystem	Issues a tail command with the -f flag on the file transfer log for the specified <i>System</i> variable, displaying the end of the log file. Press the Interrupt key to leave the file and return to the prompt.
-sSystem	Displays a summary of copy (uucico daemon) requests involving the specified system. Restrictions: • System names can contain only ASCII characters. • The -f and -s flags cannot be combined.
-x	Displays the uuxqt daemon log file for the given system.
-Number	Displays the last lines of the file. The number of lines is determined by the <i>Number</i> variable. (To display the lines, the uulog command issues a tail command with the -f flag for the specified number of lines.)

Examples

1. To display the **uucico** log file for system *hera*, enter:

```
uulog -shera
```

The output from the command is similar to the following:

```
uucp hera (10/30-10:18:38,3833,0) SUCCEEDED (call to hera)
uucp hera (10/30-10:18:39,3833,0) OK (startup)
jim hera hera661d (10/30-10:18:39,3833,0) REQUEST
(nostromo!D.hera661e6c9 --> hera!X.hera661d (jim))
jim hera hera661d (10/30-10:18:40,3833,0) FAILED (CAN'T
READ /var/spool/uucp/hera/D.hera661e6c9 13)
uucp hera (10/30-10:18:41,3833,0) OK (conversation
complete -8)
```

The preceding lines log a conversation between the local system (nostromo) and remote system *hera*. The conversation began at 10:18:38 (a.m.) on October 30th, and ended at 10:18:41. User *jim* attempted to transfer a data file, *D.hera661e6c9*, to system *hera*. The connection to *hera* was successful, but the file could not be transferred because BNU could not read it.

2. To display the **uuxqt** log file, enter:

```
uulog -x
```

3. To display the last forty lines of the file transfer log for system zeus, enter:

```
uulog -fzeus -40
```

Files

Item	Description
<code>/usr/bin/uulog</code>	Contains the uulog command.
<code>/var/spool/uucp/.Log</code>	Contain the BNU log files.

uuname Command

Purpose

Provides information about other systems accessible to the local system.

Syntax

```
uuname [ -c | -l ]
```

Description

The **uuname** command is a Basic Networking Utilities (BNU) command that displays a list of all the computers networked to the local system. This list of accessible systems is displayed on the screen of the local terminal.

In order for a local system to communicate with a remote system by way of BNU, the remote system must:

- Have a UNIX-based operating system.
- Be connected to the local system. (A telephone line can serve as the connection media.)

BNU can be used to communicate between a workstation and an operating system except UNIX, but such communications may require additional hardware or software. The remote systems accessible with BNU commands are identified when the BNU programs are installed and listed in a BNU **Systems** file (by default, the `/etc/uucp/Systems` file, or one or more files specified in the `/etc/uucp/Sysfiles` file).

Before copying a file to another system with the **uuto** or **uucp** command, issue the **uuname** command to determine the exact name of the remote system.

Flags

Item	Description

-c	Displays only the names of systems contained in the cu Systems files (configured by the <code>/etc/uucp/Sysfiles</code> file). Omission of this flag displays the names of systems contained in the uucico Systems files (also configured by the <code>/etc/uucp/Sysfiles</code> file). If <code>/etc/uucp/Sysfiles</code> is not used to separate cu and uucico configuration into separate Systems files, the names of all systems listed in <code>/etc/uucp/Systems</code> are displayed regardless of the -c flag.
-l	Displays the name of the local system.

Examples

1. To identify the remote systems connected to the local system, enter:

```
uuname
```

The system responds with a list similar to the following:

```
arthur  
hera  
merlin  
zeus
```

2. To identify the name of the local system, enter:

```
uuname -l
```

The system responds with something similar to the following:

```
nostromo
```

Files

Item	Description
/usr/bin/uuname	Contains the uuname command.
/etc/uucp/Systems	Lists accessible remote systems.
/etc/uucp/Sysfiles	Specifies alternate files to be used as Systems files.
/var/spool/uucp	Contains BNU administrative files.
/var/spool/uucppublic	Contains BNU files awaiting transfer (public directory).

uupick Command

Purpose

Completes the transfer of and handles files sent by the **uuto** command.

Syntax

```
uupick [ -sSystem ]
```

Description

The **uupick** command is a Basic Networking Utilities (BNU) command that completes the transfer and handles files that the BNU **uuto** command has transmitted to a designated user ID.

Once the copied file is in the receive directory, the **rmail** command notifies the recipient that the file has arrived. The recipient then issues the **uupick** command, which searches the public directory on the local system for files sent with some form of the following name:

/var/spool/uucppublic/receive/User/System/File

For each file or directory found, the **uupick** command displays the following message on the screen of the local system:

```
from System: [file File] [dir Directory]  
?
```

The question mark prompt (?) following the message indicates you can now enter one of the [file-handling options](#).

Flags

Item	Description
-s <i>System</i>	Searches /var/spool/uucppublic/receive/User/System for files sent from the specified system. System names contain only ASCII characters.

File-Handling Options

The question mark prompt (?) following a message indicates that one of the following file-handling options should be entered:

Option	Action
!Command	Escapes to a shell to run the specified command. After the command executes, the user is automatically returned to the uupick command.
*	Displays all the file-handling options.
a [<i>Directory</i>]	Moves all uuto files currently in the receive directory into a specified directory on the local system. The default is the current working directory. Use a full or relative path name to specify the destination directory.
Ctrl-D	Stops processing and exits from the uupick command.
d	Deletes the specified file.
m [<i>Directory</i>]	Moves the file to a specified directory. If the <i>Directory</i> variable is not specified as a complete path name, a destination relative to the current directory is assumed. If no destination is given, the default is the current working directory on the local system.
new-line	Moves to the next entry in the receive directory when the Enter key is pressed.
p	Displays the contents of the file on the workstation screen.
q	Stops processing and exits from the uupick command.

Examples

1. To receive a file sent with the **uuto** command and add it to the current working directory, enter:

```
uupick
```

The system responds with a message similar to:

```
from system anchor: file file1
?
```

Enter:

```
a
```

In this example, the **/usr/bin/file1** file sent with the **uuto** command from system anchor is added to the current working directory.

2. To receive a file sent with the **uuto** command and add it to a specified directory on your local system, enter:

```
uupick
```

The system responds with a message similar to:

```
from system anchor: file file2
?
```

Enter:

```
a /usr/bin1
```

In this example, the `/usr/bin/file2` file sent with the **uuto** command from system `anchor` is added to the `/usr/bin1` directory on the local system.

Note: The `a /usr/bin1` instruction means move *all* files, not just one. Thus, if any other files are in the `~/anchor/...` directory, they will also be moved.

3. To search for files sent from system `anchor`, enter:

```
uupick -s anchor
```

The system responds with a message similar to:

```
from system anchor: file file1
```

Files

Item	Description
<code>/usr/bin/uupick</code>	Contains the uupick command.
<code>/var/spool/uucppublic</code>	Contains the BNU public directory.

uupoll Command

Purpose

Forces a poll of a remote BNU system.

Syntax

```
uupoll [ -gGrade ] [ -n ] SystemName
```

Description

The **uupoll** command forces the Basic Networking Utilities (BNU) to poll the remote system specified by the *SystemName* parameter. The command is usually run by the **cron** daemon or by a user who wants to force a job to be executed immediately. Otherwise, remote systems are polled by the **uudemon.poll** command at times scheduled in the `/etc/uucp/Poll` file and the `/var/spool/cron/crontabs/uucp` file.

Normally, the **uucico** daemon contacts a remote system only at times specified in the **Poll** file or when there is a job queued for that system. The **uupoll** command queues a null job for the remote system and then invokes the **uucico** daemon. This forces the **uucico** daemon to contact the remote system immediately and attempt to send any jobs which are queued for that system. Use the **-g** flag to specify that only high priority jobs be sent.

Use the **-n** flag to queue the null job without starting the **uucico** daemon. Use this option to:

- Queue a null job before invoking the **uucico** daemon for debugging.
- Queue a null job just before the **uucico** daemon is usually invoked, thus forcing the daemon to poll the specified system.

The *SystemName* parameter is required, and specifies the name of the remote system to be polled.

Flags

Item	Description
-gGrade	Instructs the uupoll command to send only jobs of the given grade (specified by the <i>Grade</i> parameter) or higher on this call. Jobs of a lower grade will remain in the queue until the next time the remote system is polled.
-n	Queues the null job, but does not invoke the uucico daemon.

Examples

1. To run the **uupoll** command with the **cron** daemon, place an entry in your **crontabs** file similar to:

```
0 1,7,16 * * * /usr/bin/uupoll hera
```

This polls system hera at 0100 hours (1 a.m.), 0700 hours (7 a.m.), and 1600 hours (4 p.m.) daily.

2. If the local system already runs the **uucico** daemon at specific times, you may want to queue a null job just before the **uucico** daemon normally runs. For example, if your system runs the **uucico** daemon hourly, place an entry similar to the following in your **crontabs** file:

```
0 1,7,16 * * * /usr/bin/uupoll -n zeus
0 5,12,21 * * * /usr/bin/uupoll -n hera
5 * * * * /usr/sbin/uucp/uucico -r1
```

This queues null jobs for the remote sites on the hour, and they are processed by the **uucico** daemon when it runs at 5 minutes past the hour.

3. To force the **uucico** daemon to transfer all jobs of grade N or higher for system zeus:

```
uupoll -gN zeus
```

Files

Item	Description
/usr/bin/uupoll	Contains the uupoll command.
/etc/uucp/Poll	Specifies when the BNU program should poll remote systems to initiate tasks.
/var/spool/cron/crontabs/uucp	Schedules automatic polling of remote systems.
/var/spool/uucp/SystemName	Contain files to be transferred to remote systems.

uuq Command

Purpose

Displays the BNU job queue and deletes specified jobs from the queue.

Syntax

```
uuq [ -l | -h ] [ -sSystemName ] [ -uUser ] [ -dJobNumber ] [ -rSpoolDir ] [ -bBaudRate ]
```

Note: Only a user with root authority can use the **-d** flag.

Description

The **uuq** command is used to list or delete job entries in the Basic Networking Utilities (BNU) job queue.

When listing jobs, the **uuq** command uses a format similar to that used by the **ls** command. In the default format, the **uuq** command lists only the job numbers of the jobs waiting in the queue, followed by a summary line for each system.

In summary format (**uuq -h**) only the summary lines are listed. Summary lines give:

- System name
- Number of jobs for the system
- Total number of bytes to send

In the long format (**uuq -l**), which can be quite slow, the information listed for each job is:

- Job number
- Number of files to transfer
- User who sent the job
- Number of bytes to be sent
- Type of job requested:

Item	Description

S	Sending a file

R	Receiving a file

X	Executing a command on the remote system

- File to be sent or received or the command to be executed

A user with root authority can use the **-dJobNumber** flag to delete jobs from the queue after running a **uuq** listing to discover the job numbers.

Flags

Item	Description
-bBaudRate	Uses the baud rate given, instead of the default (1200 baud), to compute the transfer time.
-d JobNumber	Deletes the job designated by the <i>JobNumber</i> variable from the BNU queue. Only someone with root authority can delete jobs from the queue.
-h	Shows only the <u>summary lines</u> for each system.
-l	Lists the output in the <u>long format</u> .
-sSystemName	Lists only jobs for systems whose system names begin with the string specified in the <i>SystemName</i> variable.
-r SpoolDir	Searches for files in the spooling directory designated by the <i>SpoolDir</i> variable, instead of in the default spooling directory.
-uUser	Lists only jobs queued by users whose login names begin with the string specified in the <i>User</i> variable.

Examples

1. To get a long listing of all jobs spooled for system herc, type:

```
uuq -l -sherc
```

2. To get a summary listing for all systems, type:

```
uuq -h
```

3. To delete a job for user nita from the queue, first use the **uuq** command to find the number of the job you want to delete, as follows:

```
uuq -l -unita
```

This produces a list of jobs spooled for user nita. Find the job you wish to remove. If its job number is 13451, for example, the following command will delete the job:

```
uuq -d13451
```

Note: You must have root authority or be logged in as **uucp** to delete jobs from the queue.

Files

Item	Description
/usr/bin/uuq	Contains the uuq command.
/var/spool/uucp/<i>SystemName</i>	Contains spool files for the remote system designated by <i>SystemName</i> .
/var/spool/uucp/<i>SystemName</i>/C.*	Contain instructions for file transfers.
/var/spool/uucp/<i>SystemName</i>/D.*	Contain information about data files to be transferred.
/var/spool/uucp/<i>SystemName</i>/X.*	Contain instructions for executing remote commands.

uusched Daemon

Purpose

Schedules work for the Basic Networking Utilities (BNU) file transport program.

Syntax

```
uusched [ -uDebugLevel ] [ -xDebugLevel ]
```

Description

The **uusched** daemon schedules work for the Basic Networking Utilities (BNU) file transport program. It schedules the transfer of files that are queued in the **/var/spool/uucp/*SystemName*** directory. The scheduling daemon first randomizes the work and then starts the **uucico** daemon, which transfers the files.

The **uusched** daemon is usually started by the **uudemon.hour** command, a shell procedure, which is run periodically by the **cron** daemon based on instructions from the **/var/spool/cron/crontabs/uucp** file.

The **uusched** daemon can also be started from the command line for debugging purposes.

Note: Either you must be in the **/usr/sbin/uucp** directory when you start the **uusched** daemon, or you must start the daemon with the full path name, **/usr/sbin/uucp/uusched**.

Flags

Item	Description
-uDebugLevel	Passes as the -xDebugLevel flag to the uucico daemon. The <i>DebugLevel</i> variable is a number from 0 to 9, with a default of 5. Higher numbers give more detailed debugging information, which is displayed on the screen of the local system.

Item	Description
-xDebugLevel	Outputs debugging messages from the uusched daemon. The <i>DebugLevel</i> variable is a number from 0 to 9, with a default of 5. Higher numbers give more detailed debugging information, which is displayed on the screen of the local system.

Example

To start the **uusched** daemon from the command line, enter:

```
/usr/sbin/uucp/uusched &
```

This starts the **uusched** daemon as a background process. (Note that the path name is included in the command.)

Files

Item	Description
/etc/locks/*	Contains lock files that prevent multiple uses of devices and multiple calls to systems.
/usr/sbin/uucp/*	Contains the uusched daemon and the BNU configuration files.
/etc/uucp/Devices	Contains information about available devices.
/etc/uucp/Maxuuscheds	Limits scheduled jobs.
/etc/uucp/Systems	Describes accessible remote systems.
/var/spool/cron/crontabs/uucp	Schedules BNU jobs for the cron daemon, including the uudemon.hour shell procedure.
/var/spool/uucp/SystemName /*	Contain files waiting to be transferred.

uusend Command

Purpose

Sends a file to a remote host.

Syntax

```
uusend [ -mMode ] [ -r ] Sourcefile System [ !System ... ] ! RemoteFile
```

Description

The **uusend** command sends a file to a given location on a remote system. The remote system need not be directly connected to the local system, but a chain of UUCP links must connect the two systems, and the **uusend** command must be available on each system in the chain.

The chain of systems is given by the *System[!System ...]* parameter, which lists each remote system the file is to be transferred to, separated by ! (exclamation points). The *!RemoteFile* parameter gives the name under which the file is to be stored when it reaches the last system in the chain.

Note: Do not put any spaces between the system names and exclamation points or between the last exclamation point and the remote file name.

The *SourceFile* parameter specifies the name of the file on the local system. If a - (dash) is used, the **uusend** command uses standard input.

Flags

Item	Description
-m <i>Mode</i>	Specifies that the mode of the file on the remote system will be taken from the octal number given. If this flag is not specified, the mode of the input file will be used.
-r	Prevents the starting of the uucico daemon, which transfers files between systems. The default is to start the uucico daemon.

The flags are primarily used internally by the **uusend** command when it is transferring files to the next remote system in the chain.

Example

To send a file across one system to another system, enter:

```
uusend /etc/motd nostromo!gandalf!~nuucp
```

The `/etc/motd` file is sent to system `nostromo` and then to system `gandalf`, and placed in `nuucp`'s home directory, `/var/spool/uucppublic/nuucp`, where `nuucp` is a BNU login ID.

Files

Item	Description
/usr/bin/uusend	Contains the uusend command.

uusnap Command

Purpose

Displays the status of BNU contacts with remote systems.

Syntax

uusnap

Description

The **uusnap** command displays a table showing the status of the Basic Networking Utilities (BNU). The table includes the following information for each remote system:

Item	Description
SystemName	Specifies the name of the remote system.
Number Cmds	Specifies the number of command files (C.* files) queued for the remote system.
Number Data	Specifies the number of data transfers (D.* files) queued for the remote system.
Number Xqts	Specifies the number of remote command executions (X.* files) queued for the remote system.
Message	Specifies the current status message for the site, from the /var/spool/uucp/.Status/SystemName file. The Message field may include the time remaining before BNU can retry the remote system, and the count of the number of times (if any) BNU has tried unsuccessfully to reach the system.

Example

To see a snapshot of the status of BNU, enter:

```
uusnap
```

The output from this command is similar to the following:

```
nostromo  4 Cmds  2 Data  2 Xqts  SUCCESSFUL
zeus      2 Cmds  1 Data  2 Xqts  NO DEVICES AVAILABLE
```

These lines indicate that four command files, two data files, and two execute files are currently queued for system `nostromo`. The last connection to `nostromo` was successful. The last attempt to contact system `zeus`, on the other hand, was not successful because no device was available on the local system.

Files

Item	Description
<code>/usr/bin/uusnap</code>	Contains the uusnap command.
<code>/var/spool/uucp/.Status/SystemName</code>	Records the status of BNU contacts with a remote system.
<code>/var/spool/uucp/SystemName</code>	Contains C.* , D.* , and X.* files to be transferred by the uucico daemon.
<code>/var/spool/uucp/SystemName/C.*</code>	Instruct BNU about files to be transferred.
<code>/var/spool/uucp/SystemName/D.*</code>	Contain files to be transferred by BNU.
<code>/var/spool/uucp/SystemName/X.*</code>	Specify commands to be remotely executed by BNU.

uustat Command

Purpose

Reports the status of and provides limited control over BNU operations.

Syntax

```
uustat [ [ -n Number ] [ -a | -k JobID | -m | -p | -q | -r JobID ] [ -s System ] [ -u User ] ]
```

Description

The **uustat** command is a Basic Networking Utilities (BNU) command that displays status information about several types of BNU operations. It is particularly useful in monitoring the status of BNU requests.

In addition, the **uustat** command also gives a user limited control over BNU jobs queued to run on remote systems. By issuing the command with the appropriate flag, a user can check the general status of BNU connections to other systems and cancel copy requests made with the **uucp** and **uuto** commands.

If the **uustat** command is issued without any flags, the command reports the status of all BNU requests issued by the current user since the last time the holding queue was cleaned up. Such status reports are displayed in the following format:

```
jobid date/time status system_name user_ID size file
```

There are two types of BNU queues:

- The current queue, accessed with the **-q** flag, lists the BNU jobs either queued to run on or currently running on one or more specified computers.
- The holding queue, accessed with the **-a** flag, lists all jobs that have not executed during a set period of time.

After the time has elapsed, the entries in the holding queue are deleted either manually with the BNU **uucleanup** command or automatically by commands such as **uudemon.cleanu** started by the **cron** daemon.

When sending files to a system that has not been contacted recently, it is a good idea to use the **uustat** command to see when the last access occurred; the remote system may be down or out of service.

Flags

The following flags are mutually exclusive. Use only one at a time with the **uustat** command.

Item	Description
-a	Displays information about all the jobs in the holding queue, regardless of the user who issued the original BNU command.
-kJobID	Cancels the BNU process specified by the <i>JobID</i> variable. The person using this flag must either be the one who made the uucp request now being canceled or be operating with root authority. This flag cancels a process only when that job is still on the local computer. After BNU has moved the job to a remote system for execution, the -k JobID flag cannot be used to cancel the remote job.
-m	Reports the status of the most recent attempt to contact the specified system with a BNU command. If the BNU request was completed, the status report is successful. If the job was not completed, the status report is an error message saying that the login failed.
-n Number	Allows the user to specify the amount of machines from which to collect BNU status information. The amount specified should be greater than or equal to the amount of machines in the Systems file. The default is 200.
-p	Runs a ps -flp (process status: full, long list of specified process IDs) for all PID numbers in the lock files.
-q	Lists the jobs currently queued to run on each system. These jobs are either waiting to execute or in the process of executing. If a status file exists for the system, its date, time, and status information are reported. When the job is finished, BNU removes that job listing from the current queue. In a status report, a number in parentheses next to the number of a C.* (command) file or an X.* (execute) file represents the age in days of the oldest C.* or X.* file for that system. The retry field represents the number of times BNU tried and failed to execute the command because of, for example, a failed login, locked files, or an unavailable device.
-rJobID	Marks the files in the holding queue specified by the <i>JobID</i> variable with the current date and time. Use this flag to ensure that a cleanup operation does not delete files until the job's modification time reaches the end of the specified period. You can use either one or both of the following flags with the uustat command:
-s System	Reports the status of BNU requests for the workstation specified by the <i>System</i> variable. The <i>System</i> name can contain only ASCII characters.
-u User	Reports the status of BNU requests by the user specified by the <i>User</i> variable, for any workstation. The <i>User</i> name can contain only ASCII characters.

Examples

1. To display the status of all BNU jobs in the holding queue, type:

```
uustat -a
```

The system responds with a message similar to the following:

```
heraC3113 11/06-17:47 S hera amy 289 D.venus471afd8
zeusN3130 11/06-09:14 R zeus geo 338 D.venus471bc0a
merlinC3120 11/05-16:02 S merlin amy 828 /home/amy/tt
merlinC3119 11/05-12:32 S merlin msg rmail amy
```

Field	Description
1	Job ID of the operation
2	Date and time the BNU command was issued
3	An S or an R, depending on whether the job is to send or receive a file
4	Name of the system on which the command was entered
5	User ID of the person who issued the command
6	Size of the field or the name of the remote command
7	Name of the file.

When the size of the file is given, as in the first three lines of the example output, the file name is also displayed. The file name can be either the name given by the user, as in the /home/amy/tt entry, or a name that BNU assigns internally to data files associated with remote executions, such as D.venus471afd8.

2. To display the status of all jobs in the current queue, type:

```
uustat -q
```

The system responds with a message similar to the following:

```
merlin 3C 07/15-11:02 NO DEVICES AVAILABLE
hera 2C 07/15-10:55 SUCCESSFUL
zeus 1C (2) 07/15-10:59 CAN'T ACCESS DEVICE
```

This output tells how many **C.*** (command) files are waiting for each system. The number in parentheses (2) in the third line of the example indicates that the **C.*** file has been in the queue for two days. The date and time refer to the current interaction with the system, followed by a report of the status of the interaction.

3. To display all process IDs in the lock file, type:

```
uustat -p
```

The system responds with a message similar to the following:

```
LCK..tty0: 881
LCK.S.0: 879
LCK..hera: 881
F S UID PID PPID C PRI NI ADDR SZ WCHAN STIME TTY
101 S uucp 881 879 26 39 39 370 296 3ffe800 09:57:03 -
TIME CMD
0:00 UUCICO -r1 -shera
101 S uuc 879 1 11 33 39 770 156 8d874 09:57:02 -
0:00 /usr/sbin/uucp/uusched
```

4. To cancel a job in the current queue, first determine its job ID and then issue the command to cancel the job. To determine the job ID, type:

```
uustat -a
```

The system responds with a message similar to the following:

```
heraC3113 11/06-17:47 S hera amy 289 D.venus471afd8
merlinC3119 11/06-17:49 S merlin geo 338 D.venus471bc0a
```

To cancel the job with the ID of heraC3113, type:

```
uustat -k heraC3113
```

5. To report the status of jobs requested by system hera, type:

```
uustat -s hera
```

The system responds with a message similar to the following:

```
heraN1bd7 07/15-12:09 S hera amy 522 /usr/amy/A
heraC1bd8 07/15-12:10 S hera amy 59 D.3b2a12ce4924
heraC3119 07/15-12:11 S hera amy rmail msg
```

Files

Item	Description
/etc/locks	Contains lock files to prevent multiple uses of devices.
/usr/bin/uustat	Specifies the command pathname.
/var/spool/uucp	Contains BNU status information.

uuto Command

Purpose

Copies files from one system to another.

Syntax

```
uuto [ -m ] [ -p ] Source ... User
```

Description

The **uuto** command is a Basic Networking Utilities (BNU) command that copies one or more *Source* files from one system to a specified *User* on another UNIX based system. This program uses the **uucp** command for the actual file transfer, but the **uuto** command enables the recipient to use the **uupick** command options to handle the transferred file on the local system.

The sender issues the **uuto** command to copy one or more files to a specific user ID on another system. The **uucp** command then copies the file to the BNU public directory, **/var/spool/uucppublic**, on the destination system. The **uucp** command also creates an additional subdirectory called **receive** (if it does not already exist) and directories below it in which to hold the files until the recipient retrieves them with the **uupick** command. The full path names to the copied files are some form of the following name:

```
/var/spool/uucppublic/receive/UserName/System/File
```

where the *UserName* and *System* directories are created based on the *User* parameter given with the **uuto** command.

Once the copied file is in the **receive** directory, the **rmail** command notifies the recipient that a file has arrived. The recipient then issues the **uupick** command, and this command searches the public directory for files sent to the recipient and notifies the recipient about each file it locates. The recipient then enters one of the **uupick** options to handle the file.

Source and Destination File Names

The sender must give the name of the file to be sent and user and system to which the file is to be transferred. The *Source* parameter is the path name of the source file. This can be the name of the file if the file is in the directory from which the **uuto** command is issued. If the file is in a different directory, the complete or relative path name of the file must be given.

The *User* parameter is the path name to the specific location where the source file is to be copied. This path name must include the user identification of the person the file is being sent to. The *User* parameter has the form:

```
System!UserName
```

where *System* is the name of the remote system connected to the local system, and *UserName* is the login name of the recipient of the transferred files on the specified system.

When copying a file from one user to another user on the local system, omit the *System* entry; the destination is the ID of the user to whom the file is being sent. System names can contain only ASCII characters.

Flags

Item	Description

-m	Notifies the sender by the bellmail command when the source file has been successfully copied.
-p	Copies the source file to the spool directory on the local system. The source file resides in the spooling directory for a set period of time (defined in the uusched program) before the uucp command calls the uucicodaemon , which actually transfers the copy to the public directory on the specified remote system. The default is to transfer a source file directly to the specified user.

Examples

1. To copy a file to a user on a remote system, enter:

```
uuto /home/bin/file1 zeus!karen
```

In this example, the `/home/bin/file1` file is sent to user `karen` on the remote system `zeus`.

2. To copy a file to a user on a remote system and be notified whether the source file was successfully copied, enter:

```
uuto -m /home/bin/file2 zeus!karen
```

In this example, the `/home/bin/file2` file is sent to user `karen` on the remote system `zeus` and a message is returned to the sender verifying that the copy was successful.

3. To copy a file to another user on your local system, enter:

```
uuto /home/bin/file3 ron
```

In this example, the `/home/bin/file3` file is sent to user `ron` on the local system. No mail message is sent to the recipient in a local transfer.

Files

Item	Description
/usr/bin/uuto	Contains the uuto command.
/var/spool/uucppublic	Is the BNU public directory.

uutry Command

Purpose

Contacts a specified remote system with debugging turned on and allows the user to override the default retry time.

Syntax

uutry [-xDebugLevel] [-r] SystemName

Description

The **uutry** command contacts a remote system, specified by the *SystemName* parameter, using debugging mode. Debugging mode provides a means of monitoring Basic Networking Utilities (BNU) connections to remote computers and file transfers. The **uutry** command calls the **uucico** daemon to contact the remote system.

The debugging output is scrolled on the screen of the local system. Once the system has finished displaying this information, press the Interrupt key to return to the prompt.

The **-r** flag overrides the default retry time if the first attempt to contact the remote system is unsuccessful. The default retry time is 5 minutes.

The *SystemName* parameter, which is required, specifies the name of the remote system you wish to contact.

Requirement: Either you must be in the **/usr/sbin/uucp** directory when you issue the **uutry** command or you must issue the command with the full path name, **/usr/sbin/uucp/uutry**.

Tips:

- The **uutry** command is a shell script stored in the **/usr/lib/uucp** directory.
- If the debugging output scrolls too quickly to be read, use the **Uutry** command to save the output in a temporary file.

Flags

Item	Description
-r	Overrides the default retry time. If for some reason the uucico daemon cannot complete the requested connection, the daemon waits for a set amount of time and tries again. The default retry time is 5 minutes. Note: The time at which the remote system was last polled is recorded in the <i>SystemName</i> file in the /var/spool/uucp/.Status directory.
-xDebugLevel	Overrides the default amount of detail in the debugging information that the uutry command displays on the screen. The valid range for the <i>DebugLevel</i> variable is 0 to 9, with a default of 5. Higher numbers cause the final report to be more detailed. If the -x flag is not used, the uucico daemon is started with the default level, which produces a moderate amount of information.

Examples

1. To change the amount of detail the **uutry** command provides about the progress of the **uucico** operation, use the **-x** flag to specify a different debugging level. For example, entering:

```
/usr/sbin/uucp/uutry -x9 venus
```

instructs the **uutry** command to generate as much information as possible about the way in which the **uucico** daemon is working.

2. The default time at which to retry a contact to a remote system when the first contact was unsuccessful is 5 minutes. To shorten the default retry time for contacting the remote system, enter:

```
/usr/sbin/uucp/uutry -i venus
```

Using the **-r** flag instructs the **uucico** daemon to contact remote system **venus**, overriding the default retry time. The daemon attempts to contact system **venus**, retrying periodically until the connection is successful, and then produces debugging output on the display screen of the local system.

Files

Item	Description
/usr/sbin/uucp/uutry	Contains the uutry command.
/etc/uucp/Devices	Contains information about available devices.
/etc/uucp/Dialcodes	Contains dial-code abbreviations.
/etc/uucp/Dialers	Specifies initial handshaking on a connection.
/etc/uucp/Permissions	Describes access permissions for remote systems.
/etc/uucp/Systems	Describes accessible remote systems.
/var/spool/uucp/.Status/SystemName	Lists the last time the remote system named by the <i>SystemName</i> file was contacted.
/var/spool/uucppublic/*	Contain the BNU public directories.

uux Command

Purpose

Runs a command on another UNIX-based system.

Syntax

```
uux [ -c | -C ] [ -n | -z ] [ - ] [ -aName ] [ -b ] [ -gGrade ] [ -j ] [ -p ] [ -e ] [ -r ] [ -sFile ] [ -xDebugLevel ]  
CommandString
```

Description

The **uux** command is a Basic Networking Utility (BNU) that runs a specified command on a specified UNIX-based system while enabling the user to continue working on the local system. Before running the requested command, the **uux** command gathers any necessary files from the designated systems. The user can direct the output from the command to a specific file on a specific system. For security reasons, many installations permit the **uux** command to run only the **rmail** command.

The **uux** commands on other systems create execute (**X.***) files that run commands on the local system. In addition, the **uux** command on the local system creates both command (**C.***) files and data (**D.***) files for transfer to other systems. Execute files contain the command string to be executed on the designated system. Command files contain the same information as those created by the **uucp** command. Data files either contain the data for a remote command execution or else become **X.*** files on remote systems for remote command executions.

The full path name of an execute file is a form of the following:

```
/var/spool/uucp/System/X.SystemNamexxx
```

After creating the files in the spooling directory, the **uux** command calls the **uucico** daemon to transfer the files from the spooling directory on the local system to the designated remote system. Once the files are transferred, the **uuxqt** daemon on the remote system executes the *CommandString* on the specified system, placing any output from the command in the file designated by the original **uux** command request.

The *CommandString* argument is made up of one or more arguments that look like an operating system command line, except that *CommandString* argument may be prefixed by the name of the remote system in the form *System!*. The default *System* is the local system. Unless the user entering the **uux** command includes the **-n** flag, the command notifies that user if the remote system does not run the command. This response comes by mail from the remote system.

Source and Destination File Names

- When specifying the destination of the output of a command, the **uux** command can be entered in either one of the following formats:
 - **uux** [*Options*] "*CommandString*> *Destination*"
 - **uux** [*Options*] *CommandString* \ {*Destination*}.
- Destination names can be either of the following:
 - A full path name
 - A full path name preceded by *~User*, where *User* is a login name on the specified system. The **uux** command replaces this path name with the user's login directory.
- The shell pattern-matching characters ? (question mark), * (asterisk), and [...] (brackets) can be used in the path name of a source file (such as files compared by the **diff** command); the appropriate system expands them. However, using the * character may occasionally produce unpredictable or unanticipated results. Shell pattern-matching characters should not be used in the destination path name.
- Place either two backslashes (\ . . \) or a pair of quotation marks (" . . ") around pattern-matching characters in a path name so the local shell cannot interpret them before the **uux** command sends the command to a designated system.
- If you are using the special shell characters > (greater than), < (less than), ; (semicolon), or | (vertical bar) in a path name, place either \ . . \ or " . . " around the individual character or around the entire command string.
- Do not use the shell redirection characters << or >> in a path name.
- The **uux** command attempts to move all files specified on the command line to the designated system. Enclose the names of all output files in parentheses so that the **uux** command does not try to transfer them.
- When specifying a *System*, always place it before the *CommandString* argument in the entry. System names can contain only ASCII characters.
- The ! (exclamation point) preceding the name of the local system in a command is optional. If you choose to include the ! to run a command on the local system using files from two different remote systems, use ! instead of *System!* to represent the local system, and add *System!* as the first entry in any path name on the remote systems.
- The exclamation point representing a system in BNU syntax has a different meaning in C shells. When running the **uux** command in a C shell, place a \ (backslash) before the exclamation point in a system name.

Note: The notation ~ (tilde) is the shorthand way of specifying the public spooling directory, **/var/spool/uucppublic**.

Flags

Item	Description
-	Makes the standard input to the uux command the standard input to the <i>CommandString</i> argument.
-a <i>Name</i>	Replaces the user ID of the person issuing the command with the user ID specified with the <i>Name</i> variable.
-b	Returns standard input to the command if the exit status is not zero.
-c	Transfers the source files to the destination on the specified system. The source files are copied into the spooling directory, and the uucico daemon is invoked immediately. This flag is the default.
-C	Transfers the source files to the spool directory. After a set period of time (specified in the uusched program), the uucico daemon attempts to transfer the files to the destination on the specified computer. Occasionally, there are problems in transferring a source file; for example, the remote computer may not be working or the login attempt may fail. In such cases, the file remains in the spool directory until it is either transferred successfully or removed by the uucleanup command.

Item	Description
-e	Enables file expansion.
-g <i>Grade</i>	Specifies when the files are to be transmitted during a particular connection. The <i>Grade</i> variable specifies a single number (0 through 9) or letter (A through Z, a through z); lower ASCII-sequence characters cause the files to be transmitted earlier than do higher sequence characters. The number 0 is the highest (earliest) grade; z is the lowest (latest). The default is N .
-j	Displays the job identification number of the process that is running the command on the specified system. Use this job ID with the BNU uustat command to check the status of the command or with the uustat -k flag to terminate the process.
-n	Prevents user notification by the mail command of the success or failure of a command. The default is to notify the user if the command fails.
-p	Uses the standard input to the uux command as the standard input to the <i>CommandString</i> argument. A - (minus) has the same effect.
-r	Prevents the starting of the spooling program that transfers files between systems. The default is to start the spooling program.
-s <i>File</i>	Reports the status of the transfer in a file specified by the <i>File</i> variable on the designated system. File names can contain only ASCII characters.
-x <i>DebugLevel</i>	Displays debugging information on the screen of the local system. The <i>DebugLevel</i> variable must be a number from 0 to 9. A higher number gives a more detailed report.
-z	Notifies the user if the command completes successfully. This flag is the opposite of the system default, which is to notify the user only in the event of a failure.

Examples

1. To run the **qpri** command on a remote system, enter:

```
uux merlin!qpri /reports/memos/lance
```


In this example, the remote file `/reports/memos/lance` is printed on remote system merlin. Since neither the `-n` nor `-z` flag is specified, the **uux** command notifies the user only if the remote system fails to run the command. The response comes by the **mail** command from the remote system.

2. To run commands on two remote systems, enter the information on separate command lines:

```
uux merlin!qprt /reports/memos/lance
uux zeus!qprt /test/examples/examp1
```

In this example, the remote `/reports/memos/lance` file is printed on remote system merlin, and the remote `/test/examples/examp1` file is printed on remote system zeus. Since neither the `-n` nor `-z` flag is specified, the **uux** command notifies the user only if the remote system fails to run the command. The response comes by the **mail** command from the remote system.

3. To queue a job that compares a file on the local system with a file on a remote system, using the **diff** command on the local system, and get the job ID of the job, enter:

```
uux -j "/usr/bin/diff /usr/amy/f1 hera!/home/amy/f2 > ~/f1.diff"
```

In this example, the `/usr/amy/f1` file on the local system is compared to the `/home/amy/f2` file on the remote system hera and the output is placed in the `f1.diff` file in the local public directory (the full path name of this file is `/var/spool/uucppublic/f1.diff`). The destination name must be entered either preceded by a `>` with the whole command string enclosed in `" "` (quotation marks) or entered enclosed in braces and backslashes, as `\{ DestinationName \}`. The `-j` flag causes the **uux** command to return the BNU job ID of the job.

4. To use the **diff** command on the local system to compare files that are located on two different remote systems, enter:

```
uux "!/usr/bin/diff hera!/usr/amy/f1 venus!/home/amy/f2 > \ !f1.diff"
```

In this example, the `/usr/amy/f1` file from the remote system hera is compared to the `/home/amy/f2` file from the remote system venus and the output is placed in the file `f1.diff`, located in the current working directory on the local system.

The output file must be write-enabled. If you are uncertain about the permission status of a specific target output file, direct the results to the public directory. The exclamation points representing the local system are optional. The destination name must be entered either preceded by a `>` with the whole command string enclosed in `" "` (quotation marks) or entered enclosed in braces and backslashes, as `\{ DestinationName \}`.

5. To execute the **diff** command on two separate files from different systems, enter:

```
uux "hera!/usr/bin/diff /tmp/out1 zeus/tmp/out2 > ~/DF"
```

In this example, the `diff` file is on the remote system hera. The first source file is on the remote system hera, and the second file is on the system zeus. (zeus may be the local system or another remote system.) The output is directed to the file `DF` in the public directory on the local system.

6. To specify an output file on a different remote system, enter:

```
uux hera!uucp venus!/home/amy/f1 \{merlin!/home/geo/test\}
```

In this example, the **uucp** “uucp Command” on page 1110 command is run on the remote system hera, and the `/home/amy/f1` file, stored on system venus, is sent to user geo on system merlin as test. The destination name is entered enclosed in braces and backslashes.

7. To get selected fields from a file on a remote system and place them in a file on the local system, enter:

```
uux "cut -f1 -d: hera\!/etc/passwd > ~/passw.cut"
```

cut command is run on the local system. The first field from each line of the password file on system hera is placed in the `passw.cut` file in the public directory on the local system. The **uux** command is running in a C shell, so a `\` (backslash) must precede the exclamation point in the name of the remote system.

8. To use the **uux** piping option to specify a remote copy of the /tmp/example file to /tmp/examplecopy on system mercury use the following syntax:

```
uux -p mercury!  
cp /tmp/example /tmp/examplecopy
```

The user must enter a Ctrl-D in order to terminate the command input. After Ctrl-D is pressed, the command will be spooled for remote execution on system mercury.

Files

Item	Description
/usr/bin/uux	Contains the uux command.
/var/spool/uucp	Is the spooling directory.
/var/spool/uucppublic	Is the public directory.

uuxqt Daemon

Purpose

Executes Basic Networking Utilities (BNU) remote command requests.

Syntax

```
uuxqt [ -e ] [ -sSystemName ] [ -xDebugLevel ]
```

Description

The Basic Networking Utilities (BNU) **uuxqt** daemon executes commands on designated remote systems.

The **uuxqt** daemon on each networked system periodically searches the spool directory for remote execute (**X.***) files. These files are sent to the directory by the **uucico** daemon in response to a **uux** command.

When it finds **X.*** files, the **uuxqt** daemon checks each file to make sure that:

- All the required data (**D.***) files are available.
- The requesting system has the necessary permissions to access the data files and run the requested commands.

Note: The **uuxqt** daemon uses the /etc/uucp/Permissions file to validate file accessibility and command execution permission.

If the data files are present and the requesting system has the appropriate permissions, the **uuxqt** daemon executes the commands.

Note: The **uuxqt** command is usually executed from the **uudemon.hour** command, a shell procedure, and not entered from the command line. You must have root user privileges to issue the **uuxqt** command from the command line.

Flags

Item	Description
-e	Enables file expansion.

Item	Description
-s <i>SystemName</i>	Designates the remote system to be contacted. Use only when starting the uuxqt command manually. The system name is supplied internally when the uuxqt command is started automatically. Note: System names can contain only ASCII characters.
-x <i>DebugLevel</i>	Displays debugging information on the screen of the local system. The <i>DebugLevel</i> variable is a single digit between 0 and 9, with a default of 5. The higher the <i>DebugLevel</i> variable, the more detailed the debugging information.

Security

Access Control: You must have root authority to start the **uuxqt** daemon from the command line.

Example

To start the **uuxqt** daemon for debugging, enter:

```
/usr/sbin/uucp/uuxqt -svenus -x7
```

This instructs the command to contact remote system venus and provide fairly detailed information about the contact.

Files

Item	Description
/usr/sbin/uucp/uuxqt	Contains the uuxqt daemon.
/etc/locks	Contains lock files that prevent multiple uses of devices and multiple calls to systems.
/etc/uucp/Maxuuxqts	Limits remote command executions.
/etc/uucp/Permissions	Describes access permissions for remote systems.
/var/spool/uucp/*	Contain the execute and data files.

V

The following AIX commands begin with the letter v.

vacation Command

Purpose

Returns a message to the sender that the mail recipient is on vacation.

Syntax

```
vacation [ { -I | User } ] [ [ { -f Number [ Unit ] | User } ]
```

Description

The **vacation** command returns a message to the sender of a mail message to notify the sender that the recipient is on vacation. The intended use is in a **\$HOME/.forward** file that allows messages to come to you while also sending a message back to the sender.

Note: Sendmail version 8.9.3 and subsequent releases have a security enhancement that will ignore the **.forward** file if *either* of the following conditions exist:

- The **.forward** file has group or world writeable permissions
- Any of **.forward** file's parent directories have group or world writable permissions

If you think that the vacation program is not working because the **.forward** file is being ignored, check the permissions. If you must have group or world writeable permissions on any of the parent directories of the **.forward** file, then set the DontBlameSendmail option in the sendmail configuration file with the appropriate values.

The **vacation** command expects a **\$HOME/.vacation.msg** file containing a message to be sent back to each sender. If this file does not exist, the **vacation** command looks for **/usr/share/lib/vacation.def**, a systemwide default vacation message file. It should be an entire message, including any desired headers, such as From or Subject. By default, this message is sent only once a week to each person who sends mail to you. Use the **-f** flag to change the frequency intervals at which the message is sent. The names of the people who send messages are kept in the files **\$HOME/.vacation.pag** and **\$HOME/.vacation.dir**. These files are created when the **vacation** command is initialized for your user ID using the **-I** (uppercase i) flag.

If the **-I** flag is not specified, the **vacation** command reads the first line from the standard input for a From line to determine the sender. If no text is available from standard input, the command returns an error message. All properly formatted incoming mail should have a From line. No message is sent if the From header line indicates that the message is from Postmaster, MAILER-DAEMON, or if the initial From line includes the string-REQUEST@ or if a Precedence: bulk or Precedence: junk line is included in the header.

Flags

Item	Description
-I	Initializes the \$HOME/.vacation.pag and \$HOME/.vacation.dir files. Execute the vacation command using this flag before you modify your \$HOME/.forward file.

Item	Description
-fNumber [Unit]	Specifies the frequency interval at which the vacation message is sent. The <i>Number</i> parameter is an integer value and the <i>Unit</i> parameter specifies a time unit. The <i>Unit</i> parameter can be one of the following: s Seconds m Minutes h Hours d Days w Weeks

Note: The **-f** flag cannot be used with the **-I** flag.

Examples

1. Before you use the **vacation** command to return a message to the sender saying that you are on vacation, you must initialize the **\$HOME/.vacation.pag** and **\$HOME/.vacation.dir** files. To initialize these files, type:

```
vacation -I
```

2. Modify the **.forward** file. For example, Mark types the following statement in the **.forward** file:

```
mark,|"/usr/bin/vacation mark"
```

The sender receives the message that is in the **\$HOME/.vacation.msg** file, or if the file does not exist, the default message found in the **/usr/share/lib/vacation.def** file. If neither of these files exist, no automatic replies are sent to the sender of the mail message and no error message is generated. If either of these files exist, the sender receives one vacation message from mark per week, regardless of how many messages are sent to mark from the sender.

3. If the following entry is contained in your **.forward** file,

```
mark, |"/usr/bin/vacation -f10d mark"
```

The sender receives one vacation message from mark every ten days, regardless of how many messages are sent to mark from the sender.

4. To create a vacation message that is different from the default vacation message, create the file **\$HOME/.vacation.msg** and add your message to this file. The following is an example of a vacation message:

```
From: mark@odin.valhalla (Mark Smith)
Subject: I am on vacation.
Delivered-By-The-Graces-Of: the Vacation program
I am on vacation until October 1. If you have something urgent,
please contact Jim Terry <terry@zeus.valhalla>.
--mark
```

5. To cancel the vacation message, remove the **.forward** file, **.vacation.dir** file, **.vacation.pag** file, and **.vacation.msg** file from your **\$HOME** (login) directory:

```
rm .forward .vacation.dir .vacation.pag .vacation.msg
```

Files

Item	Description
<u>\$HOME/.forward</u>	Contains the names of people who you want your mail to be forwarded to.
/usr/share/lib/vacation.def	Contains the systemwide default vacation message.
<u>\$HOME/.vacation.dir</u>	Contains the names of people who have sent mail to you while the vacation command was being used.
<u>\$HOME/.vacation.msg</u>	Contains your personalized vacation message.
<u>\$HOME/.vacation.pag</u>	Contains the names of people who have sent mail to you while the vacation command was being used.
/usr/bin/vacation	Contains the vacation command.

val Command (SCCS)

Purpose

Validates SCCS files.

Syntax

val [**-s**] [**-rSID**] [**-mName**] [**-yType**] *File* ...

Description

The **val** command reads the specified file to determine if it is a Source Code Control System (SCCS) file meeting the characteristics specified by the accompanying flags. If you specify a - (minus) for the *File* value, the **val** program reads standard input and interprets each line of standard input as **val** flags and the name of an SCCS file. An end-of-file character terminates input.

The **val** command displays messages to standard output for each file processed.

Flags

Each flag or group of flags applies independently to each named file. The flags can appear in any order.

Item	Description
-mName	Compares the <i>Name</i> value with the SCCS 31 identification keyword in the specified file.
-r SID	Specifies the SID of the file to be validated. The SID must be valid and unambiguous.
-s	Suppresses the error message normally written to standard output.
-yType	Specifies a type to compare with the SCCS identification keyword in the specified file.

Exit Status

The **val** command returns 0 if successful for all files; otherwise, it returns an 8-bit code that is a disjunction of the possible errors. It is interpreted as a bit string in which set bits (clockwise) are interpreted as follows:

Item	Description
0x80	Missing file argument.
0x40	Unknown or duplicate option.

Item	Description

0x20	Corrupted SCCS file.
0x10	Cannot open file or file not SCCS.
0x08	SID is invalid or ambiguous.
0x04	SID does not exist.
0x02	, y mismatch.
0x01	31, m mismatch.

Note: The **val** command can process two or more files on a given command line and can process multiple command lines (when reading standard input). In these cases, an aggregate code is returned; a logical OR of the codes generated for each command line and file processes.

Example

To determine if file `s.test.c` is an SCCS text file, enter:

```
val -ytext s.test.c
```

varyoffvg Command

Purpose

Deactivates a volume group.

Syntax

varyoffvg [**-s**] *VolumeGroup*

Description

The **varyoffvg** command deactivates the volume group specified by the *VolumeGroup* parameter along with its associated logical volumes. The logical volumes first must be closed. For example, if the logical volume contains a file system, it must be unmounted.

To activate the volume group, use the **varyonvg** command.

Note: To use this command, you must either have root user authority or be a member of the **system** group.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit varyoffvg
```

Note:

- A volume group that has a paging space volume on it cannot be varied off while the paging space is active. Before you deactivate a volume group with an active paging space volume, ensure that the paging space is not activated automatically at system initialization, and then reboot the system.
- The varyoffvg command discards any background space reclamation process that is running for the volume group. To identify whether a space reclamation is running, you can use the `lvmstat` command with **-r** option.

Flag

Item	Description

-s	Puts the volume group into System Management mode, so that only logical volume commands can be used on the volume group. In this mode, no logical volume can be opened or accessed by users.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To deactivate volume group vg03, enter:

```
varyoffvg vg03
```

2. To deactivate volume group vg02, but allow logical volume commands to continue to take effect, enter:

```
varyoffvg -s vg02
```

Logical volumes within the volume group cannot be opened, but logical volume commands continue to take effect.

File

Item	Description
/usr/sbin/varyoffvg	Contains the varyoffvg command.

varyonvg Command

Purpose

Activates a volume group.

Syntax

```
varyonvg [ -b ] [ -c ] [ -f ] [ -M ltgsize ] [ -n ] [ -p ] [ -r ] [ -s ] [ -t ] [ -u ] [ -k loc|rem ] [ -d ] [ -o ] [ -O ]  
volume group
```

Description

The **varyonvg** command activates the volume group specified by the *volume group* parameter and all associated logical volumes. A volume group that is activated is available for use. When a volume group

is activated, physical partitions are synchronized if they are not current. Physical volumes that are in the PVMISSING state and that have been replaced will be returned to the PVACTION state by the **varyonvg** command.

Note: If a physical volume is part of a dump device, the **varyonvg** command cannot return it to the PVACTION state. To run the command effectively, temporarily change the dump device.

A list of all physical volumes with their status is displayed to standard output whenever there is some discrepancy between the Device Configuration Database and the information stored in the Logical Volume Manager (LVM). The volume group may or may not be varied on. You must carefully examine the list and take proper action depending on each reported status to preserve your system integrity.

While varying on in concurrent mode, if the varyon process detects that there are logical volumes which are not previously known to the system, their definitions are imported. The permissions and ownership of the new device special files are duplicated to those of the volume group special file. If you have changed the permissions and/or ownership of the device special files of the logical volume on the node it was created, you will need to perform the same changes on this node.

Restriction: Classic Concurrent mode is not supported in AIX 5.3.

If the *volume group* cannot be varied on due to a loss of the majority of physical volumes, a list of all physical volumes with their status is displayed. To vary on the *volume group* in this situation, you will need to use the force option.

The **varyonvg** command fails to vary on the volume group if a majority of the physical volumes are not accessible (no Quorum). This condition is true even if the quorum checking is disabled. Disabling the quorum checking will only ensure that the volume group stays varied on even in the case of loss of quorum.

The *volume group* will not vary on if any physical volumes are in the PV_MISSING state and the quorum checking is disabled. This condition is true even if a quorum of disks are available. To vary on in this situation either use the force option or set an environment variable MISSINGPV_VARYON to TRUE (set this value in **/etc/environment** if the volume group needs to be varied with missing disks at the boot time).

In the preceding cases (using the force vary on option and using the MISSINGPV_VARYON variable), you take full responsibility for the *volume group* integrity.

>|When you vary on a volume group that contains an encrypted logical volume, the **varyonvg** command attempts to unlock the encrypted logical volume. If the logical volume is configured with automated key protection methods, such as Platform keystore (PKS) or key server encryption methods, the **varyonvg** command communicates with the **hdcryptmgr** command to unlock the encrypted logical volume. If the unlock operation is successful, the logical volume is ready for I/O operations. Otherwise, any I/O requests to the encrypted logical volume is blocked and an error code of EACCES is returned. If a volume group contains an encrypted logical volume that is not configured with any automated key protection methods, the logical volume is blocked for any I/O operations until the unlock operation is performed successfully.

|<

Requirement: To use this command, you must either have root user authority or be a member of the **system** group.

You could also use the System Management Interface Tool (SMIT) **smit varyonvg** fast path to run this command.

Flags

Item	Description
-b	Breaks disk reservations on disks locked as a result of a normal varyonvg command. Use this flag on a volume group that is already varied on. Notes: • This flag unlocks all disks in a given volume group. • The -b flag opens the disks in the volume group using SC_FORCED_OPEN flag. For SCSI and FC disks this forces open all LUNS on the target address that this disk resides on. Volume groups should therefore not share target addresses when using the varyon -b option. • The -b flag can cause a system to hang if used on a volume group that contains an active paging space.
-c	Varies the volume group on Enhanced Concurrent mode. This is only possible if the volume group is Concurrent Capable or Enhanced Concurrent Capable and the system has the PowerHA SystemMirror product loaded and available. If neither is true, the volume group fails the varyon operation. Requirement: Enhanced Concurrent volume groups use Group Services. Group Services must be configured prior to activating a volume group in this mode.
-d	Allows data divergence. The -d flag only takes effect when you try to bring the volume group online while the cache at the opposite site might contain nonmirrored data updates and that cache is not accessible. If the varyonvg command detects that you might use back-level data and you do not specify the -d flag, the command fails with a meaningful error message. For more information about asynchronous mirroring of geographic LVM, see <i>Geographic Logical Volume Manager for PowerHA SystemMirror® Enterprise Edition</i>.
-f	Allows a volume group to be made active that does not currently have a quorum of available disks. All disks that cannot be brought to an active state will be put in a removed state. At least one disk must be available for use in the volume group. The -f flag (used to override quorum loss) is ignored if the volume group has not lost quorum. If a disk is put into removed state, use the chpv -v a PVname command to bring the disk back to active state.
-k loc rem	Keeps data from the local mirror copy or remote mirror copy. You can specify the following attributes with the -k flag: loc Retains the local mirror copy data. Local means local physical volumes and not primary site rem Retains the remote mirror copy data. Remote means remote physical volumes and not remote site. For more information about asynchronous mirroring of geographic LVM, see <i>Geographic Logical Volume Manager for PowerHA SystemMirror Enterprise Edition</i>.
-M ltgsize	Statically sets the <i>ltgsize</i> of the volume group. Valid values for <i>ltgsize</i> include 128K, 256K, 512K, 1M, 2M, 4M, 8M, 16M, 32M, and 128M. If any disk in the volume group is not configured with a maximum transfer of <i>ltgsize</i> or greater, the varyonvg command will fail.
-n	Disables the synchronization of the stale physical partitions within the <i>volume group</i> parameter.

Item	Description
-o	Allows using data from partitions that are stale in the copy you selected but fresh in the other copy. The varyonvg command fails if you specify the -k flag to preserve either local copy or remote copy in the data divergence case and the varyonvg command cannot preserve the complete copy because some partitions are not fresh in the local or remote copy that you selected. You can override the failure by specifying the -o flag to use data from partitions that are stale in the copy that you selected but fresh in the other copy. The -o flag is only valid with the -k flag. For more information about asynchronous mirroring of geographic LVM, see <i>Geographic Logical Volume Manager for PowerHA SystemMirror Enterprise Edition</i>.
-p	All physical volumes must be available to use the varyonvg command.
-r	Varies on the volume group in read-only mode. This mode prevents: • Write operations to logical volumes • LVM metadata updates • Stale partitions synchronization Restriction: Mounting a JFS file system on a read-only logical volume is not supported. Restriction: All LVM high-level commands that require the LVM metadata update will fail the request in this mode.
-s	Makes the volume group available in System Management mode only. Logical volume commands can operate on the volume group, but no logical volumes can be opened for input or output. Restriction: Logical volume commands also cannot read or write to or from logical volumes in a volume group varied on with the -s flag. Logical volumes that attempt to write to a logical volume in a volume group varied on with the -s flag (such as chvg or mklvcopy) may display error messages indicating that they were unable to write to and/or read from the logical volume.
-t	Checks the timestamps in the Device Configuration Database and the Logical Volume Manager. If there is a discrepancy in the timestamps, the synclvodm command is issued to synchronize the database. Tip: This check is always done if the Volume Group is varied on in concurrent mode.
-u	Varies on a volume group, but leaves the disks that make up the volume group in an unlocked state. Use this flag as part of the initial varyon operation of a dormant volume group.
-O	Force a varyon operation of the volume group even if it is varied on in some other node. Note: In AIX 61 TL8 and later releases, if the volume group created is not allowed to varyon in non-concurrent mode in more than one node then the varyonvg command updates LVM metadata and ODM with the varyon state of the volume group. During varyon time, the varyonvg command reads this data and fails if the volume group is already varied on in another node. The varyoffvg command will reset the varyon state of the volume group during the varyoff time. If the system crashes before varying off the volume group or the volume group is forced off, the varyonvg command fails after you reboot the system. In this scenario, use -O flag to force a varyon operation of the volume group.



Attention: The base design of LVM assumes that only one initiator can access a volume group. The PowerHA SystemMirror product does work with LVM in order to synchronize multi-node accesses of a shared volume group. However, multi-initiator nodes can easily access a volume group with the **-b** and **-u** flags without the use of PowerHA SystemMirror. You must be aware that volume group status information might be compromised or inexplicably altered as a result of disk protect (locking) being bypassed with these two flags. If you use the **-b** and **-u** flags, data and status output cannot be guaranteed to be consistent.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To activate volume group vg03, enter:

```
varyonvg vg03
```

2. To activate volume group vg03 without synchronizing partitions that are not current, enter:

```
varyonvg -n vg03
```

Files

Item	Description
/usr/sbin	Contains the varyonvg command directory.
/tmp	Stores the temporary files while the command is running.

vc Command

Purpose

Substitutes assigned values for identification keywords.

Syntax

```
vc [ -a ] [ -t ] [ -s ] [ -cCharacter ] [ Keyword=Value ]...
```

Description

The **vc** command copies lines from standard input to standard output. The flags and keywords on the command line and control statements in the input modify the resulting output. The **vc** command replaces user-declared keywords with the value assigned on the command line. Keywords can be replaced both in text and in control statements.

Control Statements

A control statement is a single line beginning with a control character (the default control character is a **:** (colon)). Control statements provide conditional processing of the input. The allowable types of control statements are:

:if *Condition*

Text

Item	Description
:end	Writes all the lines between the :if statement and the matching :end to standard output only if the condition is true. You can nest :if and :end statements. However, once a condition is false, all remaining nested :if and :end statements are ignored. See the Condition Syntax section for the syntax of conditions and allowable operators.
:dcl <i>Keyword</i> , [<i>Keyword</i> ...]	Declares specified keywords. All keywords must be declared.
:asg <i>Keyword=Value</i>	Assigns the specified value to the specified keyword. An :asg statement takes precedence over keyword assignment on the vc command line. A later :asg statement overrides all earlier assignments of the associated keyword. The keywords that are declared but not assigned <i>Values</i> , have null values.
:: <i>Text</i>	Removes the two leading control characters, replaces keywords with their respective values, and then copies the line to standard output.
:on or :off	Turns on or off keyword replacement on all lines.
:ctl <i>Character</i>	Changes the control character to the <i>Character</i> value.
:msg <i>Message</i>	Writes a message to standard error output in the form: Message(n): message
	where n is number of the input line on which the message appeared.
:err <i>Message</i>	Writes an error message to standard error. The vc command stops processing and returns an exit value of 1. The error message is in the form:

```
ERROR: message
ERROR: err statement on line n (vc15)
```

Condition Syntax

The items and statements allowed are:

condition	::=OR statement
	::=NOR statement
OR statement	::=AND statement
	::=AND statement OR statement
AND statement	::=expression
	::=expression & AND statement
expression	::=(OR statement)
	::=value operator value
operator value	::= = or != or < or >
	::= ASCII string
	::= numeric string

The available condition operators and their meanings are:

Item	Description
=	Equal
!=	Not equal
&	AND
&	OR
>	Greater than
<	Less than
()	Used for logical groupings
NOT	May only occur immediately after the <i>if</i> , and when present, inverts the value of the entire condition.

The > and < (greater-than and less-than) operate only on unsigned integer values; for example, 012 > 12 is false. All other operators take strings as modifiers; for example, 012 != 12 is true. The precedence of the operators, from highest to lowest precedence, is as follows:

- = != > < (all of equal precedence)
- &
- &|

Parentheses can be used to alter the order of precedence.

Values must be separated from operators or parentheses by at least one blank or tab.

Keyword Replacement

A keyword must begin and end with the same control character used in control statements. A keyword may be up to nine alphanumeric characters, where the first character must be alphabetic. Keyword values can be any ASCII string. A numeric keyword *Value* is an unsigned string of digits. Values cannot contain tabs or spaces.

Flags

Item	Description
-a	Replaces keywords surrounded by control characters with their assigned value in all text lines (not just those beginning with two control characters).
-cCharacter	Uses the <i>Character</i> value as the control character. The <i>Character</i> parameter must specify an ASCII character.
-s	Does not display the warning messages normally displayed to standard error.
-t	Ignores all characters from the beginning of a line up to and including the first tab character for detecting a control statement. If the vc command finds a control character, it ignores all characters up to and including the tab.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion.
>0	An error occurred.

Examples

1. Examples of *Keyword=Value* assignments are:

```
numlines=4
prog=acctg
pass4=yes
```

The **vc** command removes all control characters and keywords from input text lines marked with two control characters as it writes the text to standard output.

2. To prevent a control character from being interpreted, precede it with a backslash, as in the following example:

```
::the :prog: program includes several of the following\:
```

The **:prog:** keyword is replaced by its value, but the \: is passed to standard output as : (colon).

Input lines beginning with a \ (backslash) followed by a control character are not control lines, and are copied to standard output without the backslash. However, the **vc** command writes lines beginning with a backslash and no following control character without any changes (including the initial backslash).

File

Item	Description
/usr/bin/vc	Contains the vc command.

>lvfcstat Command

Purpose

Displays information about Virtual Fibre Channel (VFC) client adapters such as statistics, status, and activities.

Syntax

lvfcstat [-h] | [-l]

lvfcstat -d vfc_client_adapter [-f function_name]

lvfcstat -d vfc_client_adapter -e [-z]

lvfcstat -d vfc_client_adapter [-s [-z]] [-c no_of_times_to_run [-i interval_sec]] [-F]

Description

The **lvfcstat** command reports information such as virtual FC adapter state, status, NPort ID, I/O queues information, read/write statistics, binding information with the host (VIOS), link errors, path timeouts, and so on. The required information is fetched in real time and displayed in the output. You can reset some of the details such as statistics and error counts. You can also retrieve read/write statistics in formatted output for scripting purposes by using the **-F** option. The **lvfcstat** command displays the statistics in a specified time interval for the specific number of times. The **lvfcstat** command works on virtual FC client adapter only.

Flags

-c no_of_times_to_run

Specifies the number of times that the **lvfcstat** command must be run when you specify the stats function. Valid values are in the range 1 - 86400.

-d *vfc_client_adapter*

Specifies the name of the virtual FC client adapter.

-e

Displays error counts for link and path timeout events.

-f *function_name*

Specifies the function to display the output. For a list of functions that are available for a virtual FC client adapter, see [Example 2](#).

-F

Displays formatted statistic output.

-h

Displays usage and help information.

-i *interval_sec*

Specifies the interval between multiple executions of the **vfcstat** command. Valid values are in the range 1 - 86400 seconds. The default value is 1 second.

-l

Lists all virtual FC client adapters.

-s

Displays read/write statistics.

-z

Resets statistics or error count metrics.

Examples

1. To list all the virtual FC client adapters, enter the following command:

```
# vfcstat -l
```

The output might be similar to the following example:

```
fcs0  
fcs1
```

2. To list all the functions available for a specific virtual FC adapter, enter the following command:

```
# vfcstat -d fcs0
```

The output might be similar to the following example:

```
activity  
capabilities  
hostinfo  
link  
nport_id  
status  
tunables  
wwpn
```

3. To display the status of a specific virtual FC adapter, enter the following command:

```
# vfcstat -d fcs0 -f status
```

The output might be similar to the following example:

```

State                0x10 (ONLINE)
Opened               true
cfg_opn_state        0x0
prev_cfg_channels    0
flags                0x1400 ( UP_AFTER_LPM, SCSI_FLOGIN )
total_io_dma_size    0x20000000
total_dma_size_used  0x19900000
no_chs_reason_code   0x0 ( N/A )
migrated_count       0
online_time          61925950 (lbolt)
UP time              498417 sec ( 5 days 18 hours 26 minutes 57 seconds )

Admin Queue
  active_cmds        0
  spl_active_cmds    0
  pending_cmds       0
  spl_pending_cmds   0
  resp_cmds          0

SCSI Queue #1
  active_cmds        0
  spl_active_cmds    0
  pending_cmds       0
  spl_pending_cmds   0
  resp_cmds          0
...

```

4. To display statistics of a specific virtual FC adapter, enter the following command:

```
# vfcstat -d fcs0 -s
```

The output might be similar to the following example:

```

Current time          : May 15 2023, 01:33:13.644
Time since last reset : 499015 sec ( 5 days 18 hours 36 minutes 55 seconds )

SCSI Queue #1
  Read reqs           : 128
  Read bytes          : 10204
  Write reqs          : 29
  Write bytes         : 21034
  No DMA res count    : 2
  No SGE count        : 2
  h_send_dropped      : 1
  h_send_busy         : 1
  No cmd res count    : 3

SCSI Queue #2
  Read reqs           : 99
  Read bytes          : 17695
  Write reqs          : 44
  Write bytes         : 24125
...
...

NVMe Queue #1
  Read reqs           : 990
  Read bytes          : 107695
  Write reqs          : 84
  Write bytes         : 234125
  Domain Ceiling exceed count : 1
  No SGE count        : 2
  h_send_dropped      : 4
  h_send_busy         : 1
  No cmd res count    : 3

```

5. To display statistics of a specific virtual FC client adapter in a formatted output, enter the following command:

```
# vfcstat -d fcs0 -s -F
```

The output might be similar to the following example:

```

Current time=May 15 2023, 01:35:02.284
Time since last reset=499124 sec ( 5 days 18 hours 38 minutes 44 seconds )
Queue_Start
Domain=SCSI
Queue number=1
Read reqs=128
Read bytes=10204
Write reqs=29
Write bytes=21034
No DMA res count=2
No SGE count=2
h_send_dropped =1
h_send_busy=1
No cmd res count=3
Queue_End

Queue_Start
Domain=SCSI
Queue number=2
Read reqs=99
Read bytes=17695
Write reqs=44
Write bytes=24125
...
...
...
Queue_End
...
...

Queue_Start
Domain=NVMeOF
Queue number=1
Read reqs=990
Read bytes=107695
Write reqs=84
Write bytes=234125
Domain Ceiling exceed count=1
No SGE count=2
h_send_dropped=4
vfishost_resource_closed
h_send_busy=1
No cmd res count=3
Queue_End

```

6. To display statistics of a specific virtual FC client adapter in every 10 seconds for 5 times, enter the following command:

```
# vfcstat -d fcs0 -s -c 5 -i 10
```

7. To reset all fields in the stats function for a specific virtual FC adapter, enter the following command:

```
# vfcstat -d fcs0 -s -z
```

8. To display error counts for a specific virtual FC adapter, enter the following command:

```
# vfcstat -d fcs0 -e
```

The output might be similar to the following example:

```

Current time                : Mar 17 2023, 14:28:23.854
Time since last reset       : 33041 sec ( 9 hours 10 minutes 41 seconds )

Link Errors:
    Total errors: 4
    Last 10 minutes: 1
    Last 60 minutes: 2
    Last 24 hours: 4

Path Timeouts:
    Domain: Admin
    Total timeouts: 6
    Last 10 minutes: 1
    Last 60 minutes: 2
    Last 24 hours: 6

Path Timeouts:
    Domain: SCSI
    Total timeouts: 2
    Last 10 minutes: 0
    Last 60 minutes: 1
    Last 24 hours: 2

Path Timeouts:
    Domain: NVMeOF
    Total timeouts: 5
    Last 10 minutes: 1
    Last 60 minutes: 2
    Last 24 hours: 5

```

File

Item	Description
<code>/usr/bin/vfcstat</code>	Contains the vfcstat command.

⏪

vgrind Command

Purpose

Formats listings of programs that are easy to read.

Syntax

```

vgrind [ -f ] [ -n ] [ -t ] [ -x ] [ -PPrintdev ] [ -TName ] [ - ] [ -dFile ] [ -h Header ] [ -LLanguage ] [ -sSize ]
[ File ... ]

```

Description

The **vgrind** command formats (grinds) the program sources specified by the *File* parameters in an easily readable style using the **troff** command. Comments are placed in italics, keywords in boldface, and the name of the current function is listed down the margin of each page as it is encountered.

The **vgrind** command runs in either filter mode or regular mode.

In filter mode, the **vgrind** command acts as a filter in a manner similar to the **tbl** command. Standard input is passed directly to standard output except for lines bracketed by the following **troff**-like macros:

Ite	Description
m	
.vS	Starts processing.
.vE	Ends processing.

The preceding lines are formatted according to the **vgrind** command conventions. The output from this filter can be passed to the **troff** command for output. There is no particular ordering with the **eqn** or **tbl** command.

In regular mode, the **vgrind** command accepts input files, processes them, and passes them in order to the **troff** command, the appropriate postprocessor, and then the printer.

In both modes, the **vgrind** command passes without converting lines, beginning with a decimal point.

The **vgrind** command supports only ASCII keywords defined in either the standard **/usr/share/lib/vgrindefs** language definitions file or any alternately specified file by the **-d** flag.

Flags

Item	Description
-f	Forces filter mode.
-n	Forces no keyword bolding.
-t	Causes formatted text to go to standard output.
-x	Outputs the index file in an easily readable format. The index file itself is produced whenever the vgrind command is run with the index file in the current directory. The index of function definitions can then be run off by running the vgrind command with the -x flag and the <i>File</i> parameter.
-PPrintDev	Sends the output to <i>Printdev</i> Printer using the qpri command. If this flag is not specified, the PRINTER environment variable is used. If the PRINTER environment variable is not set, the system default is used.
-TName	Creates output for a troff device as specified by the <i>Name</i> parameter. The output is sent through the appropriate postprocessor. The default is the ibm3816 postprocessor.
-	Forces input to be taken from standard input (default if the -f flag is specified).
-dFile	Specifies an alternate language definitions file (default is the /usr/share/lib/vgrindefs file).
-h Header	Specifies a particular header to put on every output page (default is the file name).

Note: A blank space is required after the **-h** flag before the *Header* variable.

Item	Description
-lLanguage	Specifies the language to use. Currently known languages are:
c	C (the default). Function names can be preceded on a line only by spaces, tabs, or an asterisk. The parenthetical options must also be on the same line.
csH	CSH.
p	PASCAL. Function names must be displayed on the same line as the function or procedure keywords.
m	MODEL. Function names must be displayed on the same line as the isbeginproc keyword phrase.
sh	SHELL.
r	RATFOR.
mod2	MODULA2.
yacc	YACC.
isp	ISP.
I	ICON.
-s Size	Specifies a point size to use on output (exactly the same as a .ps request).

Files

Item	Description
index	Contains the file the where source for the index is created.
/usr/bin/vgrind	Contains the vgrind command.
/usr/share/lib/tmac/tmac.vgrind	Contains the macro package.
/usr/share/lib/vfontedpr	Contains the preprocessor.
/usr/share/lib/vgrindefs	Contains the language descriptions.

vi or vedit Command

Purpose

Edits files with a full-screen display.

Syntax

```
{ vi | vedit } [ -l ] [ -R ] [ -tTag ] [ -v ] [ -wNumber ] [ -yNumber ] [ -r [ File ] ] [ { + | -c } { Subcommand } ]
[ File ... ]
```

Description

The **vi** command starts a full-screen editor based on the underlying ex editor. Therefore, ex subcommands can be used within the vi editor. The **vedit** command starts a version of the vi editor intended for beginners. In the vedit editor, the **report** option is set to 1, the **showmode** option is set, and the **novice** option is set, making it a line editor.

You start the vi editor by specifying the name of the file or files to be edited. If you supply more than one *File* parameter on the command line, the vi editor edits each file in the specified order. The vi editor on an existing file displays the name of the file, the number of lines, and the number of characters at the end of the screen. In case of multibyte locales the number of characters need to be interpreted as the number of bytes.

Since the vi editor is a full-screen editor, you can edit text on a screen-by-screen basis. The vi editor makes a copy of the file you are editing in an edit buffer, and the contents of the file are not changed until you save the changes. The position of the cursor on the display screen indicates its position within the file, and the subcommands affect the file at the cursor position.

vi Editor Limitations

The following list provides the maximum limits of the vi editor. These counts assume single-byte characters.

- 256 characters per global command list
- 2048 characters in a shell escape command
- 128 characters in a string-valued option
- 30 characters in a tag name
- 128 map macros with 2048 characters total
- 1,048,560 lines silently enforced
- The macro name and the macro text are limited to 100 characters.

Note: The vi editor supports a maximum of 2 GB edit buffer.

vi Editing Modes

The vi editor operates in the following modes:

Item	Description
command mode	When you start the vi editor, it is in command mode. You can enter any subcommand except those designated for use only in the text input mode. The vi editor returns to command mode when subcommands and other modes end. Press the Esc key to cancel a subcommand.
text-input mode	You use the vi editor in this mode to add text. Enter text input mode with any of the following subcommands: the a subcommand, A subcommand, i subcommand, I subcommand, o subcommand, O subcommand, cx subcommands (where the x represents the scope of the subcommand), C subcommand, s subcommand, S subcommand, and R subcommand. After entering one of these subcommands, you can enter text into the editing buffer. To return to command mode, press the Esc key for normal exit or press Interrupt (the Ctrl-C key sequence) to end abnormally.

Item	Description
last-line mode	Subcommands with the prefix : (colon), / (slash), ? (question mark), ! (exclamation point), or !! (two exclamation points) read input on a line displayed at the end of the screen. When you enter the initial character, the vi editor places the cursor at the end of the screen, where you enter the remaining characters of the command. Press the Enter key to run the subcommand, or press Interrupt (the Ctrl-C key sequence) to cancel it. When the !! prefix is used, the cursor moves only after both exclamation points are entered. When you use the : prefix to enter the last-line mode, the vi editor gives special meaning to the following characters when they are used before commands that specify counts: % All lines regardless of cursor position \$ Last line . Current line Note: The history of last line mode subcommands can be navigated using the Up and Down Arrow keys.

Customizing the vi Editor

You can customize the vi editor by:

- Setting vi editor options
- Defining macros
- Mapping keys
- Setting abbreviations

Setting vi Editor Options

The following list describes the vi editor options you can change with the **set** command. The default setting for these options is **off**. If you turn on one of these toggle options, you can turn it off again by entering the word **no** before the option. If you want to discontinue the **autowrite** vi option, enter **noaw**, where **no** turns off the option and **aw** specifies the **autowrite** option.

Note: Do not include parentheses when entering vi options.

vi Option (Abbreviation)	Description
autoindent (ai)	Indents automatically in <u>text input mode</u> to the indentation of the previous line by using the spacing between tab stops specified by the shiftwidth option. The default is noai . To back the cursor up to the previous tab stop, press the Ctrl-D key sequence. This option is not in effect for global commands.
autoprin (ap)	Prints the current line after any command that changes the editing buffer. The default is ap . This option applies only to the last command in a sequence of commands on a single line and is not in effect for global commands.
autowrite (aw)	Writes the editing buffer to the file automatically before the :n subcommand, the :ta subcommand, the Ctrl-A, Ctrl-], and Ctrl -T key sequences, and the ! subcommand if the editing buffer changed since the last write subcommand. The default is noaw .

vi Option (Abbreviation)

backtags (bt)

Description

Allows the Ctrl-T subcommand to return the file editing position to the location where the previous Ctrl-] subcommand was issued. If **nobacktags** is set, then Ctrl-T is the same as Ctrl-]. The default is **backtags**.

beautifying text (bf)

Prevents the user from entering control characters in the editing buffer during text entry (except for tab, new-line, and form-feed indicators). The default is **nobf**. This option applies to command input.

closepunct (cp=)

Handles a list of closing punctuation, especially when wrapping text (**wraptypes** option). Precedes multicharacter punctuation with the number of characters; for example, `cp=3 . . ; } }`. The **vi** command does not split closing punctuation when wrapping.

directory (dir=)

Displays the directory that contains the editing buffer. The default is **dir = /var/tmp**.

edcompatible (ed)

Retains **g** (global) and **c** (confirm) subcommand suffixes during multiple substitutions and causes the **r** (read) suffix to work like the **r** subcommand. The default is **noed**.

exrc (exrc)

If not set, ignores any **.exrc** file in the current directory during initialization, unless the current directory is that named by the **HOME** environment variable. The default is **noexrc**.

hardtabs (ht=)

Tells the vi editor the distance between the hardware tab stops on your display screen. (This option must match the tab setting of the underlying terminal or terminal emulator.) The default is **ht=8**.

history (hist=)

Sets the limit on last line mode history commands. The initial value is `hist=32`. The history size is zero (`hist=0`) for the **tv** command.

ignorecase (ic)

Ignores distinction between uppercase and lowercase while searching for regular expressions. The default is **noic**.

linelimit (ll=)

Sets the maximum number of lines, as per the **-y** command-line option. This option only is effective if used with the **.exrc** file or the **EXINIT** environment variable.

lisp (lisp)

Removes the special meaning of `()`, `{ }`, `[ ]`, and `=` and enables the `=` (formatted print) operator for s-expressions, so you can edit list processing (LISP) programs. The default is **nolisp**.

list (list)

Displays text with tabs (^I) and the marked end of lines (\$). The default is **nolist**.

magic (magic)

Treats the `.` (period), `[` (left bracket), and `*` (asterisk) characters as special characters when searching for a pattern. In off mode, only the `()` (parentheses) and `$` (dollar sign) retain special meanings. However, you can evoke special meaning in other characters by preceding them with a `\` (backslash). The default is **magic**.

vi Option (Abbreviation)

mesg (**mesg**)

Description

Turns on write permission to the terminal if set while in visual mode. This option only is effective if used with the **.exrc** file or the **EXINIT** environment variable. The default is **on**.

modeline (**modeline**)

Runs a vi editor command line if found in the first five or the last five lines of the file. A vi editor command line can be anywhere in a line. For the vi editor to recognize a command line, the line must contain a space or a tab followed by the **ex :** or **vi :** string. The command line is ended by a second **:** (colon). The vi editor tries to interpret any data between the first and second colon as vi editor commands. The default is **nomodeline**.

novice

Indicates whether you are in **novice** mode. You cannot change the value by using the **set** command.

number (**nu**)

Displays lines prefixed with their line numbers. The default is **nonu**.

optimize (**opt**)

Speeds the operation of terminals that lack cursor addressing. The default is **noopt**.

paragraphs (**para=**)

Defines vi macro names that start paragraphs. The default is **para=IPLPPPQPP\ LIpplpipnbp**. Single-letter **nroff** macros, such as the **.P** macro, must include the space as a quoted character if respecifying a paragraph.

partialchar (**pc=**)

Appears in the last display column where a double-wide character would not be displayed completely. The default character is - (minus sign).

prompt

Prompts for a new vi editor command when in command mode by printing a **:** (colon). The default is **on**.

readonly (**ro**)

Sets permanent read-only mode. The default is **noreadonly**.

redraw (**redraw**)

Simulates a smart workstation on a dumb workstation. The default is **nore**.

remap

Allows defining macros in terms of other macros. The default is **on**.

report (**re=**)

Sets the number of times you can repeat a command before a message is displayed. For subcommands that produce many messages, such as global subcommands, the messages are displayed when the command sequence completes. The default is **report=5**.

scroll (**scr=**)

Sets the number of lines to be scrolled when the user scrolls up or down. The default is 1/2 of the window size, rounded down.

sections (**sect=**)

Defines vi macro names that start sections. The default is **sect=NHSHHH\ HUuhsh+c**. Single-letter **nroff** macros, such as the **.P** macro, must include the space as a quoted character if respecifying a paragraph.

shell (**sh=**)

Defines the shell for the **!** subcommand or the **!!** subcommand. The default is the login shell.

vi Option (Abbreviation)

shiftwidth (sw=)

showmatch (sm)

showmode (smd)

slowopen (slow)

tabstop (ts=)

tags (tags =)

term (term=)

terse (terse)

timeout (to)

ttytype

warn (warn)

window (wi=)

wrapmargin (wm=)

wrapscan (ws)

Description

Sets the distance for the software tab stops used by the **autoindent** option, the shift commands (**>** and **<**), and the text input commands (the **Ctrl-D** and **Ctrl-T** key sequences). This vi option only affects the indentation at the beginning of a line. The default is **sw=8**.

Shows the ((matching left parenthesis) or { (left bracket) as you type the) (right parenthesis) or } (right bracket). The default is **nosm**.

Displays a message to indicate when the vi editor is in input mode. The default is **nosmd**.

Postpones updating the display screen during inserts. The default is **noslow**.

Sets the distance between tab stops in a displayed file. The default is **ts=8**.

Defines the search path for the database file of function names created using the **ctags** command. The default is **tags=tags\ /usr/lib/tags**.

Sets the type of workstation you are using. The default is **term=\$TERM**, where **\$TERM** is the value of the **TERM** shell variable.

Allows the vi editor to display the short form of messages. The default is **noterse**.

Sets a time limit of two seconds on an entry of characters. This limit allows the characters in a macro to be entered and processed as separate characters when the **timeout** option is set. To resume use of the macro, set the **notimeout** option. The default is **to**.

Indicates the tty type for the terminal being used. You cannot change this value from the vi editor.

Displays a warning message before the **!** subcommand executes a shell command if it is the first time you issued a shell command after changes were made in the editing buffer but not written to a file. The default is **warn**.

Sets the number of lines displayed in one window of text. The default depends on the baud rate at which you are operating: 600 baud or less, 8 lines; 1200 baud, 16 lines; higher speeds, full screen minus 1 line.

Sets the margin for automatic word wrapping from one line to the next. The default is **wm=0**. A value of 0 turns off word wrapping.

Allows string searches to wrap from the end of the editing buffer to the beginning. The default is **ws**.

vi Option (Abbreviation)	Description
wraptyp (wt=)	Indicates the method used to wrap words at the end of a line. The default value is general . You can specify one of the following four values: general Allows wraps on word breaks as white space between two characters. This setting is the default. word Allows wraps on words. rigid Allows wraps on columns and before closing punctuation. flexible Allows wraps on columns, but one character of punctuation can extend past the margin.
writeany (wa)	Turns off the checks usually made before a write subcommand. The default is nowa .

To see a list of the vi editor settings that have changed from the default settings, enter **set** and press the spacebar. Press the Enter key to return to the command mode.

To see a complete list of the vi editor settings, enter **set all**. Press the Enter key to return to the command mode.

To turn on a vi editor option, enter **set Option**. This command automatically returns you to the command mode.

To turn on multiple vi editor options, enter **set Option Option Option**. This command turns on the three designated vi editor options and returns you to the command mode.

To turn off a vi editor option, enter **set noOption**. This command automatically returns you to the command mode.

To change the value of a vi editor option, enter **set Option=Value**. This command automatically returns you to the command mode.

You can use the **:set** subcommand of the vi editor to set options for this editing session only, or to set options for this editing session and all future editing sessions.

To set or change vi editor options *for this editing session only*, enter the **:set** subcommand from the command line.

To set vi options for *all editing sessions*, put the **:set** subcommand in the **EXINIT** environment variable in the **.profile** file (read by the shell on login) or put the **set** subcommand into a **.exrc** file. The vi editor first looks for the **EXINIT** environment variable and runs its commands. If the **EXINIT** environment variable does not exist, the vi editor then looks for the **\$HOME/.exrc** file and runs its commands. Last, and regardless of any previous results, the vi editor looks for the local **.exrc** file and runs its commands.

Note: This process is true except with the **tvi** command (trusted vi). In this instance, the vi editor looks for and runs only the **/etc/.exrc** file.

For information about changing an option by setting the **EXINIT** environment variable, see the description of environment variables in the **environment** file.

The **.exrc** file can contain subcommands of the form **set Option=Value**; for example:

```
set cp=3 . . ;
```

To include a comment in the **.exrc** file, use a " (double quotation mark) as the first character in the line.

Defining Macros

If you use a subcommand or sequence of subcommands frequently, you can use the vi editor to define a macro that issues that subcommand or sequence.

To define a macro, enter the sequence of subcommands into a buffer named with a letter of the alphabet. The lowercase letters a through z overlay the contents of the buffer, and the uppercase letters A through Z append text to the previous contents of the buffer, allowing you to build a macro piece by piece.

For example, to define a buffer macro named c that searches for the word `corner` and makes the third line after the word `corner` the current line, enter the following command:

```
o /corner/+3
```

Then press the Esc key and enter the following command:

```
"c
```

where c is the name of the buffer macro.

To add text to the previous contents of the defined buffer, enter the o viSubcommand, press the Esc key, and enter "CapitalLetter, where the *CapitalLetter* variable specifies an uppercase letter A through Z. For example, to build a buffer macro named T that searches for the word `corner` and allows you to add more commands, enter the following command:

```
o corner
```

Then press the Esc key and enter the following command:

```
"T
```

where T is the name of the buffer macro. You can repeat this process at any time to add more vi subcommands to the same buffer.

For example, to add commands that move the cursor to the previous line and delete that line, enter the following command:

```
o -dd
```

where - (minus sign) means to move the cursor up one line, and dd means to delete the current line. Press the Esc key and enter the following command:

```
"Tdd
```

To start the macro, enter @Letter, where the *Letter* variable specifies the letter name of the buffer macro you want to use. To use the same macro again, enter @@ (two at symbols). For example, enter @T to start the T buffer macro and run the **search**, **move cursor**, and **delete line** commands. Enter @@T to start the T buffer macro again.

The character set used by your system is defined by the collation table. This table affects the performance of vi macros.

Mapping Keys

You can use the **:map**, **:map!**, and **:ab** subcommands to map a keystroke to a command or a sequence of commands. The **:map** subcommand is used in the command mode. The **:map!** and **:ab** subcommands are used in the text input mode. You can map keys for this editing session and all future editing sessions or only for the current editing session from either mode.

To map keys *for all future editing sessions*, put the subcommand into a **\$HOME/.exrc** file. Each time you start the vi editor, it reads this file. The mapping remains in effect for every editing session.

To map keys *for the current editing session only* from the *command mode*, start the subcommand during the vi editor session. To map keys for the current editing session only from the *text input mode*, enter the subcommand on the command line during the vi editor session. The mapping remains in effect only for the current editing session.



Attention: If you use an IBM 3161 ASCII display station, IBM 3163 ASCII display station, or IBM 3101 ASCII display station, the default key-mapping of the vi editor can cause you to lose data. To see the default mapping, issue a **:map** subcommand. Specific problems arise with the Esc-J or Shift-J key sequence. These key sequences delete all information from the current position of the cursor to the end of the file. To avoid problems, change this key sequence using a **.exrc** file.

The **:map**, **:map!**, and **:ab** subcommands are defined and used as follows:

Item	Description

:map	Defines macros in the command mode. The :map subcommand allows you to run a specified command or sequence of commands by pressing a single key while in the vi editor.

To map keys in the command mode, start the vi editor with an empty editing buffer and do not name a vi file using the **vi** command or type anything into the buffer after the vi editor starts. You can use the **:map** subcommand to do the following:

- To map a character to a sequence of editing commands, enter:

```
:map Letter viSubcommand
```

- To unmap a character previously mapped in command mode, enter:

```
:unmap Letter
```

- To display a list of current mappings for the command mode, enter

```
:map
```

The following keys are not used by the vi editor, but are available for use with the **:map** subcommand in the command mode:

- Letters g, K, q, V, and v
- Control key sequences Ctrl-A, Ctrl-K, Ctrl-O, Ctrl-W, and Ctrl-X
- Symbols _ (underscore), * (asterisk), \ (backslash), and = (equal sign)

Although you can map a key that is already used by the vi editor, the key's usual function is not available as long as the map is in effect. Some terminals allow you to map command sequences to function keys. If you are in LISP mode, the = (equal sign) cannot be used because it is used by the vi editor.

To map the letter v to the sequence of commands that would locate the next occurrence of the word map and change it to the word MAP, enter the following command:

```
:map v /map<Ctrl-V><Enter>cwMAP<Ctrl-V><Esc><Ctrl-V><Enter>
```

The previous example instructs the vi editor to locate the next occurrence of map (/map<Ctrl-V><Enter>), change map to MAP (cwMAP), end the change-word subcommand (<Ctrl-V><Esc>), and enter the command (<Ctrl-V><Enter>).

Requirement: To prevent the vi editor from interpreting the Enter key, it must be preceded by the Ctrl-V key sequence when being mapped. This condition is also true of the Esc, Backspace, and Delete keys.

To map the control characters Ctrl-A, Ctrl-K, and Ctrl-O, simultaneously press the Ctrl key and the letter. For example, to map the Ctrl-A key sequence to the sequence of commands that saves a file and edits the next one in a series, enter the following command:

```
:map <Ctrl-A> :w<Ctrl-V><Enter>;n<Ctrl-V><Enter>
```

To map the control characters Ctrl-T, Ctrl-W, and Ctrl-X, you must first escape them with the Ctrl-V key sequence.

Item	Description

To map the | (pipe symbol), you must first escape it with the two Ctrl-V key sequences, as illustrated by the following example that maps the character g to the sequence of commands that escapes to the shell, concatenates the file **/etc/motd**, and pipes the output to the **wc** command:

```
:map g :!cat /etc/motd <Ctrl-V><Ctrl-V>| wc<Ctrl-V><Enter>
```

If your terminal permits you to map function keys, you must reference them with the **#number** key sequence to designate the number of the function key that you want to map. In the following example, the F1 function key is mapped to the sequence of commands that deletes a word and moves the cursor three words down:

```
:map #1 dwwww
```

In order for function key mapping to work, the output of the function key for your terminal type must match the output defined in the **terminfo** file. These definitions are denoted by the **kfnumber** entries, where kf1 represents the F1 function key, kf2 represents the F2 function key, and so on. If the output that you get when you press the function key does not match this entry, you must use the terminal's setup mode to correct the settings to match these terminal database entries before any mapping can occur.

You can also map certain keyboard special keys, such as the Home, End, Page Up, and Page Down keys. For most terminals, these keys are already mapped in the vi editor. You can verify this mapping by using the **:map** subcommand. If these keys are not already mapped, you can use the **:map** subcommand as follows:

```
:map <Ctrl-V><End> G
:map <Ctrl-V><Home> 1G
:map <Ctrl-V><PageUp> <Ctrl-F>
:map <Ctrl-V><PageDown> <Ctrl-B>
```

To get a listing of all current maps in the command mode, enter the **:map** subcommand. The preceding examples are then displayed as follows:

```
v          v          /map<Ctrl-M>cwMAP<Ctrl-[><Ctrl-M>
<Ctrl-A>  <Ctrl-A>      :w<Ctrl-M>:n<Ctrl-M>
g          g          :!cat /etc/motd | wc <Ctrl-M>
```

Tip: The Ctrl-V and Enter key sequence is displayed as the Ctrl-M key sequence, and the Ctrl-V and Esc key sequence is displayed as the Ctrl-[key sequence.

Item	Description

:map!	Maps character strings to single keys while in text input mode. To map keys in the text input mode, start the vi editor with an empty editing buffer and do not name a vi file using the vi command or type anything into the buffer after the vi editor starts. You can use the :map! subcommand to do the following:

- To map a letter to one or more vi strings in text input mode, enter:

```
:map! Letter String
```

- To unmap a letter previously mapped in text input mode, enter:

```
:unmap! Letter
```

- To display a list of existing strings that are mapped to specific keys in text input mode, enter:

```
:map!
```

Typing the mapped key while in text input mode produces the specified string. The Ctrl-V and Esc key sequence puts you into command mode, backs up to the beginning of the current word (**bbw**), and starts the **cw** (change-word) subcommand. For example:

```
:map! % <Ctrl-V><Esc>bbwcw
```

When typing text, if you realize that you have mistyped a word, you can change it by pressing the % (percent) key and retyping the word. You are automatically returned to insert mode.

Important: Be careful when choosing keys to be used for the **:map!** subcommand. Once keys have been mapped, they can no longer be input as text without first issuing the **:unmap!** subcommand.

Item	Description

:ab	Maps a key or sequence of keys to a string of characters for use in the text input mode. The :ab subcommand is useful when inputting text that possesses several repetitive phrases, names, or titles.

The following example replaces the word `city` with the phrase `Austin, Texas 78759` whenever it is typed in text input mode and followed by a white space, period, or comma:

```
:ab city Austin, Texas 78759
```

For example, if while inputting text, you type the following:

```
My current residence is city.
```

Pressing the Tab key expands the word `city` to read:

```
My current residence is Austin, Texas 78759.
```

The abbreviation is not expanded within a word. For example, if you type `My current residence iscity`, the word `iscity` is not expanded.

If the **:map!** subcommand is used to map abbreviations for insert mode, then all occurrences of the abbreviations are expanded regardless of where it occurs. If you used the **:map!** subcommand for the preceding example (`:map! city Austin, Texas 78759`), then whenever you type the word `city`, regardless of what precedes or follows, the word will be expanded to `Austin, Texas 78759`. Therefore, the word `iscity` becomes `isAustin, Texas 78759`.

Important: Be careful when choosing the keys that are used for the **:ab** subcommand. Once keys are defined, they can no longer be input as text without first issuing the **:unab** subcommand.

Setting Abbreviations

The **set** command has behavior similar to the **map!** command except that the **set** command substitutes the string for the abbreviation only when the abbreviation is a separate word. You can use the **set** command of the vi editor to:

- List existing abbreviations
- Remove an abbreviation
- Set (define) an abbreviation

Tip: Start the vi editor with an empty editing buffer. Do not name a vi file using the **vi** command or type anything into the buffer after the vi editor starts. Press the Esc key to be sure you are in the command mode.

Item	Description

To list abbreviations	

	Enter the :ab command to list existing abbreviations. Press the Enter key to return to command mode.

To remove abbreviations	

	Enter the :anabAbbreviation command to remove an abbreviation, where the <i>Abbreviation</i> variable specifies the character string you do not want abbreviated any more.

Item	Description
To set (define) an abbreviation	Enter the :ab <i>Abbreviation String</i> command to set an abbreviation, where the <i>Abbreviation</i> variable specifies the character string being defined as an abbreviation and the <i>String</i> variable specifies the character string being abbreviated. The abbreviation can be substituted for the string only when the abbreviation is a separate word. For example, if you enter the :ab kn upper command and then type acknowledge while in the text input mode, the set abbreviation string is not started because the kn string in the word acknowledge is not a separate word. However, if you type the :ab kn upper command and then type make the kn line all kn case while in the text input mode, the result is make the upper line all uppercase.

Flags

Item	Description
-c <i>Subcommand</i>	Carries out the ex editor subcommand before viewing with vi begins. The cursor moves to the line affected by the last subcommand to be carried out. When a null operand is entered, as in -c' , the vi editor places the cursor on the first line of the file. The -c flag is incompatible with the + flag. Do not specify both flags at the same time.
-l	Enters the vi editor in LISP mode. In this mode, the vi editor creates indents appropriate for LISP code, and the (,) , { , } , [[, and]] subcommands are modified to act appropriately for LISP.
-r [<i>File</i>]	Recovers a file after a vi editor or system malfunction. If you do not specify the <i>File</i> variable, the vi editor displays a list of all saved files.
-R	Sets the readonly option to protect the file against overwriting.
-t <i>Tag</i>	Edits the file containing the <i>Tag</i> variable and positions the vi editor at its definition. To use this flag, you must first create a database of function names and their locations using the ctags command.
-v	Enters the vi editor in the verbose mode.
-w <i>Number</i>	Sets the default window size to the value specified by the <i>Number</i> variable. This flag is useful when you use the vi editor over a low-speed line.
-y <i>Number</i>	Overrides the maximum line setting of 1,048,560 with any value greater than 1024. You should request twice the number of lines that you require because the vi editor uses the extra lines for buffer manipulation.
+ [<i>Subcommand</i>]	Carries out the ex editor subcommand before editing begins. If you do not specify the <i>Subcommand</i> variable, the cursor is placed on the first line of the file. This + flag is incompatible with the -c flag. Do not specify both flags at the same time.

vi General Subcommand Syntax

Use the following general syntax to enter subcommands:

[*Named_Buffer*] [*Operator*] [*Number*] *Object*

Tip: Square brackets indicate optional items.

Item	Description
[<i>Named_Buffer</i>]	Specifies a temporary text storage area.
[<i>Operator</i>]	Specifies the subcommand or action; instructs the vi editor.
[<i>Number</i>]	Specifies either the extent of the action or a line address as a whole number.
<i>Object</i>	Specifies what to act on, such as a text object (a character, word, sentence, paragraph, section, character string) or a text position (a line, position in the current line, screen position).

Counts before Subcommands

You can put a number in front of many subcommands. The vi editor interprets this number in one of the following ways:

- Go to the line specified by the *Number* parameter:

```
5G
10Z
```

- Go to the column specified by the *Number* parameter:

```
25|
```

- Scroll the number of lines up or down specified by the *Number* parameter:

```
10Ctrl-U
10Ctrl-D
```

vi Editor Subcommands

Use the subcommands to perform these kinds of actions:

- Moving the cursor
- Editing text
- Manipulating files
- Other actions

Moving the Cursor

Use subcommands to move the cursor within a file in these ways:

- Moving within a line
- Moving within a line by character position
- Moving to words
- Moving by line position
- Moving to sentences, paragraphs, or sections
- Moving by redrawing the screen
- Paging and scrolling
- Searching for patterns
- Marking a specific location in a file and returning

Moving within a Line

Enter the following subcommands in command mode. You can cancel an incomplete command by pressing the Esc key. If you need information about the format of vi subcommands, see [vi General Subcommand Syntax](#).

Item	Description
Left Arrow or h or Ctrl-H	Moves the cursor one character to the left.
Down Arrow or j or Ctrl-J or Ctrl-N	Moves the cursor down one line (it remains in the same column).
Up Arrow or k or Ctrl-P	Moves the cursor up one line (it remains in the same column).
Right Arrow or l	Moves the cursor one character to the right.

Moving within a Line by Character Position

Enter the following subcommands in command mode. You can cancel an incomplete command by pressing the Esc key. If you need information about the format of vi subcommands, see [vi General Subcommand Syntax](#).

Item	Description
^	Moves the cursor to the first nonblank character.
0	Moves the cursor to the beginning of the line.
\$	Moves the cursor to the end of the line.
f <i>x</i>	Moves the cursor to the next <i>x</i> character.
F <i>x</i>	Moves the cursor to the last <i>x</i> character.
t <i>x</i>	Moves the cursor to one column before the next <i>x</i> character.
T <i>x</i>	Moves the cursor to one column after the last <i>x</i> character.
;	Repeats the last f , F , t , or T subcommand.
,	Repeats the last f , F , t , or T subcommand in the opposite direction.
<i>Number</i> 	Moves the cursor to the specified column.

Moving to Words

Enter the following subcommands in command mode. For more information about the format of vi subcommands, see [vi General Subcommand Syntax](#).

Item	Description
m	
w	Moves the cursor to the next small word.
b	Moves the cursor to the previous small word.
e	Moves the cursor to the next end of a small word.
W	Moves the cursor to the next big word.
B	Moves the cursor to the previous big word.
E	Moves the cursor to the next end of a big word.

Moving by Line Position

Enter the following subcommands in command mode. If you need information about the format of vi subcommands, see vi General Subcommand Syntax.

Item	Description
H	Moves the cursor to the first line on the screen.
L	Moves the cursor to the last line on the screen.
M	Moves the cursor to the middle line on the screen.
+	Moves the cursor to the next line at its first nonblank character.
-	Moves the cursor to the previous line at its first nonblank character.
Enter	Moves the cursor to the next line at its first nonblank character.

Moving to Sentences, Paragraphs, or Sections

Enter the following subcommands in command mode. You can cancel an incomplete subcommand by pressing the Esc key. If you need information about the format of vi subcommands, see vi General Subcommand Syntax.

Item	Description
(	Places the cursor at the beginning of the previous sentence, or the previous s-expression if you are in LISP mode.
)	Places the cursor at the beginning of the next sentence, or the next s-expression if you are in LISP mode.
{	Places the cursor at the beginning of the previous paragraph, or at the next list if you are in LISP mode.
}	Places the cursor at the beginning of the next paragraph, at the next section if you are in C mode, or at the next list if you are in LISP mode.
]]	Places the cursor at the next section, or function if you are in LISP mode.
[[	Places the cursor at the previous section, or function if you are in LISP mode.

Moving by Redrawing the Screen

Enter the following subcommands in command mode. You can cancel an incomplete subcommand by pressing the Esc key. If you need information about the format of vi subcommands, see vi General Subcommand Syntax.

Item	Description
z	Redraws the screen with the current line at the start of the screen.
z-	Redraws the screen with the current line at the end of the screen.
z.	Redraws the screen with the current line at the center of the screen.
/Pattern/z-	Redraws the screen with the line containing the character string, specified by the <i>Pattern</i> parameter, at the end.

Paging and Scrolling

Enter the following subcommands in command mode. You can cancel an incomplete subcommand by pressing the Esc key. If you need information about the format of vi subcommands, see vi General Subcommand Syntax.

Item	Description
Ctrl-U	Scrolls up one-half screen.
Ctrl-D	Scrolls down one-half screen.
Ctrl-F	Scrolls forward one screen.
Ctrl-B	Scrolls backward one screen.
Ctrl-E	Scrolls the window down one line.
Ctrl-Y	Scrolls the window up one line.
z+	Pages up.
z^	Pages down.

Searching for Patterns

Enter the following subcommands in [command mode](#). You can cancel an incomplete subcommand by pressing the Esc key. If you need information about the format of vi subcommands, see [vi General Subcommand Syntax](#).

Item	Description
[Number]G	Places the cursor at the line number specified by the <i>Number</i> parameter or at the last line if the <i>Number</i> parameter is not specified.
/Pattern	Places the cursor at the next line containing the character string specified by the <i>Pattern</i> parameter.
?Pattern	Places the cursor at the next previous line containing the character string specified by the <i>Pattern</i> parameter.
n	Repeats the last search for the text specified by the <i>Pattern</i> parameter in the same direction.
N	Repeats the last search for the text specified by the <i>Pattern</i> parameter in the opposite direction.
/Pattern/+Number	Places the cursor the specified number of lines after the line matching the character string specified by the <i>Pattern</i> parameter.
?Pattern?-Number	Places the cursor the specified number of lines before the line matching the character string specified by the <i>Pattern</i> parameter.
%	Finds the parenthesis or brace that matches the one at current cursor position.

Editing Text

The subcommands for editing enable you to perform the following tasks:

- Marking a specific location in a file and returning
- Adding text to a file
- Changing text while in input mode
- Changing text from command mode
- Copying and moving text
- Restoring and repeating changes

Marking a Specific Location in a File and Returning

Enter the following subcommands in [command mode](#). You can cancel an incomplete subcommand by pressing the Esc key. If you need information about the format of vi subcommands, see [vi General Subcommand Syntax](#).

Item	Description

"	Moves the cursor to the previous location of the current line.
"	Moves the cursor to the beginning of the line containing the previous location of the current line.
mx	Marks the current position with the letter specified by the x parameter.
`x	Moves the cursor to the mark specified by the x parameter.
'x	Moves the cursor to the beginning of the line containing the mark specified by the x parameter.

Adding Text to a File (Text Input Mode)

Enter the following subcommands in [command mode](#) to change the vi editor into text input mode. If you need information about the format of vi subcommands, see [vi General Subcommand Syntax](#).

Item	Description

a <i>Text</i>	Inserts text specified by the <i>Text</i> parameter after the cursor. End text input mode by pressing the Esc key.
A <i>Text</i>	Adds text specified by the <i>Text</i> parameter to the end of the line. End text input mode by pressing the Esc key.
i <i>Text</i>	Inserts text specified by the <i>Text</i> parameter before the cursor. End text input mode by pressing the Esc key.
I <i>Text</i>	Inserts text specified by the <i>Text</i> parameter before the first nonblank character in the line. End text input mode by pressing the Esc key.
o	Adds an empty line after the current line. End text input mode by pressing the Esc key.
O	Adds an empty line before the current line. End text input mode by pressing the Esc key.

Changing Text While in Input Mode

Use the following subcommands only while in [text input mode](#). These commands have different meanings in [command mode](#). If you need information about the format of vi subcommands, see [vi General Subcommand Syntax](#).

Item	Description

Ctrl-D	Goes back to previous autoindent stop.
^ Ctrl-D	Ends autoindent for this line only.
0Ctrl-D	Moves cursor back to left margin.
Esc	Ends insertion and returns to command state.
Ctrl-H	Erases the last character.
Ctrl-Q	Enters any character if xon is disabled.
Ctrl-V	Enters any character.
Ctrl-W	Erases the last small word.
\	Quotes the erase and kill characters.
Ctrl-?	Interrupts and ends insert or the Ctrl-D key sequence.

Changing Text from Command Mode

Use the following subcommands in [command mode](#). An incomplete subcommand can be canceled by pressing the Esc key. If you need information about the format of vi subcommands, see [vi General Subcommand Syntax](#).

Item	Description
c	Changes the rest of the line (same as c\$).
cc	Changes a line.
cw	Changes a word.
cwText	Changes a word to the text specified by the <i>Text</i> parameter.
D	Deletes the rest of the line (same as d\$).
dd	Deletes a line.
dw	Deletes a word.
J	Joins lines.
rx	Replaces the current character with the character specified by x.
RText	Overwrites characters with the text specified by the <i>Text</i> parameter.
s	Substitutes characters (same as cl).
S	Substitutes lines (same as cc).
u	Undoes the previous change.
x	Deletes a character at the cursor.
X	Deletes a character before the cursor (same as dh).
<<	Shifts one line to the left.
<L	Shifts all lines from the cursor to the end of the screen to the left.
>>	Shifts one line to the right.
>L	Shifts all lines from the cursor to the end of the screen to the right.
~	Changes letter at the cursor to the opposite case.
!	Indents for LISP.

Copying and Moving Text

Use the following subcommands in [command mode](#). An incomplete subcommand can be canceled by pressing the Esc key. If you need information about the format of vi subcommands, see [vi General Subcommand Syntax](#).

Ite	Description
m	
p	Puts back text from the undo buffer after the cursor.
P	Puts back text from the undo buffer before the cursor.
"x	Puts back text from the x buffer.
p	
"x	Deletes text into the x buffer.
d	

Item	Description

y	Places the object that follows (for example, w for word) into the undo buffer.
"x	Places the object that follows into the x buffer, where x is any letter.
y	
Y	Places the line in the undo buffer.

Restoring and Repeating Changes

Use the following subcommands in command mode. An incomplete subcommand can be canceled by pressing the Esc key. If you need information about the format of vi subcommands, see vi General Subcommand Syntax.

Item	Description

u	Undoes the last change. Tip: After an undo, the cursor moves to the first non-blank character on the updated current line.
U	Restores the current line if the cursor has not left the line since the last change.
.	Repeats the last change or increments the "np command.

Note:

1. This subcommand will repeat the last change, including an undo. Therefore, after an undo, repeat performs an undo rather than repeat the last change.
2. This subcommand is not meant for use with a macro. Enter @@ (two at signs) to repeat a macro.

"n	Retrieves the <i>n</i> th last delete of a complete line or block of lines.
p	

Manipulating Files

The subcommands for manipulating files allow you to do the tasks outlined in the following sections:

- Saving changes to a file
- Editing a second file
- Editing a list of files
- Finding file information

Saving Changes to a File

Use the following subcommands in command mode. If you need information about the format of vi subcommands, see vi General Subcommand Syntax.

Item	Description
:w	Writes the edit buffer contents to the original file. If you are using this subcommand within the <u>ex</u> editor, you do not need to type the : (colon).
:w File	Writes the edit buffer contents to the file specified by the <i>File</i> parameter. If you are using this subcommand within the <u>ex</u> editor, you do not need to type the : (colon).
:w! File	Overwrites the file specified by the <i>File</i> parameter with the edit buffer contents. If you are using this subcommand within the <u>ex</u> editor, you do not need to type the : (colon).

Editing a Second File

Enter the following subcommands in command mode. If you need information about the format of vi subcommands, see vi General Subcommand Syntax.

Item	Description
:e <i>File</i>	Edits the specified file. If you are using this subcommand from the <u>ex</u> editor, you do not need to type the : (colon).
:e!	Re-edits the current file and discards all changes.
:e + <i>File</i>	Edits the specified file starting at the end.
:e + <i>Number File</i>	Edits the specified file starting at the specified line number.
:e #	Edits the alternate file. The alternate file is usually the previous file name before accessing another file with a :e command. However, if changes are pending on the current file when a new file is called, the new file becomes the alternate file. This subcommand is the same as the Ctrl-A subcommand.
:r <i>File</i>	Reads the file into the editing buffer by adding new lines after the current line. If you are using this subcommand from the <u>ex</u> editor, you do not need to type the : (colon).
:r ! <i>Command</i>	Runs the specified command and places its output into the file by adding new lines after the current cursor position.
:ta <i>Tag</i>	Edits a file containing the <i>Tag</i> tag starting at the location of the tag. To use this subcommand, you must first create a database of function names and their locations using the ctags command. If you are using this subcommand from the <u>ex</u> editor, you do not need to type the : (colon).
Ctrl-]	Edits a file containing the tag associated with the current word starting at the location of the tag. To use this subcommand, you must first create a database of function names and their locations using the ctags command. Ctrl-T edits a file at the editing position where the previous Ctrl-] subcommand was issued. If multiple Ctrl-] subcommands have been issued, then multiple Ctrl-T subcommands can be used to return to previous editing positions where Ctrl-] subcommands were issued.
Ctrl-A	Edits the alternate file. The alternate file is usually the previous current file name. However, if changes are pending on the current file when a new file is called, the new file becomes the alternate file. This subcommand is the same as the :e # subcommand.

Editing a List of Files

Enter the following subcommands in command mode. If you need information about the format of vi subcommands, see vi General Subcommand Syntax.

Item	Description
:n	Edits the next file in the list entered on the command line. If you are using this subcommand from the <u>ex</u> editor, a : (colon) is not needed.
:n <i>Files</i>	Specifies a new list of files to edit. If you are using this subcommand from the <u>ex</u> editor, a : (colon) is not needed.

Finding File Information

Enter the following subcommand in command mode. If you need information about the format of vi subcommands, see vi General Subcommand Syntax.

Item	Description
Ctrl-G	Shows the current file name, current line number, number of lines in the file, and percentage of the way through the file where the cursor is located.

Other Actions

The vi editor provides the subcommands described in the following sections:

- Adjusting the screen
- Entering shell commands
- Interrupting and ending the vi editor

Adjusting the Screen

Enter the following subcommands in command mode. An incomplete subcommand can be canceled by pressing the Esc key. If you need information about the format of vi subcommands, see [vi General Subcommand Syntax](#).

Item	Description
Ctrl-L	Clears and redraws the screen.
Ctrl-R	Redraws the screen and eliminates blank lines marked with @ (at sign).
zNumber	Makes the window the specified number of lines long.

Entering Shell Commands

The following subcommands allow you to run a command within the vi editor. Enter these subcommands in command mode. If you need information about the format of vi subcommands, see [vi General Subcommand Syntax](#).

Item	Description
:sh	Enters the shell to allow you to run more than one command. You can return to the vi editor by pressing the Ctrl-D key sequence. If you are using this subcommand within the <u>ex</u> editor, a : (colon) is not needed.
:!Command	Runs the specified command and then returns to the vi editor. If you are using this subcommand within the <u>ex</u> editor, a : (colon) is not needed. Tip: The # (alternate file), % (current file), and ! (previous command) special characters are expanded when following a :! subcommand. To prevent any of these characters from being expanded, use the \ (backslash).
:!!	Repeats the last :!Command subcommand.
Number!!Command	Runs the specified command and replaces the lines specified by <i>Number</i> with the output of the command. If a number is not specified, the default value is 1. If the command expects standard input, the specified lines are used as input.
!Object Command	Runs the specified command and replaces the object specified by the <i>Object</i> parameter with the output of the command. If the command expects standard input, the specified object is used as input.

Interrupting and Ending the vi Editor

Enter the following subcommands in command mode. If you need information about the format of vi subcommands, see vi General Subcommand Syntax.

Item	Description
Q	Enters the <u>ex</u> editor in command mode.
ZZ	Exits the vi editor, saving changes.
:q	Quits the vi editor. If you have changed the contents of the editing buffer, the vi editor displays a warning message and does not quit. If you are using this subcommand from the <u>ex</u> editor, a : (colon) is not needed.
:q!	Quits the vi editor, discarding the editing buffer. If you are using this subcommand from the <u>ex</u> editor, a : (colon) is not needed.
Esc	Ends text input or ends an incomplete subcommand.
Ctrl-?	Interrupts a subcommand.

Exit Status

The following exit values are returned:

Item	Description
0	Indicates successful completion.
>0	Indicates an error occurred.

Input Files

Input files must be text files or files that are similar to text files except for an incomplete last line that contains no null characters.

The **.exrc** files must be text files consisting of **ex** commands.

The `$HOME/.vi_history` file is an auto-generated text file that records the last line mode command history.

By default, the vi editor reads lines from the files to be edited without interpreting any of those lines as any form of vi editor command.

view Command

Purpose

Starts the vi editor in read-only mode.

Syntax

view [-cSubcommand] [-l] [-t Tag] [-wNumber] [-y] [-r File] [+ Subcommand] [File ...]

Description

The **view** command starts the vi full-screen editor in read-only mode. The read-only mode is only advisory to prevent accidental changes to the file. To override read-only mode, use the **!** (exclamation point) when executing a command. The *File* parameter specifies the name of the file you want to browse. Use vi subcommands for moving within the file. Use the **:q** subcommand to exit the **view** command. If you modify the file you can save your modifications by pressing the Esc key and **wq!**.

Flags

Item	Description
-c <i>Subcommand</i>	Carries out the ex editor subcommand before viewing with vi begins. When a null operand is entered, as in -c ' ', the editor places the cursor on the last line of the file.
-l	Enters a version of the vi editor with specialized features designed for writing programs in the LISP language. In this mode, the vi editor indents appropriately for LISP programming, and the (,), {, }, [[, and]] subcommands are modified to act appropriately for LISP.
-r [<i>File</i>]	Recovers a file after an editor or system crash. If you do not specify a <i>File</i> parameter, the editor displays a list of all saved files.
-t <i>Tag</i>	Edits the file containing the tag specified by the <i>Tag</i> parameter and positions the editor at its definition. To use this flag, you must first create a database of function names and their locations using the ctags command.
-w <i>Number</i>	Sets the default window size to the value specified by the <i>Number</i> parameter. This is useful when your terminal communicates with the system running the editor over a slow communications line.
-y	Overrides the maximum line setting of 1,048,560 with any value greater than 1024.
+ [<i>Subcommand</i>]	Carries out the ex editor subcommand specified by the <i>Subcommand</i> parameter before viewing with vi begins. If you do not specify a subcommand, the cursor is placed on the last line of the file.

viosupgrade command

Purpose

To back up the virtual and logical configuration data, install the specified image, and restore the virtual and logical configuration data of the Virtual I/O Server (VIOS).

Syntax

To perform the **bosinst** type of upgrade operation, use the following syntax:

```
viosupgrade -t bosinst -n hostname -m ios_mksysbname  
-p spotname { -a RootVGCloneddisk: ... | -r RootVGInstallDisk: ... | -s } [ -c ] [ -e  
resources: ... ] [ -F skipclusterstate ] [ -v ]
```

To perform the **altdisk** type of upgrade operation, use the following syntax:

```
viosupgrade -t altdisk -n hostname -m ios_mksysbname  
-a RootVGInstallDisk: ... [ -c ] [ -e  
resources: ... ] [ -F Options: ... ] [ -v ] [ -g filename ] [ -P post_install_script ] [  
-k pre_restore_script ] [ -X options: ... ] [ -u ] [ -bg ] [ -noprompt options,... ]
```

To perform a **bosinst** or **altdisk** type of upgrade operation across multiple nodes, use the following syntax:

```
viosupgrade -t { bosinst | altdisk } -f filename [ -v ]
```

To check the status of the triggered upgrade operation, use the following syntax:

```
viosupgrade -q { [ -n hostname | -f filename ] } [ -S ]
```

To create the `ios_mkysb` image file from the International Organization for Standardization (ISO) image files, use the following syntax:

```
viosupgrade -I ISOImage1:ISOImage2 -w directoryPath  
-x iosmkysbResourceName [ -y spotResourceName ]
```

Description

When you run the **viosupgrade** command, the following operations are performed in the background:

Backup

Backs up the virtual and logical configuration data to ensure that the VIOS partition can be recovered after a new installation.

Installation

Installs a new VIOS partition from the provided VIOS image.

Restore

Restores the virtual and logical configuration data of the VIOS partition.

Notes:

- When you use the **viosupgrade** command, you can install and configure the logical partition only once during the upgrade operation.
- When you run the **viosupgrade** command, by default the following security configurations are saved and restored:
 - User accounts and groups with login attributes and password.
 - Role based access control (RBAC) configurations such as user roles, authorizations, privileged commands database (privcmds), privileged device database (privdevs), privileged files (privfiles), and the domains.
 - Login and authentication configurations.
 - AIX audit sub system configurations.
 - Trusted execution policies and the databases.
 - IPSec filters and the tunnel configurations.
 - Lightweight directory access protocol (LDAP) client configurations.
 - Open Secure Sockets Layer (OpenSSL), Open Secure Shell protocol (OpenSSH), and the Kerberos configurations.
- Some of the security configurations are restored from the previous version of VIOS. Some new security configurations are activated on the latest version of VIOS. For the remaining security configurations, the old and the new versions are merged.
- Security configurations for which both the previous and current versions are not used are saved in the `/usr/ios/security/saveconf` directory. You can review and merge the configuration files manually.

Flags

Flag name	Description
-a	Specifies one or more alternative disks for the installation. Based on the type of installation, one of the following actions is performed: bosinst type of installation: The provided disks are used to clone the current rootvg disks. After the completion of the migration process, the current rootvg disk is installed with the provided image. The provided disks are at the VIOS level before the migration process. altdisk type of installation: The provided disks are used to install the provided image. The current rootvg disk on the VIOS partition is not impacted during the installation process. The VIOS partition remains in the running state during the installation of the alternative disk.
-c	Specifies that the cluster-level backup and restore operations are performed. Note: The -c flag is mandatory for the VIOS that is part of a Shared Storage Pool (SSP) cluster.
-e	Specifies the configuration resources to be applied after the installation. The valid values for this attribute are resolv_conf, script, fb_script, file_res, image_data, and log. Notes: • The file_res option is applicable only to bosinst type of installations and is not supported for altdisk type of installations. • If you do not specify the -e flag or if the <code>/etc/resolv.conf</code> and <code>/etc/hosts</code> files are not included in the specified configuration resources, the upgrade process backs up these files from the current system before the installation process. The backed-up files are restored during the migration operation.
-f	Specifies the file name that contains the list of VIOS nodes. The values and fields in the file must be specified in a particular sequence and format. The details of the format are specified in the <code>/usr/samples/nim/viosupgrade.inst</code> file and they are separated by a comma. The maximum number of nodes that can be installed through the -f option is 30. The VIOS images are installed on the nodes simultaneously. The installation status is displayed for each node. For an SSP cluster, you must run the viosupgrade command on individual nodes. Out of the n number of nodes in the SSP cluster, maximum n-1 nodes can be upgraded at the same time. Hence, you must ensure that at least one node is always active in the cluster and is not a part of the upgrade process. Note: Enter all the information in the specified format. Do not specify any value for blank or optional fields. The fields that are blank in alternate vg and rootvg disk indicates the SKIP option from the user.

Flag name

>|**-F options**|<

Description

>|Overrides certain default parameters to proceed with the **viosupgrade** installation.

The following options are supported for the **viosupgrade** installation:

skipclusterstate

Skips the verification of the SSP cluster state so that the VIOS upgrade operation can be triggered on multiple VIOS nodes simultaneously.

Note: Ensure that all SSP cluster nodes are not in the DOWN state at the same time. Otherwise, the SSP cluster services might become inactive permanently.

skipdevname

By default, the **viosupgrade** command allows you to preserve the device names in the new rootvg volume group. This option allows you to override the default behavior, and not preserve the device names. By default, the following device names are preserved:

- vhost, vfchost adapter devices
- hdisk device names
- fcnvme, nvme devices
- fscsi devices
- iSCSI devices
- Virtual and physical network adapter devices
- Shared Ethernet Adapters (SEA)
- Virtual local area network (VLAN) devices

Note: The device names are presented by default, and the **skipdevname** option is supported in the VIOS version 4.1.1.00 and later. For previous versions, use the **-devname** option to preserve the device names.

forcecopy

Copies the list of files that are specified in the filename with the **-g** flag, to the respective paths in the newly installed rootvg disks.

Notes:

- The file name that is specified with the **-F forcecopy** option must be in the **-F forcecopy=<file>** format.
- The **forcecopy** option is applicable without the **-g** flag only if a file is specified with the **-F forcecopy=<filename>** option.
- If both the **-g** flag and the **-F forcecopy=<filename>** option are specified with a filename that has a list of files, the files with the **-g** flag are copied to the /home/padmin/backup_files directory. The files with the **-F forcecopy=** option are copied to their respective directories.
- Do not use the **forcecopy** option for copying system-level files.
- The **-g** flag, without the **-F forcecopy** option, copies the backup files to the /home/padmin/backup_files directory in the new rootvg volume group. Merge the contents of the backup files from the /home/padmin/backup_files directory into the corresponding directories of the new rootvg volume group.

|<

Flag name	Description
-F options (Continued)	skiprestore Skips the restore operation of the VIOS metadata configuration after the viosupgrade command installation operation is completed and the VIOS is booted from the newly installed rootvg disk. Notes: • Backup file details are displayed in the output of the viosupgrade command. You can use the backup file to manually restore the VIOS metadata configuration. • To restore the VIOS metadata configuration, run the viosupgrade -o restore -A args= <viosbr arguments> command. The VIOS might restart during the restore process. • Alternatively, you can restore the VIOS metadata configuration by using the viosbr -restore command. The VIOS must be manually rebooted. skipseccfg Skips the restore of the VIOS Security configuration files.
> -g filename <	Specifies the file that contains the list of files that must be backed up from the current system and saved in the new VIOS installed image. Each line must contain a single filename along with its path. Multiple files must be specified in separate new lines.
-I	Specifies the ISO image files that must be used to create the ios_mksysb resource.
> -k pre_restore_script <	Specifies the script name that runs before the system configuration restoration. Notes: • You cannot use the -k flag with the -F skiprestore option. • If there are dependencies that are associated with the <i>pre_restore_script</i>, you must ensure that those files or dependencies are copied to the newrootvg volume group by using the -g flag or through the script mount path. • If a nonzero value is returned from the <i>pre_restore_script</i>, the upgrade operation exits with a failure message. You must resolve the issue and rerun the operation by using -o rerun option on the target VIOS machine.
-m	Specifies the IOS_MKSYSB resource name on the Network Installation Management (NIM) Master server for the specified VIOS installation.
-n	Specifies the target VIOS host name to perform the VIOS upgrade operation.
-p	Specifies the resource object name of the Shared Product Object Tree (SPOT) for NIM installation.
> -P post_install_script <	Specifies and runs a script file after the VIOS metadata configuration restore process is successful. Note: You cannot use the -P flag with the -F skiprestore option.
-q	Queries the status of the VIOS upgrade operation.
-r	Specifies a new rootvg disk where the specified image must be installed. If you specify the -r flag, the existing rootvg disks are not used and the new disk is used for the installation.

Flag name	Description
> -s <	Provides the detailed summary of the VIOS upgrade operation. You must use the -q option with the -s flag.
-s	Skips the cloning of current rootvg disks to alternative disks and continues with the VIOS installation on the current rootvg disk. If the storage disks are not available, you can specify the -s flag to continue with the installation.
-t	Specifies the type of installation from the NIM server. The supported types of installation are bosinst and altdisk . bosinst Indicates a new installation on the current rootvg disk. altdisk Indicates a new installation on the alternative disk. The new installation does not impact the current rootvg disk on the VIOS partition. The VIOS partition that has the current rootvg disk remains in the running state during the installation of the alternative disk.
> -u <	Specifies that the VIOS must be manually rebooted to activate the installed root volume group in the VIOS partition. If the VIOS is in an SSP cluster, the cluster services are down until you restart the VIOS. Note: The VIOS must be rebooted to complete the VIOS metadata restore process in the newly installed root volume group.
-v	Validates whether the VIOS hosts are ready for the installation. The -v flag must be specified only for validation and can be used only for preview of the installation image.
> -bg <	Specifies the command to be run in the background.
-w	Specifies the directory path to create new ios_mkysyb image file.
-x	Specifies the ios_mkysyb image file name that must be created.
> -X options <	Specifies the list of system files or configuration settings that must be excluded from the mkysyb image installation. You can exclude the following file types: PROFILE Indicates the /usr/ios/cli/.profile file. ROOTCRON Indicates the root Cron tab entries. KSHRC Indicates the home/padmin/.kshrc file. SSHKEY Indicates the Secure Shell (SSH) keys of the padmin user that are used in the system files. PASSWD Indicates the padmin password. ROOTUSRFS Indicates the user file systems on the rootvg disk. TUNABLES Indicates the /etc/tunables tunables.
-y	Specifies a new SPOT resource that must be created from the ios_mkysyb resource.

Flag name

>|**-noprompt**
options|<

Description

Overrides the user prompts to proceed with the VIOS upgrade installation operation. When the **-noprompt** flag is used, the **viosupgrade** command automatically accepts all the user prompts. You can specify the following options with the **-noprompt** flag to accept or reject the individual user prompts:

additionalfs=Y|N

You can use **additionalfs=Y|N** option to accept or reject the installation when the **viosupgrade** command finds additional filesets in the current VIOS node or in the mksysb image file that contains the new VIOS node. The following filesets are checked during the VIOS upgrade installation operation:

- **devices.fcp.disk.ibm.mpio**
- **devices.sddpcm**
- **EMC.CLARiiON.fcp.MPIO.rte**
- **EMC.Symmetrix.fcp.MPIO.rte**

skiplv=Y|N

You can use **skiplv=Y|N** option to accept or reject the installation when the **viosupgrade** command finds logical volumes that you cannot migrate. The logical volumes that do not have a mount point or the logical volumes that are of the type Journaled File System (JFS) cannot be migrated.

Note: You can use the **additionalfs=Y|N** and **skiplv=Y|N** options together separated by a comma. If any one option is specified, you must specify the other option as part of the upgrade process.

Exit Status

Return code

Description

0

Indicates success

1

Indicates failure

Requirements

The following requirements must be met before you use the **viosupgrade** command:

- Alternate disks that are used with the **-a** and **-r** options as part of the **viosupgrade** command must be free. You must be able to list the disks by running the **lspv -free** command on the VIO Server.
- Installations through the **viosupgrade** command are categorized as new and complete installation. Any customized configurations on the current system that were running before the installation, including the time zone, are not carried to the new installation image. If you need to copy a customized file to the new image, use the **-e** flag with the **file_res** option. The **-e** flag with the **file_res** option specifies the customized file that must be backed up after the installation. You can use the **viosupgrade** command with the **-e** flag and **file_res** option only for the **bosinst** type of installations. After the installation, the customized files are copied to the **/home/padmin/backup_files** directory.

Note: Follow the manual steps on the NIM master to define the **file_res** resources before you run the **viosupgrade** command. The customized files are stored on the NIM master in the source directory **/export/nim/viosupgrade/copyfiles**. The customized files are copied on the VIOS after the installation in the destination directory **/home/padmin/backup_files**.

For example, to restore the files in the **/etc/environment** and **/var/custom.conf** directories, perform the following steps:

1. Create a source directory.

```
mkdir -p /export/nim/viosupgrade/copyfiles
```

2. Define the **file_resource** option.

```
nim -o define -t file_res -a location=/export/nim/viosupgrade/copyfiles -a  
dest_dir=/home/padmin/ backup_files -a server=master file_res_user
```

3. Create the /export/nim/viosupgrade/copyfiles/etc and /export/nim/copyfiles/var directories under the source directory.

```
mkdir -p /export/nim/viosupgrade/copyfiles/etc  
mkdir -p /export/nim/copyfiles/var.
```

4. Copy the files from VIOS to the NIM master source directories.

```
scp -r root jaguar13:/etc/environment /export/nim/viosupgrade/copyfiles/etc  
scp -r root jaguar13:/var/custom.conf /export/nim/viosupgrade/copyfiles/var
```

- The **viosbr** restore process does not support the virtual device mappings with **vscsi** disks that are created on the **rootvg** disks of the VIO server. Therefore, the **viosupgrade** command cannot restore the **vscsi** mappings if logical volumes (LVs) are created from the **rootvg** disk of VIO server.
- To enable the VIOS to be managed remotely by the AIX NIM master, run the **remote_management** command from VIOS.
- Considerations for the target **ios_mkysb** image level are as follows:
 - The level of the target **ios_mkysb** image must be at 3.1.0.00 level, or later.
 - The target **ios_mkysb** image level must be higher than the current VIOS **rootvg** level.
- The **viosupgrade** command on NIM server is supported from IBM AIX 7.2 with Technology Level 3, or later.
- For the NIM **bosinst** type of installation, the following VIOS levels are supported:
 - 2.2.6.30, or later for an SSP Cluster environment
 - 2.2.x.x, or later for a non-SSP Cluster environment
- On the NIM master, VIOS must be defined along with the Network Adapter Hardware Address (MAC) or Network Adapter Logical Device Name (ent name) in the **if1** definition of NIM object. If not defined, the **viosupgrade** command displays a message that the network boot operation might be delayed or fail.
- For a NIM **bosinst** type of installation, the interface resource (MAC or ent name), defined on the **if1** definition of NIM object for any given VIOS, must adhere to the following considerations:
 - The interface resource can be any available Ethernet interface that is connected to the network.
 - The interface resource can be a physical interface that is part of a SEA.
 - The interface resource cannot be a SEA interface.

This is mandatory for the **viosupgrade** command to trigger the VIOS restore process after the installation.

- For a NIM **altdisk** type of installation, the current VIOS levels that are supported are 2.2.6.30, or later.
- The NIM **altdisk** type of installation is not supported for VIOS when an SEA is the only primary interface for communication.

Note: To enable the NIM **altdisk** type of installation for the preceding case, you must configure an additional Ethernet interface as the primary interface for remote management. The NIM master must be able to establish the network connection with the VIOS by using the IP address that is configured on this interface by default.

- If a node is part of a cluster, then the hostname of the node must be resolvable during the metadata restore process after the installation is complete. This can be achieved either by passing the **resolv.conf** file or a script to add an entry in the **/etc/hosts** file, along with the **-e** option.

- If the VIOS belongs to an SSP cluster and if the current VIOS version is older than 2.2.6.30, the following two-step upgrade process is necessary to upgrade to VIOS 3.1.0.00, or later.
 1. Upgrade the VIOS to version 2.2.6.30 through the upgrade methods such as by using the **updateios** command.
 2. Use the **viosupgrade** command to upgrade to the VIOS to version 3.1.0.00, or later.
- If you are using **altdisk** type of installation for the VIOS, and if the VIOS has `altinst_rootvg` or `old_rootvg` disks, the **viosupgrade** command fails to indicate the user to rename the disks.
- The NIM master might restart the VIOS and reinitiate the restore process to restore multiple virtual IO mappings through the `viosbr` restore process after the installation.
- If the NIM master fails to restore the mappings, you must manually reinitiate the restore operation on the NIM master by using the following commands:

– VIOS-non-SSP Cluster:

```
nim -o viosbr -a viosbr_action=restore -a ios_backup=<BackupObjectName> <VIOSObjectName>
```

– VIOS-SSP Cluster:

```
nim -o viosbr -a viosbr_action=restore -a ios_backup=<BackupObjectName> -a  
clustername=<clusterName> -a viosbr_flags="-curnode" <VIOSObjectName>
```

Note: The *BackupObjectName* is `<VIOSName>_backup`. You can get the list of backup object names from the command, **lsnim -t ios_backup**.

- If you installed additional software on the VIOS that is not a part of the base VIOS image, the **viosupgrade** command might fail to restore software-related configurations. Create a customized VIOS image with the software applications that you want to include and provide this customized VIOS image as an input to the **viosupgrade** command for installation.

The **viosupgrade** command identifies the software applications that are not included in the `ios_mksysb` image file, but are installed in the current system, such as the Subsystem Device Driver (SDD) or the Subsystem Device Driver Path Control Module (SDDPCM). The **viosupgrade** command displays a warning prompt on the console output during software validation or before software installation. You can choose to continue or stop the upgrade process by selecting the corresponding option. For example, if the software application that is installed on the VIOS is SDDPCM, you must customize the `ios_mksysb` image file to include the SDDPCM application and to provide the customized VIOS image file as an input to the **viosupgrade** command.

Note: For more information about creating a customized VIOS image, see **backupios** command.

- To create the `ios_mksysb` image file from the ISO image files, you require approximately 4 GB memory in the directory that you specify by using the **-w** flag in the **viosupgrade** command.

Examples

1. To validate the VIOS upgrade operation by using the **bosinst** type of installation, enter the following command:

```
viosupgrade -v -t bosinst -n systemA -m mkysbA -p spotA -s
```

Where, target VIOS node is `systemA`, `mkysbA` image name is `mkysbA`, spot name is `spotA` and the type of install is **bosinst**.

2. To validate the VIOS upgrade operation by using the **bosinst** method, enter the following command:

```
viosupgrade -v -t bosinst -n clusternodeA -m mkysbA -p spotA -r hdisk1:hdisk2 -c -e resolv_conf:script
```

Where, target cluster VIOS node is `clusternodeA`, `mkysbA` image name is `mkysbA`, spot name is `spotA`, NIM resources are `resolv_conf` and `script`, type of install is **bosinst** and provided new rootvg disks are `hdisk1` and `hdisk2`.

3. To perform the VIOS upgrade operation by using the **bosinst** method by skipping the current rootvg cloning, enter the following command:

```
viosupgrade -t bosinst -n systemA -m mksysbA -p spotA -s
```

Where, target VIOS node is systemA, mksysb image name is *mksysbA*, spot name is *spotA* and type of install is **bosinst**.

Note: After the installation, the new rootvg disk will be the current rootvg disk.

4. To perform the VIOS upgrade operation by using the **bosinst** method by installing on provided disks, enter the following command:

```
viosupgrade -t bosinst -n systemA -m mksysbA -p spotA -r hdisk1:hdisk2
```

Where, target VIOS node is systemA, mksysb image name is *mksysbA*, spot name is *spotA*, type of install is **bosinst** and provided new rootvg disks are hdisk1 and hdisk2.

Note: After reinstall the new rootvg disk is on the provided disks.

5. To perform the VIOS upgrade operation by using the **bosinst** method by backing up the current rootvg disk on provided alternate disks, enter the following command:

```
viosupgrade -t bosinst -n systemA -m mksysbA -p spotA -a hdisk3:hdisk4
```

6. To perform the SSP cluster VIOS upgrade operation by using the **bosinst** method on the provided disks, enter the following command:

```
viosupgrade -t bosinst -n clusternodeA -m mksysbA -p spotA -r hdisk1:hdisk2 -c
```

Where, target cluster VIOS node is clusternodeA, mksysb image name is mksysbA, spot name is spotA, type of install is **bosinst** and provided disks for new rootvg are hdisk1 and hdisk2.

7. You can perform the VIOS upgrade operation by using the **bosinst** method and the provided NIM resources. You can also create a backup of the current rootvg on a provided alternate disk. To perform all three of these operations, enter the following command:

```
viosupgrade -t bosinst -n systemA -m mksysbA -p spotA -a hdisk3:hdisk4 -e resolv_conf:script:fb_script
```

Where the target VIOS node is systemA, mksysb image name is mksysbA, spot name is spotA, type of install is **bosinst** and provided alternate rootvg disks are hdisk3 and hdisk4, provided NIM resources are resolv_conf, script, and fb_script.

8. To validate the VIOS upgrade operation by using the **altdisk** method, enter the following command:

```
viosupgrade -v -t altdisk -n systemA -m mksysbA -a hdisk3:hdisk4
```

Where the target VIOS node is systemA, mksysb image name is mksysbA, alternate disks are hdisk3, hdisk4, and the type of install is **altdisk**.

9. To perform the VIOS upgrade operation by using the **altdisk** method, enter the following command:

```
viosupgrade -t altdisk -n systemA -m mksysbA -a hdisk3:hdisk4
```

Where the target VIOS node is systemA, mksysb image name is *mymksysbA*, type of install is **altdisk** and provided alternate rootvg disks are hdisk3 and hdisk4.

10. To perform the SSP cluster VIOS upgrade operation by using the **altdisk** method, enter the following command :

```
viosupgrade -t altdisk -n clusternodeA -m mksysbA -a hdisk3:hdisk4:hdisk5 -c
```

Where the target cluster node is clusternodeA, mksysb image name is mksysbA, type of install is **altdisk** and provided alternate rootvg disks are hdisk3, hdisk4, and hdisk5.

11. To validate the upgrade operation of one or more VIOS nodes in the provided file by using the **bosinst** method, enter the following command:

```
viosupgrade -v -t bosinst -f "/usr/samples/nim/viosupgrade.inst"
```

12. To perform the upgrade operation on a provided VIOS node in file by using the **bosinst** method, enter the following command:

```
viosupgrade -t bosinst -f "/usr/samples/nim/viosupgrade.inst"
```

If the target VIOS node is `systemA`, `mksysb` image name is `mksysbA`, spot name is `spotA`, provided alternate `rootvg` are `hdisk3` and `hdisk4`, type of install is **bosinst** and file that contains VIOS node information is `/usr/samples/nim/viosupgrade.inst`, then the file must contain the following data:

```
systemA, mksysbA, spotA, ,hdisk3:hdisk4
```

13. To perform the upgrade operation on provided SSP cluster VIOS nodes in file through the **bosinst** method, enter the following command:

```
viosupgrade -t bosinst -f "/usr/samples/nim/viosupgrade.inst"
```

If the target cluster VIOS node is `clusternodeA`, `mksysb` image name is `mksysbA`, spot name is `spotA`, type of install is **bosinst** and the file that contains VIOS nodes information is `/usr/samples/nim/viosupgrade.inst`, then the file must contain the following data:

```
clusternodeA, mksysbA, spotA, , , , c,
```

14. To perform the upgrade operation on the provided VIOS nodes in the file by using the **altdisk** method, enter the following command:

```
viosupgrade -v -t altdisk -f "/usr/samples/nim/viosupgrade.inst"
```

15. To perform the VIOS upgrade operation by using the **bosinst** method and to restore files from the current `rootvg` disk to a newly installed image, enter the following command:

```
viosupgrade -t bosinst -n systemA -m mksysbA -p spotA -s -e file_res
```

Where the target VIOS node is `systemA`, `mksysb` image name is `mksysbA`, spot name is `spotA`, **file_res** option is `file_res_user`, and the type of install is **bosinst**.

16. To create the `ios_mksysb` image file and SPOT resource from ISO image files, enter the following command:

```
viosupgrade -I /home/padmin/dvdimage.v1.iso:/home/padmin/dvdimage.v2.iso -w /home/  
myNewIosMksysbImageDir  
-x myIosMksysbRes -y mySPOTResource
```

Where, `/home/padmin/dvdimage.v1.iso` and `/home/padmin/dvdimage.v2.iso` are the ISO image files, `/home/myNewIosMksysbImageDir` is the path where the `ios_mksysb` image file is created, `myIosMksysbRes` is the name of the `ios_mksysb` image file that is created, and `mySPOTResource` is the name of the SPOT resource that is created.

17. >|To perform the VIOS upgrade operation by using the **bosinst** method on specified disks and to override the verification of the SSP cluster state, enter the following command:

```
viosupgrade -t bosinst -n systemA -m mksysbA -p spotA -r hdisk1:hdisk2 -F skipclusterstate  
-c
```

Where the target VIOS node is `systemA`, `mksysbA` is the name of the `mksysb` image, `spotA` is the name of the SPOT resource, the type of installation is **bosinst**, and the specified new `rootvg` disks are `hdisk1` and `hdisk2`.|<

18. To perform the VIOS upgrade operation by using the **altinst** method and to copy the files from the current rootvg disk to a newly installed VIOS image in /home/padmin/backup_files directory, enter the following command:

```
viosupgrade -t altdisk -n systemA -m mksysbA -a hdisk3:hdisk4 -g file_list_name
```

Where the target VIOS node is systemA, the image name of the mksysb is mymksysbA, the type of install is **altdisk**, the alternate rootvg disks are hdisk3 and hdisk4, and the file_list_name file is the file that contains the list of files that needs to be copied to the new rootvg image.

19. To perform the VIOS upgrade operation by using the **altinst** method and to copy the files from the current rootvg disk to the respective directories of a newly installed rootvg disk, enter the following command:

```
viosupgrade -t altdisk -n systemA -m mksysbA -a hdisk3:hdisk4 -g file_list_name -F forcecopy
```

Where the target VIOS node is systemA, the image name of the mksysb image is mksysbA, the type of install is **altdisk**, the alternate rootvg disks are hdisk3 and hdisk4, and the file_list_name file is the file that contains the list of files that needs to be copied to the new rootvg image.

20. To perform the VIOS upgrade operation by using the **altinst** method and to run a script after the restore process of the VIOS metadata configuration is successful on the new rootvg disk, enter the following command:

```
viosupgrade -t altdisk -n systemA -m mksysbA -a hdisk1:hdisk2 -P <post_install_script>
```

Where the target VIOS node is systemA, the image name of the mksysb image is mksysbA, the type of install is **altdisk**, the alternate rootvg disks are hdisk1 and hdisk2, and the post_install_script is the script that you must run.

21. To perform the VIOS upgrade operation by using the **altinst** method and to run a script before the restore process of the virtual devices on the new rootvg disk, enter the following command:

```
viosupgrade -t altdisk -n systemA -m mksysbA -a hdisk1:hdisk2 -k <pre_restore_script>
```

Where the target VIOS node is systemA, the image name of the mksysb image is mksysbA, the type of install is **altdisk**, the alternate rootvg disks are hdisk1 and hdisk2, and the pre_restore_script is the script that you must run.

22. To perform the VIOS upgrade operation by using the **altinst** method and to run the command in the background, enter the following command:

```
viosupgrade -t altdisk -n systemA -m mksysbA -a hdisk1:hdisk2 -bg
```

Where the target VIOS node is systemA, the image name of the mksysb image is mksysbA, the type of install is **altdisk**, and the alternate rootvg disks are hdisk1 and hdisk2.

23. To perform the VIOS upgrade operation by using the **altinst** method to copy the list of files from the file_list_name file to the /home/padmin/backup_files directory, and to copy the list of files from the file_list_name2 file to their respective paths, enter the following command:

```
viosupgrade -t altdisk -n systemA -m mksysbA -a hdisk1:hdisk2 -g file_list_name -F  
forcecopy=file_list_name2
```

Where the target VIOS node is systemA, the image name is mksysbA, the type of install is **altdisk**, and the new rootvg disks are hdisk1 and hdisk2. The file_list_name file contains the list of files that needs to be copied to the new rootvg image in the /home/padmin/backup_files directory. The file_list_name2 file contains the list of files that needs to be copied to the new rootvg image in their respective paths.

24. To use the **-noprompt** flag without the additionalfs=Y|N and the skip1v=Y|N options, enter the following command:

```
viosupgrade -t altdisk -n systemA -m mksysbA -a hdisk1:hdisk2 -noprompt
```

Where the target VIOS node is systemA, the image name is mksysbA, the type of install is **altdisk**, and the new rootvg disks are hdisk1 and hdisk2.

25. To perform the VIOS upgrade operation by using the **altinst** method and to use the **-noprompt** flag with the **additionalfs=Y|N** and **skiplv=Y|N** options, enter the following commands:

```
viosupgrade -t altdisk -n systemA -m mksysbA -a hdisk1:hdisk2 -noprompt  
skiplv=Y,additionalfs=N
```

```
viosupgrade -t altdisk -n systemA -m mksysbA -a hdisk1:hdisk2 -noprompt skiplv=Y
```

Where the target VIOS node is systemA, the image name is mksysbA, the type of install is **altdisk**, and the new rootvg disks are hdisk1 and hdisk2.

26. To verify the VIOS upgrade operation by using the **altinst** method to copy the list of files in the **file_list_name2** file to their respective paths, and to skip the restore operation on the new rootvg disks, enter the following command:

```
viosupgrade -v -t altdisk -n systemA -m mksysbA -a hdisk1:hdisk2 -F  
forcecopy=file_list_name2:skiprestore
```

Where the target VIOS node is systemA, the image name is mksysbA, the type of install is **altdisk**, the new rootvg disks are hdisk1 and hdisk2, and the **file_list_name2** file is the file that contains the list of files that needs to be copied to the new rootvg image in their respective paths.

27. To verify the VIOS upgrade operation by using the **altinst** method to exclude the system files like the **SSHKEYS**, and the root Cron tab entries, enter the following command:

```
viosupgrade -v -t altdisk -n systemA -m mksysbA -a hdisk1:hdisk2 -X SSHKEYS:ROOTCRON
```

Where the target VIOS node is systemA, the image name is mksysbA, the type of install is **altdisk**, and the new rootvg disks are hdisk1 and hdisk2.

vmh Command

Purpose

Starts a visual interface for use with MH commands.

Syntax

vmh [**-prompt** *String*] [**-vmhproc** *CommandString* | **-novmhproc**]

Description

The **vmh** command starts a visual interface for use with MH commands. The **vmh** command implements the server side of the MH window management protocol and maintains a split-screen interface to any program that implements the client side of the protocol.

The **vmh** command prompts for commands and sends them to the client side of the protocol. If the command produces a window with more than one screen of output, the **vmh** command prompts the user for a subcommand. The **vmh** subcommands enable you to display specific portions of the command output.

vmh Subcommands

Item	Description
Ctrl-L	Refreshes the screen.
Space	Advances to the next screen.

Item	Description
[<i>Number</i>] Enter	Advances the specified number of lines. The default is one line.
[<i>Number</i>] d	Advances 10 times the specified number of lines. The default for the <i>Number</i> variable is 1, for a total of 10 lines.
[<i>Number</i>] g	Goes to the specified line.
[<i>Number</i>] G	Goes to the end of the window. If the <i>Number</i> variable is specified, this command acts like the g flag.
[<i>Number</i>] u	Goes back 10 times the specified number of lines. The default for the <i>Number</i> variable is 1, for a total of 10 lines.
[<i>Number</i>] y	Goes back the specified number of lines. The default is one line.
h	Displays a help message.
q	Ends output.

Flags

Item	Description
-help	Lists the command syntax, available switches (toggles), and version information. Note: For MH, the name of this flag must be fully spelled out.
-novmproc	Runs the default vmproc without the window management protocol.
-prompt <i>String</i>	Uses the specified string as the prompt.
-vmhproc <i>CommandString</i>	Specifies the program that implements the client side of the window management protocol. The default is the msh program.

Profile Entries

The following entries are entered in the *UserMhDirectory/.mh_profile* file:

Item	Description
Path:	Specifies the user's MH directory.
mshproc:	Specifies the program used for the MH shell.

Files

Item	Description
\$HOME/.mh_profile	Contains the MH user profile.
/usr/bin/vmh	Contains the vmh command.

vmo Command

Purpose

Manages Virtual Memory Manager tunable parameters.

Syntax

vmo [**-p** | **-r** [**-K**]] [**-y**] { **-o** *Tunable* [= *Newvalue*] }

vmo [**-p** | **-r** [**-K**]] [**-y**] { **-d** *Tunable* }

vmo [**-p** | **-r** [**-K**]] [**-y**] **-D**

vmo [**-p** | **-r** | **-K**] [**-F**] **-a**

vmo **-h** [*Tunable*]

vmo [**-F**] [**-K**] **-L** [*Tunable*]

vmo [**-F**] [**-K**] **-x** [*Tunable*]

Note: Multiple **-o**, **-d**, **-x**, and **-L** are allowed.

Description

Note: The **vmo** command can only be run by root. The **vmo** command is a self-documenting command. The information about some of the flags or tunable parameters might be missing or out-of-date. You can find an up-to-date list of all the flags and by using the **-h**, **-L**, or **-x** flag.

Use the **vmo** command to configure Virtual Memory Manager tuning parameters. This command sets or displays current or next boot values for all Virtual Memory Manager tuning parameters. This command can also make permanent changes or defer changes until the next reboot. Whether the command sets or displays a parameter is determined by the accompanying flag. The **-o** flag specifies both actions. It can either display the value of a parameter or set a new value for a parameter.

The Virtual Memory Manager (VMM) maintains a list of free real-memory page frames. The page frames are available to hold virtual-memory pages that are needed to satisfy a page fault. When the number of pages on the free list falls below the values that are specified by the `minfree` parameter, the VMM steals pages to add to the free list. The VMM continues to steal pages until the free list has at least the number of pages that are specified by the `maxfree` parameter.

If the number of file pages (permanent pages) in memory is less than the number specified by the `minperm%` parameter, the VMM steals frames from either computational or file pages, regardless of repage rates. If the number of file pages is greater than the number specified by the `maxperm%` parameter, the VMM steals frames only from file pages. Between the two, the VMM normally only steals file pages, but if the repage rate for file pages is higher than the repage rate for computational pages, computational pages are stolen as well.

You can also modify the thresholds that are used to decide when the system is running out of paging space. The `npswarn` parameter specifies the number of paging-space pages available at which the system begins warning processes that paging space is low. The `npskill` parameter specifies the number of paging-space pages available at which the system begins stopping processes to release paging space.

Note: Options **-o**, **-d**, and **-D**, which attempt to change the value of a virtual memory manager tunable parameter, are not supported within a workload partition.

Understanding the Effect of Changing Tunable Parameters

Misuse of this command can cause performance degradation or operating-system failure. Before you experiment with the **vmo** command, familiarize yourself with both [Performance overview of the Virtual Memory Manager](#) and [Enhanced JFS file system cache limit with the `maxclient` parameter](#).

Before modifying any tunable parameter, you must first carefully read about all its characteristics in the [Tunable Parameters](#) section to fully understand its purpose.

You must then make sure that the Diagnosis and Tuning sections for this parameter truly apply to your situation and that changing the value of this parameter could help improve the performance of your system.

If the Diagnosis and Tuning sections both contain only "N/A", you must not change this parameter unless directed by AIX development.

Flags

Item	Description
-a	Displays current value, reboot value (when used with the -r option), Live Update value (when used with the -K option), or permanent values for all tunable parameters (when used with the -p option), one per line in pairs <i>Tunable = Value</i> . For the permanent option, a value is displayed only for a parameter if its reboot and current values are equal. Otherwise, NONE is displayed as the value.
-d <i>Tunable</i>	Resets the <i>Tunable</i> parameters to its default value. If a <i>Tunable</i> parameter, which must be changed because it is not set to its default value, meets one or more of the following sets of criteria, a warning message is displayed and no change is made to the parameter: • The tunable parameter is of type Bosboot or Reboot. • The tunable parameter is of type Incremental and was changed from its default value, and the -r flag is not used in combination. • > The tunable parameter is of type Reboot and is supported throughout the Live Update operation. <
-D	Resets all <i>Tunable</i> parameters to their default values. If <i>Tunables</i> that need to be changed because they are not set to their default values meet one or more of the following sets of criteria, a warning message is displayed and no change is made: • The tunable is of type Bosboot or Reboot. • The tunable is of type Incremental and was changed from its default value, and -r is not used in combination. • > The tunable parameter is of type Reboot and is supported throughout the Live Update operation. <
-F	Forces display of the restricted tunable parameters when the -a , -L or -x options are specified alone on the command line to list all tunables. When the -F flag is not specified, restricted tunables are not displayed, unless these restricted tunables are named with a display option.
-h [<i>Tunable</i>]	Displays help about the <i>Tunable</i> parameter if it is specified. Otherwise, displays the usage statement of the vmo command.
> > <-K	Sets the tunable parameter value in both <code>/etc/tunables/nextboot</code> and <code>/etc/tunables/nextliveupdate</code> files. The -K flag can be used only with the -r flag. When you specify the -K flag with the -r and -d (or -D) flags, the tunable parameter value is set to its default value in the <code>/etc/tunables/nextboot</code> and <code>/etc/tunables/nextliveupdate</code> files to be used during the next boot or Live Update operations. > When you specify the -K flag with the -L or -x flag, the vmo command displays the Live Update values. < < <

Item	Description
-L [<i>Tunable</i>]	Lists the characteristics of one or all tunable parameters, one per line, in the following format. If you specify the -K flag with the -L flag, the Live Update values are also displayed. <pre> NAME CUR DEF BOOT MIN MAX UNIT TYPE ----- --- vmm_klock_mode 2 -1 2 -1 3 numeric B ----- --- ... where: CUR = current value DEF = default value BOOT = reboot value MIN = minimal value MAX = maximum value UNIT = tunable unit of measure TYPE = parameter type: D (for Dynamic), S (for Static), R for Reboot), B (for Bosboot), M (for Mount), I (for Incremental), C (for Connect), and d (for Deprecated) DEPENDENCIES = list of dependent tunable parameters, one per line </pre>
-o <i>Tunable</i> [= <i>Newvalue</i>] -p	Displays or sets the value of a tunable parameter to a new value. > The /etc/tunables/usermodified file is updated with the new tunable parameter value whenever you use the -o flag to change the value of a dynamic tunable. < If a tunable parameter that you want to change meets one or more of the following sets of criteria, a warning message is displayed and no change is made to the parameter: • The tunable parameter is of type Bosboot or Reboot. • The tunable parameter is of type Incremental, and its current value is greater than the specified value, and the -r flag is not used in combination. • The tunable parameter is of type Reboot, and the tunable parameter is supported throughout the Live Update operation. When you specify the -r flag with the -o flag without specifying a new value, the next boot value for tunable is displayed. > When you specify the -K flag with the -o flag without specifying a new value, the next Live Update value for the tunable parameter is displayed. < When you specify the -p flag with the -o flag without specifying a new value, a value is displayed only if the current and next boot values for the tunable are the same. Otherwise, NONE is displayed as the value. When used in combination with -o, -d or -D, makes changes apply to both current and reboot values, that is, turns on the updating of the /etc/tunables/nextboot file in addition to the updating of the current value. These combinations cannot be used on Reboot and Bosboot type parameters because their current value can't be changed. When used with -a or -o without specifying a new value, values are displayed only if the current and next boot values for a parameter are the same. Otherwise NONE is displayed as the value.

Item	Description
-r	When the -r flag is used with the -a or -o options without specifying a new value, the values are displayed only if the current and next boot values for a parameter are the same. Otherwise, NONE is displayed as the value. The -r flag changes the reboot values when it is used with the -o, -d, or -D flags. For example, you can update the <code>/etc/tunables/nextboot</code> file when you use the -r flag. If any parameter of type Bosboot is changed, the user is prompted to run the bosboot command. When used with the -a or the -o flags without specifying a new value, next boot values for tunable parameters are displayed instead of current values. > When used with the -K flag, changes in the tunable parameter of type Reboot apply to both the <code>/etc/tunables/nextboot</code> and <code>/etc/tunables/nextliveupdate</code> files.<
-x [<i>Tunable</i>]	Lists characteristics of one or all tunable parameters, one per line, using the following (spreadsheet) format. If you specify the -K flag with the -x flag, the Live Update values are also displayed.

```
tunable,current,default,reboot,min,max,unit,type,{dtunable}

where:
  current = current value
  default = default value
  reboot = reboot value
  min = minimal value
  max = maximum value
  unit = tunable unit of measure
  type = parameter type: D (for Dynamic), S (for Static), R (for Reboot),
                        B (for Bosboot), M (for Mount), I (for Incremental),
                        C (for Connect), and d (for Deprecated)
  dtunable = list of dependent tunable parameters
```

-y	Suppresses the confirmation prompt before running the bosboot command.

If a restricted tunable parameter is changed, a warning message is displayed that indicates that a tunable of the restricted use type has been modified. If the **-r** or the **-p** option is specified, you are prompted to confirm the change. In addition, at system reboot, restricted tunables that are displayed in the `/etc/tunables/nextboot` file and are changed to values that are different from their default values (using a command line specifying the **-r** or **-p** option) causes an error log entry that identifies the list of these changed tunables.

When modifying tunable, the tunable value might be specified using abbreviations such as K, M, G, T, P and E to indicate units. See the following lists for abbreviations and their correspondent values:

- K=2¹⁰
- M=2²⁰
- G=2³⁰
- T=2⁴⁰
- P=2⁵⁰
- E=2⁶⁰

Thus, a tunable value of 1024 might be specified as 1K.

Any change (with **-o**, **-d** or **-D**) to a parameter of type Mount will result in a message being displayed to warn the user that the change is only effective for future mountings.

Any change (with **-o**, **-d** or **-D** flags) to a parameter of type Connect will result in **inetd** being restarted, and a message displaying a warning to the user that the change is only effective for future socket connections.

Any attempt to change (with **-o**, **-d** or **-D**) a parameter of type **Bosboot** or **Reboot** without **-r**, will result in an error message.

Any attempt to change (with **-o**, **-d** or **-D** but without **-r**) the current value of a parameter of type Incremental with a new value smaller than the current value, will result in an error message.

Tunable Parameters Type

All the tunable parameters manipulated by the tuning commands (**no**, **nfso**, **vmo**, **ioo**, **raso**, and **schedo**) have been classified into these categories:

Item	Description
Dynamic	If the parameter can be changed at any time
Static	If the parameter can never be changed
Reboot	If the parameter can only be changed during reboot
Bosboot	If the parameter can only be changed by running bosboot and rebooting the machine
Mount	If changes to the parameter are only effective for future file systems or directory mounts
Incremental	If the parameter can only be incremented, except at boot time
Connect	If changes to the parameter are only effective for future socket connections
Deprecated	If changing this parameter is no longer supported by the current release of AIX.

For parameters of type Bosboot, whenever a change is performed, the tuning commands automatically prompt the user to ask if they want to execute the **bosboot** command. For parameters of type Connect, the tuning commands automatically restart the **inetd** daemon.

Note that the current set of parameters managed by the **vmo** command only includes Static, Dynamic, and Bosboot types.

Compatibility Mode

When running in compatibility mode (controlled by the **pre520tune** attribute of **sys0**), reboot values for parameters, except those of type **Bosboot**, are not meaningful because in this mode they are not applied at boot time. For more information, see *Performance management*.

In compatibility mode, you can set reboot values to tuning parameters by imbedding calls to tuning commands in scripts called during the boot sequence. Parameters of type **Reboot** can be set without using the **-r** flag, so that existing scripts continue to work.

Tunable Parameters

To view the default and range of values allowed for the tunables, run the **vmo** command with the **-h** option as follows:

```
vmo -h <tunable_parameter_name>
```


Tunable	Description
ame_cpus_per_pool	Purpose Determines the ratio of CPUs per compressed memory pool. For every <i>ame_cpus_per_pool</i> CPUs, at least one compressed memory pool is created. Tuning Lower ratios are used to reduce contention on compressed memory pools. This ratio is not the only factor used to determine the number of compressed memory pools (amount of memory and the layout is also considered) so certain changes to this ratio may not result in any change to the number of compressed memory pools. This tunable parameter is supported throughout the Live Update operation.
ame_maxfree_mem	Purpose Specifies the average amount of free memory in a compressed memory pool free list at which the VMM will shrink the compressed pool. Tuning Excessive shrink and grow operations can occur if compressed memory pool size tends to change significantly. This can occur if the workload working set size frequently changes. Increase this tunable to raise the threshold at which the VMM will shrink a compressed memory pool and thus reduce the number of overall shrink and grow operations.
ame_min_ucpool_size	Purpose Defines the minimum size of the uncompressed pool. Tuning If compressed memory pool grows too large, there may not be enough space in memory to house decompressed memory that can slow down application performance due to excessive use of the compressed memory pool. Increase this value to limit the size of the compressed memory pool and make more decompressed pages available.

Tunable	Description
ame_minfree_mem	Purpose Specifies the amount of free memory in a compressed memory pool free list at which the VMM will grow the compressed pool. Tuning If processes are being delayed waiting for compressed memory to become available, increase <i>ame_minfree_mem</i> to improve response time. Note, that this must be at least 64 KB less than <i>ame_maxfree_mem</i>.
ame_mpsize_support	Purpose Enables all supported page sizes in an Active Memory Expansion (AME) environment for POWER8 processor-based servers, or later, which supports the 64 KB accelerator. Tuning A value of -1 allows the AIX operating system to choose the optimal mode for the AME environment. A value of 0 enables the page sizes of 4 KB and 16 MB only. A value of 1 enables all supported page sizes in an AME environment. You can change this tunable parameter only in POWER8 processor-based servers, or later, which supports the 64 KB accelerator.
ams_loan_policy	Purpose This tunable toggles the loaning behavior when shared memory mode is enabled. Tuning When the tunable is set to 0, loaning is disabled. When set to 1, loaning of file cache is enabled. When set to 2, loaning of any type of data is enabled. In response to low memory in the AMS pool, the VMM will free memory and loan it to the hypervisor.

Tunable	Description
> dr_lmb_meta <	Purpose When memory is dynamically added to a logical partition, some physical memory must be isolated and allocated to the VMM data structures to support the newly added memory. This tunable parameter specifies whether the physical memory is allocated from the existing memory or from the newly added memory. Tuning A value of 0 allows the system to determine whether the physical memory must be isolated and allocated to the VMM data structures from the existing memory or newly added memory. A value of 1 enforces the isolation and allocation from the existing physical memory whenever possible. A value of 2 enforces the isolation and allocation of memory from the newly added physical memory. The default value is 0.
> dr_multi_lmb <	Purpose When memory is dynamically added to a logical partition, the system processes multiple LMBs simultaneously to increase performance. This tunable parameter enforces processing of a single LMB at a time. Tuning A value of 0 enforces the system to process a single LMB at a time. A value of 1 allows the system to process multiple LMBs simultaneously. The default value is 1.
force_relalias_lite	Purpose If set to 0, a heuristic is used, when tearing down a mmap region, to determine when to avoid locking the source mmapped segment Tuning This is a scalability tradeoff, controlled by <i>relalias_percentage</i>, possibly costing more compute time used. If set to 1, the source segment lock is avoided whenever possible, regardless of the value of <i>relalias_percentage</i>.

Tunable	Description
lgpg_size	Purpose Specifies the size in bytes of the hardware-supported large pages used for the implementation for the shmget() system call with the SHM_LGPAGE flag. Tuning Supported on systems from POWER4 onwards. Although this parameter is Dynamic on DLPAR-capable systems, the nextboot value is written into the boot image when a bosboot command is issued so that the setting is optimally restored at reboot. The <i>lgpg_regions</i> parameter must be set to a nonzero value in addition to this parameter. The application must be modified to specify the SHM_LGPAGE flag when calling the shmget() subroutine. This will improve the performance in the case where there are many TLB misses and large amounts of memory is being accessed.
low_ps_handling	Purpose Specifies the action to change the system behavior in relation to process termination during low paging space conditions. Tuning A value of 1 indicates current behavior of process termination on low paging space. A value of 2 indicates a new behavior where processes with SIGDANGER handler will be killed, if no other processes were found earlier to recover from low paging space condition.
maxfree	Purpose Specifies the number of frames on the free list at which page-stealing is to stop. Tuning Observe free-list-size changes with vmstat -n command. If the vmstat -n command displays the free-list size frequently driven below <i>minfree</i> by application demands, increase the <i>maxfree</i> value to reduce calls to replenish the free list. Setting the value too high causes page replacement to run for a longer period of time. The difference between <i>maxfree</i> and <i>minfree</i> should be of the order of <i>maxpgahead</i>, and no less than 8.

Tunable	Description
maxpin%	Purpose Specifies the maximum percentage of real memory that can be pinned. Tuning Change if cannot pin memory, although free memory is available. If this value is changed, the new value should ensure that at least 4 MB of real memory will be left unpinned for use by the kernel. The vmo command converts maxpin% to the corresponding maxpin absolute value, which is the value used by the kernel. Change this parameter only in extreme situations, such as maximum-load benchmarking. This dynamic parameter will have its nextboot value written into the boot image if a bosboot command is issued.
memory_frames	Purpose Number of valid memory frames. Tuning N/A
memplace_data	Purpose Specifies the default memory placement policy for data. Tuning Refers to the data of the main executable (initialized data, BSS), heap, shared library and object modules loaded at run-time. Data placement can be set to first-touch (value of 1), round-robin across the system (value of 2) or automatic (value of 0), where the system decides the best placement for the memory.
memplace_mapped_file	Purpose Specifies the default memory placement policy for files that are mapped into the address space of a process (such as through shmat() and mmap()). Tuning Default placement of memory mapped files can be set to first-touch (value of 1) or round-robin across the system (value of 2) or automatic (value of 0), where the system decides the best placement for the memory.

Tunable	Description
memplace_shm_anonymous	Purpose Specifies the default memory placement policy for anonymous shared memory. Tuning Anonymous shared memory refers to working storage memory, created via shmget() or mmap(), that can be accessed only by the creating process or its descendants. This memory is not associated with a name (or key). Default placement of anonymous shared memory can be set to first-touch (value of 1) or round-robin across the system (value of 2) or automatic (value of 0), where the system decides the best placement for the memory.
memplace_shm_named	Purpose Specifies the default memory placement policy for named shared memory. Tuning Named shared memory refers to working storage memory, created via shmget() or shm_open(), which is associated with a name (or key) that allows more than one process to access it simultaneously. Default placement of named shared memory can be set to first-touch (value of 1) or round-robin across the system (value of 2) or automatic (value of 0), where the system decides the best placement for the memory.
memplace_stack	Purpose Specifies the default memory placement policies for the program stack. Tuning Stack placement can be set to first-touch (value of 1) or round-robin across the system (value of 2) or automatic (value of 0), where the system decides the best placement for the memory.
memplace_text	Purpose Specifies the default memory placement policy for the application text. Tuning This applies only to the text of the main executable and not to its dependencies. Text placement can be set to first-touch (value of 1) or round-robin across the system (value of 2) or automatic (value of 0), where the system decides the best placement for the memory.

Tunable	Description
memplace_unmapped_file	Purpose Specifies the default memory placement policy for unmapped file access, such as through read()/write(). Tuning Default placement of unmapped file access can be set to first-touch (value of 1) or round-robin across the system (value of 2) or automatic (value of 0), where the system decides the best placement for the memory.
minfree	Purpose Specifies the number of frames on the free list at which the VMM starts to steal pages to replenish the free list. Tuning Page replacement occurs when the number of free frames reaches <i>minfree</i>. If the processes are being delayed by page stealing, increase <i>minfree</i> to improve response time. The difference between <i>maxfree</i> and <i>minfree</i> should be of the order of <i>maxpgahead</i>, and no less than 8.
minperm%	Purpose Specifies the point below which the page-stealer will steal file or computational pages regardless of repaging rates. Tuning You can decrease this parameter if large number of file pages in memory is causing working storage pages to be replaced. On the other hand, if some files are known to be read repetitively, and I/O rates do not decrease with time from startup, <i>minperm</i> may be too low.
nokilluid	Purpose The user IDs lower than this value will be exempt from getting killed due to low page-space conditions. Tuning A value of 0 indicates off. Useful when system is out of paging space and the system administration processes are being killed. Either set this tunable to 1 in order to protect specific user ID processes from getting killed due to low page space or ensure there is sufficient paging space available.

Tunable	Description
npsassert	Purpose Asserts the LPAR when the system runs out of paging space. Tuning • A value of 0 for this tunable provides legacy behavior where processes are killed when free paging space reaches the <i>npskill</i> threshold. Vmo tunable <i>low_ps_handling</i> influences which process is killed. SIGDANGER signal is sent to processes when the free paging space reaches danger levels. • A value of 1 for this tunable asserts the LPAR when paging space is completely exhausted. Processes are not killed when free paging space reaches <i>npskill</i> levels. SIGDANGER signal mechanism is suppressed, and processes are not notified when free paging space reaches danger levels.
npskill	Purpose Specifies the number of free paging-space pages at which the operating system begins killing processes. Tuning The default value is the maximum of 64 and (number of paging space pages)/128. The <i>npskill</i> value must be greater than zero and less than the total number of paging space pages on the system.
npswarn	Purpose Specifies the number of free paging-space pages at which the operating system begins sending the SIGDANGER signal to processes. Tuning The default value is the maximum of 512 and (4*npskill). The value of <i>npswarn</i> must be greater than zero and less than the total number of paging space pages on the system. Increase the value if you experience processes being killed because of low paging space.
numpsblks	Purpose Total number of paging-space blocks. Tuning N/A

Tunable	Description
pinnable_frames	Purpose Number of pages available for pinning Tuning N/A
relalias_percentage	Purpose If <i>force_relalias_lite</i> is set to 0, then this specifies the factor used in the heuristic to decide whether to avoid locking the source mmapped segment. Tuning This is used when tearing down an mmapped region and is a scalability statement, where avoiding the lock may help system throughput, but, in some cases, at the cost of more compute time used. If the number of pages being unmapped is less than this value divided by 100 and multiplied by the total number of pages in memory in the source mmapped segment, then the source lock is avoided. A value of 0 for <i>relalias_percentage</i>, with <i>force_relalias_lite</i> also set to 0, will cause the source segment lock to always be taken. Effective values for <i>relalias_percentage</i> will vary by workload, however, a suggested value is 200.
scrub	Purpose Enables or Disables freeing of paging space disk blocks from pages in memory for Deferred Page Space Allocation Policy pages. Tuning A value of 0 disables scrubbing completely. A value of 1 enables scrubbing of in memory paging space disk blocks when the number of system free paging space blocks is below npsscrubmin, and continues until above npsscrubmax.
v_pinshm	Purpose If set to 1, will allow pinning of shared memory segments. Tuning A value of 0 indicates off. Change this value when the overhead is high and in pinning or unpinning of AIO buffers from shared memory segments. Useful only if the application also sets the SHM_PIN flag when doing a shmget() call and if doing async I/O from shared memory segments.

Tunable	Description
vmm_default_pspa	Purpose This tunable controls the default aggressiveness of page size promotion. The value is an abstract aggressiveness weighting which is treated by the operating system as the inverse of the page promotion threshold. Tuning A value of 0 for the <i>vmm_default_pspa</i> setting is equivalent to a page promotion threshold of 100%, that is, a memory range must have 100% real memory occupancy in order to be promoted. A value of 100 for the <i>vmm_default_pspa</i> setting is equivalent to a page promotion threshold of 0%, that is, a memory range should be promoted immediately on first reference to memory in the range. A value of -1 for the <i>vmm_default_pspa</i> setting is equivalent to a page promotion threshold of -1, that is, never do page promotion for a memory range. Page size promotion thresholds are only considered at segment creation time. Thus, changing <i>vmm_default_pspa</i> will only affect the page size promotion thresholds for segments created after the tunable is adjusted.
wlm_memlimit_nonpg	Purpose Selects whether non-pageable page sizes (16M, 16G) are included in the WLM <i>realmem</i> and <i>virtmem</i> counts. If 1 is selected, then non-pageable page sizes are included in the <i>realmem</i> and <i>virtmem</i> limits count. If 0 is selected, then only pageable page sizes (4K, 64K) are included in the <i>realmem</i> and <i>virtmem</i> counts. This value can only be changed when WLM Memory Accounting is off, or the change will fail. Tuning When this tunable is set to 0, WLM virtual and real memory limits will only apply to pageable pages consumed by a WLM class. Because heavy use of pageable pages is what causes paging on a system, a value of 0 provides more granular control over how much a WLM class pages when non-pageable pages are in use. This tunable should only be adjusted when WLM real or virtual memory limits are being used on a system configured with non-pageable pages.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To list the current and reboot value, range, unit, type, and dependencies of all tunable parameters managed by the **vmo** command, enter the following command:

```
vmo -L
```

2. To turn on and reserve 16 MB large pages on a POWER4 system, enter the following command:

```
vmo -o lgpg_regions=10 -o lgpg_size=16777216
```

This command will prompt **bosboot** to the user, and warn that a reboot is necessary before the change is effective.

3. To display help on the **nokilluid** tunable parameter, enter the following command:

```
vmo -h nokilluid
```

4. To turn on the **v_pinshm** tunable parameter after the next reboot, enter the following command:

```
vmo -r -o v_pinshm=1
```

5. To permanently reset all **vmo** tunable parameters to default values, enter the following command:

```
vmo -p -D
```

6. To list the reboot value for all virtual memory manager tuning parameters, enter the following command:

```
vmo -r -a
```

7. To list (spreadsheet format) the current and reboot value, range, unit, type and dependencies of all tunable parameters managed by the **vmo** command, enter the following command:

```
vmo -x
```

8. **>** To turn on the **msem_nlocks** tunable parameter for the next Live Update operation, enter the following command:

```
vmo -r -K -o msem_nlocks=1M
```

|<

vmstat Command

Purpose

Reports virtual memory statistics.

Syntax

```
vmstat [ -f ] [ -i ] [ -s ] [ -I [ -W ] ] [ -t ] [ -v ] [ -h ] [ -w ] [ -l ] [ -c ] [ -L level ] [ -@ wparname | ALL ] [ { -p | -P } physicalvolume | ALL ] ALL [ -S power ] [ physicalvolume ] [ interval ] [ count ]
```

Note: The *wparname* option and the **-i** flag must not be used together in the workload partitions.

Description

The **vmstat** command reports the statistics of the kernel threads, virtual memory, disks, hypervisor pages, traps, and processor activity. The reports that are generated by the **vmstat** command are used to balance the system load activity. These system-wide statistics are calculated as averages when values are

expressed as percentages across all processors, and are calculated as sums in other cases. The **vmstat** command might return inconsistent statistics if the statistics are not read atomically.

If you run the **vmstat** command without specifying any flags, the **vmstat** command report contains a summary of the virtual memory activity since system startup. If you specify the **-f** flag, the **vmstat** command reports the number of forks since system startup. The **physicalvolume** parameter specifies the name of the physical volume.

The **interval** parameter specifies the amount of time in seconds between each report. If you do not specify the **interval** parameter, the **vmstat** command generates a single report that contains statistics for the time since system startup and then exits. You can specify the **count** parameter only with the **interval** parameter. If you specify the **count** parameter, the **count** parameter value determines the number of reports that are generated and the number of seconds apart. If you specify the **interval** parameter without the **count** parameter, reports are continuously generated. Do not specify a value of zero to the **count** parameter.

The kernel maintains the statistics for kernel threads, paging, and interrupt activity, which the **vmstat** command can access by using the *perfstat* kernel extension. The disk input/output (I/O) statistics are maintained by the device drivers. For disks, the average transfer rate is determined by using the active time and number of times information is being transferred. The % active time is computed from the amount of time the drive is busy during the report.

The **vmstat** command reports the number of physical processors consumed (pc), and the percentage of entitlement consumed (ec), in Micro-Partitioning environments.

The report that is generated by the **vmstat** command contains a system configuration row and the column headings. If the **-@** flag is specified, the **vmstat** command report consists the system configuration and WPAR configuration.

System configuration row

The system configuration row has the following values:

lcpu

The **lcpu** field indicates the number of logical processors.

mem

The **mem** field indicates the amount of memory.

tmem

The **tmem** field indicates the true memory size of the Logical partition (LPAR).

Note: This flag is available only when **-c** option is provided and the Active Memory Expansion is enabled.

ent

The **ent** field indicates the entitled capacity. This information is displayed only when you run the partition on a shared processor.

drives

The **drives** field indicates the number of disks. This information is displayed only when the physical volume name is monitored.

Workload Partitions (WPARs)

The **WPARs** field indicates the number of active workload partitions. This information is displayed only when you specify the **-@** flag.

memlim

The **memlim** field indicates the limit (in megabytes) of the memory resource of the workload partition. This information is displayed only for the WPAR with enforced memory resource limit.

cpulim

The **cpulim** field indicates the limit of processor resource of the workload partition in processor units. This information is displayed only for the WPAR with enforced processor resource limit.

rset

The **rset** field indicates the type of the **rset** registry that is associated with a WPAR. The **rset** registry type can be of type regular or exclusive. This information is displayed only for the **WPARs** that are associated with a **rset** registry.

mmode

The **mmode** field indicates the memory mode. This metric is displayed automatically when Active Memory Sharing is enabled on a system. This metric is also displayed when you specify a **-c** option.

mpsz

The **mpsz** field indicates the size of the memory pool in gigabytes. This metric is displayed only in a shared-memory mode.

Column headings

The column headings of the report that is generated by the **vmstat** command has the following values:

WPAR

WPAR is the Workload partition name and provides information about the workload partitions. This information is displayed only when you specify the **-@** flag.

Notes:

1. The system WPAR name indicates the system-wide statistics and the global WPAR name indicates that the statistics belong to Global only.
2. When you start the **vmstat** command with the **-@ ALL** option and the WPAR specific information is not available for a metric, an en dash sign (-) is displayed instead of a value.
3. When you start the **vmstat** command with the **-@ wparname** or inside a WPAR, if the WPAR information is not available for a metric, then that metric is marked with the at sign (@), and the system-wide value is displayed for that metric.
4. If a metric is not supported, then a en dash sign (-) is displayed instead of a value.

kthr

Provides information about the kernel thread states.

r

Represents the average number of kernel threads that can run over the sampling interval. The kernel threads that can run over the sampling interval consist of the threads that are ready but still waiting to run, and the threads that are already running.

b

Represents the average number of kernel threads that are placed in the Virtual Memory Manager (VMM) wait queue (awaiting resource, awaiting I/O) over the sampling interval.

Memory

Provides information about the usage of virtual and real memory. The virtual pages are considered to be active if they are accessed. A virtual page is 4096 bytes.

avm

Indicates the number of active virtual pages.

fre

Indicates the size of the free list.

Note: A large portion of real memory is used as a cache for file system data. The size of the free list remains small.

Page

Provides information about the page faults and paging activity. This information is averaged over the interval and displayed in units per second.

re

Indicates the I/O list of the pager.

pi

Indicates the pages that are paged in from paging space.

po

Indicates the pages that paged out to paging space.

fr

Indicates the pages that are freed (page replacement).

sr

Indicates the pages that are scanned by a page-replacement algorithm.

cy

Indicates the clock cycles by page-replacement algorithm.

Faults

Provides information about average rate of trap and interrupt rate per second over the sampling interval.

in

Indicates the device interrupts.

sy

Indicates the system calls.

cs

Indicates the kernel thread context switches.

CPU

Provides information about breakdown of percentage usage of processor time.

us

Indicates the user time.

If the current physical processor consumption of the uncapped partitions exceeds the entitled capacity, the percentage becomes relative to the number of physical processors consumed.

sy

Indicates the system time.

If the current physical processor consumption of the uncapped partitions exceeds the entitled capacity, the percentage becomes relative to the number of physical processors consumed.

id

Indicates the processor idle time.

If the current physical processor consumption of the uncapped partitions exceeds the entitled capacity, the percentage becomes relative to the number of physical processors consumed.

wa

Indicates the processor idle time during which the system had an outstanding disk or NFS I/O request.

If the current physical processor consumption of the uncapped partitions exceeds the entitled capacity, the percentage becomes relative to the number of physical processors consumed.

pc

Indicates the number of physical processors that are used. This information is displayed only when you run a partition on a shared processor.

ec

Indicates the percentage of entitled capacity that is consumed. This information is displayed only when you run a partition on a shared processor. As the time base over which this data is computed can vary, the entitled capacity percentage can sometimes exceed 100% and the excess is noticeable only with small sampling intervals.

rc

Indicates the percentage of processor resource that is used. This information is displayed only for the WPARs with enforced processor resource limit.

Disk

Provides information about the number of transfers per second to the specified physical volumes that occurred in the sample interval. The **physicalvolume** parameter can be used to specify one to four names. Transfer statistics are given for each specified drive in the order specified. This count represents requests to the physical device. It does not imply an amount of data that is read or written. Several logical requests can be combined into one physical request. If the **physicalvolume** parameter is used, the physical volume names are printed at the beginning of the command execution.

If you specify the **-I** flag, an I/O oriented view is presented with the following column changes:

kthr

The p column is displayed in addition to the r and b columns.

p

Indicates the average number of threads waiting for I/O messages from the raw devices. The raw devices are the devices that are directly attached to the system.

If the **-W** flag is specified along with the **-I** flag, an additional w column is also displayed along with the r, b, and p columns.

w

Indicates the number of threads per second of time that are waiting for the file system direct I/O event to occur. The file system direct I/O events are of the following types:

- Asynchronous I/O (AIO)
- Buffer cache subsystem
- Concurrent I/O (CIO)
- File system direct I/O
- NFS subsystem
- A thread is waiting for an action from the virtual memory manager (VMM) waiting list.

page

A new fi and fo columns are displayed instead of the re and cy columns.

fi

Indicates the file page-ins per second.

fo

Indicates the file page-outs per second.

If you specify the **-c** flag, an Active Memory Expansion view is presented with the following column changes:

memory

The columns csz, cfr, and dxm are displayed in addition to columns avm and fre.

csz

Indicates the current compressed pool size, in 4 KB page units.

cfr

Indicates the free pages available in a compressed pool, in 4 KB page units.

dxm

Indicates a deficit in expanded Memory Size, in 4 KB page units.

page

New columns ci and co are displayed instead of the re and cy columns.

ci

Indicates the number of page-ins per second from a compressed pool.

co

Indicated the number of page-outs per second to a compressed pool.

If there is a change in system configuration that effects the output of the **vmstat** command while the **vmstat** command is running, the **vmstat** command prints a warning message about the configuration change. The **vmstat** command continues to display the output after printing the information about the updated system configuration and the header.

If you specify a **-l** flag, an additional large - page section is displayed with the following columns:

alp

Indicates the number of large pages that are in use.

flp

Indicates the number of large pages that are on the large page freelist.

If you specify the **-p** flag, additional lines of VMM statistics are displayed for the specified page sizes. If you specify **-I** and **-t** flags, the **-p** flag produces an additional line for the specified page size. This additional line contains the following VMM statistics relevant to the specified page size:

- avm
- fre
- re
- fi
- fo
- pi
- po
- ci
- co
- fr
- sr
- cy

Notes:

- The **-I** flag affects the display of the **re**, **fi**, **fo**, and **cy** options.
- The **-c** flag affects the display of the **re**, **ci**, **co**, and **cy** options.
- If there is no resource control, then the **avm** and **fre** options are system-wide. Therefore, if the **-@** option is set, both the **avm** and **fre** options are marked with the at sign (@).

These VMM statistics are preceded by a **psz** column and followed by an **siz** column.

psz

Indicates the page size (for example, 4 KB, 64 KB).

siz

Indicates the number of frames of the specified page size that exist on the system.

If you specify the **-s** flag, the **-p** flag produces a separate stanza of output that contains only the statistics relevant to the specified page size. This additional stanza is preceded by a page size header.

The **-P** option generates the following report for the specified page size:

pgsz

Indicates the page size (for example, 4 KBB, 64 KBB).

Memory

Indicates the memory statistics for the specified page sizes.

siz

Indicates the number of frames of the specified page size that exist on the system.

avm

Indicates the active virtual pages applicable to the specified page size.

fre

Indicates the size of the free list for the specified page size.

Page

Indicates the relevant page faults and paging activity for the specified page size. The page-related columns *re*, *pi*, *po*, *fr*, *sr*, *cy*, *fi*, *fo*, *ci*, and *co* are also applicable to this report.

Flags

Note: If you specify either **-f** or **-s** flag on the command line, the system accepts the **-f** or **-s** flag and ignores the other flags. If you specify both the **-f** and **-s** flags on the command line, the system accepts only the first flag and ignores the second flag.

Item	Description
-@ <i>wparname</i>	Reports the Virtual Memory activity of a workload partition: - The -@ ALL option indicates that the report pertains to the system and global environment, in addition to all the workload partitions in the system. Note: The values that are system-wide statistics are marked with dash sign (-) against the WPAR section. - The -@ wparname flag indicates that the activity is only for that workload partition. In a workload partition, if you specify the -@ flag, system-wide statistics and workload partition statistics are displayed. The system-wide statistics are marked with the at sign (@). Note: Do not use the -@ flag with any combination of the -i flag.
-c	Displays the memory compression statistics with the new columns in the output as, <i>csz</i>, <i>cfr</i>, and <i>dxm</i> under the heading <i>memory</i>, and columns <i>ci</i> and <i>co</i> under the heading <i>page</i> instead of the columns <i>re</i> and <i>cy</i>. Note: This option is available only when you enable the Active Memory Expansion.
-f	Reports the number of forks since system startup.
-i	Displays the number of interrupts that are taken by each device since system startup.
-I	Displays the I/O oriented view with the new columns of output, <i>p</i> under the heading <i>kthr</i>, and columns <i>fi</i> and <i>fo</i> under the heading <i>page</i> instead of the columns <i>re</i> and <i>cy</i> in the <i>page</i> heading. Note: The -I, -t, -w, and -l flags are ignored if these flags are specified with the -i flag.
-l	Displays an extra large -page section with the <i>alp</i> and <i>flp</i> columns.
-L <i>level</i>	Displays the soft page promotion statistics. Note: You can only use -v flag with the -L flag. Do not use other flags with -L flag.
-p <i>pagesize</i>	Appends the VMM statistics for the specified page size to the regular vmstat command output.
-P <i>pagesize</i>	Displays only the VMM statistics, which are relevant for the specified page size.

Item	Description
-s	Writes the contents of the sum structure to the standard output, which contains an absolute count of paging events since system initialization. You can specify the -s flag only with the -v flag. These events are described as follows: address translation faults Incremented for each occurrence of an address translation page fault. I/O might not be required to resolve the page fault. Storage protection page faults (lock misses) are not included in this count. backtracks Incremented for each page fault that occurs while resolving a previous page fault. Resolve the new page fault first and then initial page faults can be backtracked. CPU context switches Incremented for each processor context switch (dispatch of a new process). decrementer interrupts Incremented on each decrementer interrupt. device interrupts Incremented on each hardware interrupt. executable-filled page faults Incremented for each instruction page fault. extend XPT waits Incremented each time that a process is waited by VMM due to a commit in progress for the segments accessed. free frame waits Incremented each time that a process requests a page frame. The free list is empty, and the process is forced to wait while the free list is replenished. iodones Incremented at the completion of each VMM I/O request. mpc send interrupts Incremented each time a multipath channel (mpc) sends an interrupt. mpc receive interrupts Incremented each time that a mpc receives an interrupt. page-ins Incremented for each page read in by the virtual memory manager. The count is incremented for page-ins from page space and file space. Along with the page-out statistic, this value represents the total amount of real I/O initiated by the virtual memory manager. page-outs Incremented for each page that is written by the virtual memory manager. The count is incremented for page-outs to page space and for page-outs to file space. Along with the page-in statistic, this statistic represents the total amount of real I/O initiated by the virtual memory manager. paging space page-ins Incremented for VMM initiated page-ins from paging space only. paging space page-outs Incremented for VMM initiated page-outs to paging space only. pages examined by the clock VMM uses a clock-algorithm to implement a pseudo least recently used (LRU) page replacement scheme. Pages are aged by being examined by the clock. This count is incremented for each page examined by the clock. pages freed by the clock Incremented for each page the clock-algorithm selects to free from real memory. pending I/O waits Incremented each time that a process is waited by VMM for a page-in I/O to complete.

Item	Description
-s	phantom interrupts Incremented on each phantom interrupt. revolutions of the clock hand Incremented for each VMM clock revolution (that is after each complete scan of memory). start I/Os Incremented for each read or write I/O request that is initiated by VMM. syscalls Incremented for each system call. total reclaims Incremented when an address translation fault is satisfied without initiating a new I/O request. This event occurs in the following conditions: • If the page is previously requested by VMM, but the I/O is not yet complete. • If the page was pre-fetched by the read-ahead algorithm of VMM, but was hidden from the faulting segment • If the page was put on the free list and is not yet reused. traps Indicates that the event is not maintained by the operating system. zero-filled page faults Incremented if the page fault is to working storage and can be satisfied by assigning a frame and zero-filling it. If you specify the -c flag along with the -s flag, the following additional metrics are displayed: compressed pool page ins Indicted the number of page-ins from the Compressed Pool since system boot. compressed pool page outs Indicates the number of page-outs to the Compressed Pool since system boot.
-s	When used with the -p pagesize flag, the -s flag appends the sum structure for the specified page size to the system-wide sum structure. This additional stanza is preceded by a page size header (for example, 4 KB pages). The following details are not displayed in this page size-based stanza as these statistics are not related to page sizes: • Processor context switches • Device interrupts • Software interrupts • Decrementer interrupts • MPC-sent interrupts • MPC-received interrupts • Phantom interrupts • Traps • Syscalls Notes: 1. If you specify -s flag with the -@ ALL flag, the system-wide statistics are repeated in the workload partition section. 2. If you specify -s flag with the -@ wparname flag, all metrics are reported and the system-wide statistics are marked with the at sign (@). 3. If you specify -s flag with the -l flag, the vmstat command displays the following metric: large-page hi water count Specifies the maximum value of the large-page in use count.

Item	Description
-S <i>power</i>	Multiplies the statistics of the processor with a value of 10^{power}. The default value of the power is 0. The following statistics are scaled: • us • sy • id • wa • pc • ec Notes: 1. Do not specify the -S flag with the -f, -s, -i, -v, or -p flags. 2. When the -S flag is specified, the us, sy, id, and wa statistics change. By default, the us, sy, id, and wa statistics are relative to the processor consumption of WPAR. When the -S flag is specified with a value of <i>power</i> that is not equal to zero, these statistics are relative to system-wide processor consumption. 3. The value of <i>power</i> for -S flag is in the range 0-3.
-t	Prints the time-stamp next to each line of output of the vmstat command. The time-stamp is displayed in the HH:MM:SS format. Note: A timestamp is not printed if -f, -s, or -i flags are specified.

Item	Description
-v	Writes various statistics that are maintained by the Virtual Memory Manager to the standard output. The -v flag can be used only with the -s, -L, and -h flags. If you specify the -v flag, the following statistics are displayed: compressed percentage Indicates the percentage of memory that is used by compressed pages. client filesystem I/Os blocked with no fsbuf Indicates the number of client file system I/O requests blocked because no <code>fsbuf</code> was available. Network File System (NFS) and Veritas (VxFS) are client file systems. <code>Fsbuf</code> are pinned memory buffers that are used to hold I/O requests in the file system layer. client pages Indicates the number of client pages. compressed pages Indicates the number of compressed memory pages. external pager filesystem I/Os blocked with no fsbuf Indicates the number of external pager client file system I/O requests blocked because no <code>fsbuf</code> was available. JFS2 is an external pager client file system. <code>Fsbuf</code> are pinned memory buffers that are used to hold I/O requests in the file system layer. file pages Indicates the number of 4 KBB pages that are used by the file cache. free pages Indicates the number of free 4 KBB pages. filesystem I/Os blocked with no fsbuf Indicates the number of file system I/O requests blocked because no <code>fsbuf</code> was available. <code>Fsbuf</code> are pinned memory buffers that are used to hold I/O requests in the file system layer. lrutable pages Indicates the number of 4 KBB pages that are considered for replacement. This number excludes the pages that are used for VMM internal pages, and the pages that are used for the pinned part of the kernel text. maxclient percentage Indicates the tuning parameter (managed by using <code>vmo</code>) that specifies the maximum percentage of memory, which can be used for client pages. maxperm percentage Indicates the tuning parameter in the percentage of real memory that is managed by using <code>vmo</code>. maxpin percentage Indicates the tuning parameter (managed by using <code>vmo</code>) that specifies the percentage of real memory that can be pinned. memory pages Indicted the size of real memory in a number of 4 KBB pages. memory pools Indicates the tuning parameter that specifies the number of memory pools that are managed by using <code>vmo</code>. minperm percentage Indicates the tuning parameter in the percentage of real memory that is managed by using <code>vmo</code>. numclient percentage Indicates the percentage of memory that is occupied by client pages.

Item**-v** (continued)**Description****numperm percentage**

>| Indicates the percentage of 1ruable memory that is used by the file cache. |<

paging space I/Os blocked with no psbuf

Indicates the number of paging space I/O requests that are blocked because the psbuf space is not available. The psbufs space is pinned memory buffers that are used to hold I/O requests at the virtual memory manager layer.

pending disk I/Os blocked with no pbuf

Indicates the number of pending disk I/O requests that are blocked because no pbuf was available. The pbuf are pinned memory buffers that are used to hold I/O requests at the logical volume manager layer.

pinned pages

Indicates the number of pinned 4 KBB pages.

Note: When the kernel locking feature (**vmm_klock_mode** parameter) is enabled, the pinned pages include the kernel locking (klocked) pages. For more information about the kernel locking feature, enter the following command:

```
vmo -h vmm_klock_mode
```

remote page-outs scheduled

Indicates the number of page-outs that are scheduled for client file systems.

If you specify the **-h** flag with the **-v** flag, the following additional metrics are displayed:

Time resolving virtualized partition memory page faults

Indicates the total time that the virtual partition is blocked to wait for the resolution of its memory page fault. The time is measured in seconds, with millisecond granularity.

Virtualized partition memory page faults

Indicates the total number of virtual partition memory page faults that are recorded for the virtualized partition.

Number of 4 KB page frames loaned

Indicates the number of 4 KB pages of the memory of the partition that is loaned to the hypervisor.

Percentage of partition memory loaned

Indicates the percentage of the memory of the partition that is loaned to the hypervisor.

Notes:

1. When the **-v** flag is used with the **-@ ALL** flag, the system-wide statistics are not repeated in the workload partition section.
2. When the **-s** flag is used with the **-@ wparname** flag, all metrics are reported and the system-wide statistics are marked with the at sign (@).

If you specify the **-c** flag along with the **-v** flag, the following additional metrics are displayed:

Compressed Pool Size

Indicates the size of Compressed Pool, in 4 KB page unit.

percentage of true memory that is used for compressed pool

Indicates the percentage of unexpanded memory that is used for compressed pool.

free pages in compressed pool (4 KB pages)

Indicates the number of free pages in a compressed pool, in a 4 KB page unit.

target memory expansion factor

Indicates the target memory expansion factor that is configured for the LPAR.

achieved memory expansion factor

Indicates the Current memory expansion factor that is achieved.

Item	Description
-h	Displays the hypv -page section that includes the hypervisor page information. The hypv -page section contains the following metrics: hpi Indicates the number of hypervisor page-in per second. hpit Indicates the average time that is spent in milliseconds per hypervisor page-in. pmem Indicates the amount of physical memory that backs the logical memory of partitions. The value is measured in gigabytes. If you specify the -h flag with the -v flag, the following metrics are displayed in addition to the metrics that are displayed by using the -v flag: Time resolving virtualized partition memory page faults Indicates the total time that the virtual partition is blocked to wait for the resolution of its memory page fault. The time is measured in seconds, with millisecond granularity. Virtualized partition memory page faults Indicates the total number of virtual partition memory page faults that are recorded for the virtualized partition. Number of 4 KBB page frames loaned Indicates the number of the 4 KBB pages of the memory that is loaned to the hypervisor in the partition. Percentage of partition memory loaned Indicates that the percentage of the memory loaned to the hypervisor in the partition.
-w	Displays the report in wide mode.
-W	Displays an extra field w in the kthr section. This option is allowed only with -I flag.

Notes:

1. If Active Memory Expansion is enabled, the **vmstat** command reports memory statistics in the expanded view. However, if the environment variable AME_MEMVIEW is set to TRUE, the memory statistics represent the true view.
2. The AME_MEMVIEW environment variable has no impact on memory statistics reported by using the **-c** flag.

Examples

1. To display a summary of the statistics since boot, enter the following command:

```
vmstat
```

2. To display five summaries at an interval of 2 seconds, enter the following command:

```
vmstat 2 5
```

3. To display a summary of the statistics since boot that includes statistics for logical disks scdisk13 and scdisk14, enter the following command:

```
vmstat scdisk13 scdisk14
```

4. To display fork statistics, enter the following command:

```
vmstat -f
```

5. To display the count of various events, enter the following command:

```
vmstat -s
```

6. To display time-stamp next to each column of the output of the **vmstat** command, enter the following command:

```
vmstat -t
```

7. To display the I/O oriented view with an alternative set of columns, enter the following command:


```
vmstat -I
```

8. To display all the VMM statistics available, enter the following command:

```
vmstat -vs
```

9. To display the large-page section with the `alp` and `flp` columns at an interval of 8 seconds, enter the following command:

```
vmstat -l 8
```

10. To display the VMM statistics specific to a particular page size (in the example, 4 KB), enter the following command:

```
vmstat -p 4K
```

11. To display the VMM statistics for all page sizes that are supported on the system, enter the following command:

```
vmstat -p ALL
```

Or enter the following command:

```
vmstat -p all
```

12. To display only the VMM statistics for a particular page size (in this example, 4 KB), enter the following command:

```
vmstat -P 4K
```

13. To display only the per-page breakdown of VMM statistics for all supported page sizes, enter the following command:

```
vmstat -P ALL
```

or enter the following command:

```
vmstat -P all
```

14. To display a summary of the statistics for all the workload partitions after boot, enter the following command:

```
vmstat -@ ALL
```

15. To display all the VMM statistics available for all the workload partitions, enter the following command:

```
vmstat -vs -@ ALL
```

16. To display both WPAR and system-wide VMM statistics from a workload partition, enter the following command:

```
vmstat -@
```

17. To multiply the processor values with 10 and display the results, enter the following command:

```
vmstat -S 1
```

18. To display the statistics for the hypervisor page, enter the following command:

```
vmstat -h
```

19. To display the information about pages that are loaned to the hypervisor, enter the following command:

```
vmstat -vh
```

20. To display memory compression statistics (in an LPAR with Active Memory Expansion enabled), enter the following command:

```
vmstat -c
```

21. To display memory compression statistics specific to per-pagesize (in an LPAR with Active Memory Expansion enabled), enter the following command:

```
vmstat -c -P ALL
```

22. To append memory compression information to the statistics displayed by **-s** flag (in an LPAR with Active Memory Expansion enabled), enter the following command:

```
vmstat -s -c
```

23. To append memory compression information to the statistics displayed by **-v** flag (in an LPAR with Active Memory Expansion enabled), enter the following command:

```
vmstat -v -c
```

24. To display the soft page promotion statistics, enter the following command:

```
vmstat -L 1
```

Files

Item	Description
/usr/bin/vmstat	Contains the vmstat command.

vpdadd Command

Purpose

Adds entries to the product, lpp, history, and vendor databases.

Syntax

```
vpdadd { -c Component | -p Product | -f Feature } -v v.r.m.f [ -D Destdir ] [ -U Command ] [ -R Prereq ] [ -S Msg_Set ] [ -M Msg_Number ] [ -C Msg_Catalog ] [ -P Parent ] [ -I Description ]
```

Description

The **vpdadd** command is for use with or by installers that wish to be listed in Vital Product Database (VPD). The VPD consists of the product, lpp, and history databases. Entries to the inventory database must be added by the **sysck** command. A new vendor database is now included to track products that use destination directories and **non-installp** uninstallers.

The **vpdadd** command uses a tree structure of *Product* at the highest level, then *Feature*, and then *Component*.

The *Component* is the lowest installable unit, but in this hierarchy, a *Component* is not selectable for install or uninstall. Therefore, if an installer is using the **vpdadd** command to update the install database, they should look at their own tree representation and add entries based on their structure. If only adding one entry per install, then adding a *Product* type rather than *Component* type would allow that entry to be listed in the uninstall SMIT interfaces. All the entries are made in the VPD, but *Components* and *Features* are filtered out in the default **lslpp** listings (**-Lc**).

Flags

Item	Description
-C <i>Msg_Catalog</i>	Specifies the message catalog to search for a translated description of the <i>Component</i> . The default (English) description is specified with the -I flag. If the message catalog is not in the standard NLSPATH, then the full path name should be given.
-c <i>Component</i>	Specifies the <i>Component</i> name to add to the VPD. An entry is only added if it is unique. Uniqueness is described as having a different destination directory. If the same instance of a <i>Component</i> is already in the database, then no entry is added, and an error is returned. This allows a force install (that is, reinstall).
-D <i>Destdir</i>	Specifies the root (prefix) path that is added to all the files in a <i>Component</i> when being installed (and when being added to the inventory database by the sysck command). Files in a <i>Component</i> are listed with relative path names, so the root path is allowed to change. The default destination directory is /opt .
-f <i>Feature</i>	Specifies the <i>Feature</i> name to add to the VPD. An entry is only added if it is unique. Uniqueness is described as having a different VRMF or destination directory. If the same instance of a <i>Feature</i> is already in the database, then no entry is added, and an error is not returned. This allows for a force install (that is, reinstall).
-I <i>Description</i>	Specifies the default description of the <i>Component</i> , <i>Feature</i> or <i>Product</i> . The description must be specified in double quotation marks. Single quotation marks are allowed inside the description, and double quotation marks must be prepended with a \ .
-M <i>Msg_Number</i>	Specifies the message number for the description.
-P <i>Parent</i>	Specifies the parent software unit. A <i>Component</i> specifies either a <i>Feature</i> or a <i>Product</i> as its parent, depending on where it was in the tree.
-p <i>Product</i>	Specifies the <i>Product</i> name to add to the VPD. An entry is only added if it is unique. Uniqueness is described as having a different VRMF or destination directory. If the same instance of a <i>Product</i> is already in the database, then no entry is added, and an error is not returned. This allows a force install (that is, reinstall).
-R <i>Prereq</i>	Specifies a <i>Component</i> (fileset) that is a requisite of the installing <i>Component</i> . The argument must be specified in quotation marks. This flag can be used more than once to specify multiple prerequisites. Although these are treated as prerequisites at install time (by the installer), they are listed as corequisites in the <i>Product</i> database to avoid creating circular requisite chains.
-S <i>Msg_Set</i>	Specifies the message set (if more than one in the catalog).
-U <i>Command</i>	Specifies the <i>Command</i> to launch the uninstaller for this <i>Component</i> . This may be just a command path name, or it may include parameters if there is a global uninstaller. The geninstall command calls this uninstaller, and installp does not deinstall a fileset if this value is set in the VPD.
-v <i>v.r.m.f</i>	The VRMF of the <i>Component</i> , <i>Feature</i> or <i>Product</i> being added.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. The following example shows how the Registry service would call **vpdadd** to add a *Component* for the *Foo product*. This *Component* has two requisites, one that is specific to the operating system, and one that is listed as GUID.

```
vpdadd -c EPL2890198489F -v 1.2.3.0 -R "bos.rte.odm 4.3.3.0" -R "8KDE0KY90245686 1.1.0.0" \
-U /usr/opt/foo/uninstaller.class -p KID892KYLIE25 -I "Foo Database Component"
```

2. To add a new product devices.pci.cool.rte to the VPD, enter:

```
vpdadd -p devices.pci.cool.rte -v 5.1.0.0 -U /usr/sbin/udisetaup
```

Files

/usr/sbin/vpdadd

vpddel Command

Purpose

Removes entries from the product, lpp, history, and vendor databases.

Syntax

vpddel { **-c** *Component* | **-p** *Product* | **-f** *Feature* } **-v** *V.R.M.F* **-D** *Dest_dir*

Description

The **vpddel** command removes entries from the product, lpp, history, and vendor databases. The *vrmf* and destination directory must be specified so that the correct entries are removed.

Flags

Item	Description
-c <i>Component</i>	Removes the specified <i>Component</i> . The VRMF must also be included when removing a <i>Component</i> .
-D <i>Dest_dir</i>	Specifies the destination directory of the <i>Component</i> to remove. If a destination directory is not included, then the default /opt is used.
-f <i>Feature</i>	Specifies the <i>Feature</i> to remove from the vendor database.
-p <i>Product</i>	The <i>Product</i> to remove from the vendor database.
-v <i>V.R.M.F</i>	Specifies the version, release, modification and fix level of the <i>component</i> to delete from the VPD and vendor database.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Example

To remove the *Component* EPL2890198489F from the product, history, lpp, and vendor databases, type:

```
vpddel -c EPL2890198489F -v 1.2.3.0 -D /usr/lpp/Foo
```

Files

/usr/sbin/vpddel

vsdata1st Command

Purpose

vsdata1st – Displays virtual shared disk subsystem information.

Syntax

vsdata1st {-g | -n | -v | -c}

Description

Use this command to display one of several kinds of information to standard output.

You can use the System Management Interface Tool (SMIT) to run the **vsdata1st** command. To use SMIT, enter:

```
smit list_vsd
```

and select the option for the kind of virtual shared disk SDR information you want to see.

Flags

Only one of the following flags can be specified with each invocation of **vsdata1st**:

-g

Displays the following global volume group data:

global_group_name,
local_group_name,
primary_server_node,
secondary_server_node. (This is only enabled with the Recoverable virtual shared disk subsystem.)
eio_recovery
recovery
CVSD server_list

-n

Displays the following Node data:

node_number,
host_name,
adapter_name,
min_buddy_buffer_size,
max_buddy_buffer_size,
max_buddy_buffers.

-v

Displays the following definition data:

vsd_name,
logical_volume_name,
global_group_name,

minor_number.

-c

Displays the following cluster information:

node_number

cluster_name

Parameters

None.

Security

You must have root authority to run this command.

Exit Status

0

Indicates the successful completion of the command.

nonzero

Indicates that an error occurred.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **starttrpdomain** command. To bring a particular node online in an existing peer domain, use the **starttrpnode** command. For more information on creating and administering an RSCT peer domain, refer to the *RSCT: Administration Guide*.

Standard Output

Current RVSD subsystem run level.

Examples

1. To display global volume group date, enter:

```
vsdatalst -g
```

The system displays a message similar to the following:

VSD Global Volume Group Information							
Global Volume Group name		Local VG name	Server Node Numbers			recovery	server_list
			primary	backup	eio_recovery		
gpfs0vg		gpfs0vg	1	2	0	0	0
gpfs1vg		gpfs1vg	2	1	0	0	0
gpfs3vg		gpfs3vg	1	0	0	0	1:2
							CVSD

2. To display global volume group date, enter:

```
vsdatalst -n
```

The system displays a message similar to the following:

VSD Node Information						
node number	host_name	VSD adapter	IP packet size	Buddy Buffer		
				minimum size	maximum size	# maxbufs
1	host1	m10	61440	4096	262144	128
2	host2	m10	61440	4096	262144	128

3. To display global volume group date, enter:

```
vsdata1st -v
```

The system displays a message similar to the following:

VSD Table VSD name	logical volume	Global Volume Group	minor#	size_in_MB
gpfs0vsd	gpfs0lv	gpfs0gvg	3	4096
gpfs1vsd	gpfs1lv	gpfs1gvg	1	4096
gpfs3vsd	gpfs3lv	gpfs3gvg	4	4096

Location

/opt/rsct/vsd/bin/vsdata1st

vsdchgserver Command

Purpose

vsdchgserver – Switches the server function for one or more virtual shared disks from the node that is currently acting as the server node to the other.

Syntax

vsdchgserver

-g vsd_global_volume_group_name -p primary_node

[-b secondary_node] [-o EIO_recovery]

Description

The **vsdchgserver** command allows the serving function for a global volume group defined on a primary node to be taken over by the secondary node, or to be taken over by the primary node from the secondary node. This allows an application to continue to use virtual shared disks in situations where the cable or adapter between the physical disks and one of the attached nodes is not working.

The Recoverable virtual shared disk subsystem automatically updates the virtual shared disk devices if, and only if, the **vsdchgserver** command is used to flip the currently-defined primary node and secondary node in the global volume group specified in the **-g** flag.

Flags

-g

Specifies the Global Volume Group name for the volume group that represents all the virtual shared disks defined on a particular node.

-p

Specifies the primary server node number for the global volume group.

-b

Specifies the secondary node number for the global volume group. If the **-b** flag is not specified, the secondary node definition will be removed.

-o

Specified as 0, for no recovery on an EIO error, or 1, for recovery on an EIO error.

Parameters

None.

Security

You must have root authority to run this command.

Exit Status

0

Indicates the successful completion of the command.

nonzero

Indicates that an error occurred.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **startdomain** command. To bring a particular node online in an existing peer domain, use the **startnode** command. For more information on creating and administering an RSCT peer domain, refer to the *RSCT: Administration Guide*.

Standard Output

Current RVSD subsystem run level.

Examples

To change the primary server node for the global volume group node12vg to node 1 and the secondary node to node 2, with EIO recovery, enter:

```
vsdchgserver -g node12vg -p 1 -b 2 -o 1
```

Location

/opt/rsct/vsd/bin/vsdchgserver

vsdelnode Command

Purpose

Removes virtual shared disk information for a node or series of nodes.

Syntax

vsdelnode *node_number* ...

Description

This command is used to remove virtual shared disk data for a node or series of nodes.

The **vsdelnode** command makes the listed nodes no longer virtual shared disk nodes so that no virtual shared disks can be accessed from them. This command is unsuccessful for any nodes that are servers for any global volume groups.

You can use the System Management Interface Tool (SMIT) to run the **vsdelnode** command. To use SMIT, enter:

```
smit delete_vsd
```

and select the **Delete Virtual Shared Disk Node** Information option.

Flags

- g**
Specifies the Global Volume Group name for the volume group that represents all the virtual shared disks defined on a particular node.
- p**
Specifies the primary server node number for the global volume group.
- b**
Specifies the secondary node number for the global volume group. If the **-b** flag is not specified, the secondary node definition will be removed.
- o**
Specified as 0, for no recovery on an EIO error, or 1, for recovery on an EIO error.

Parameters

node_number

Specifies the node number of the node whose virtual shared disk information you want to remove.

Security

You must have `root` authority to run this command.

Restrictions

The recoverable virtual shared disk subsystem must be stopped on the node(s) you are deleting. Otherwise, the results may be unpredictable. For more information, see *RSCT for AIX 5L Managing Shared Disks*.

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **startprdomain** command. To bring a particular node online in an existing peer domain, use the **startprnode** command. For more information on creating and administering an RSCT peer domain, refer to *RSCT Administration Guide*.

Examples

To delete virtual shared disk node information for nodes **3** and **6**, enter:

```
vsdelnode 3 6
```

Location

/opt/rsct/vsd/bin/vsdelnode

vsdelvg Command

Purpose

vsdelvg – Removes virtual shared disk global volume group information.

Syntax

vsdelvg [-f] *global_group_name* ...

Description

Use this command to remove virtual shared disk global volume group information. If any virtual shared disks are defined on a global volume group, the **vsdelvg** command is unsuccessful unless **-f** is specified.

If **-f** is specified, any such virtual shared disks must be unconfigured and in the defined state on all the virtual shared disk nodes to be deleted.

You can use the System Management Interface Tool (SMIT) to run the **vsdelvg** command. To use SMIT, enter:

```
smit delete_vsd
```

and select the **Delete Virtual Shared Disk Global Volume Group Information** option.

Flags

-f

Forces the removal of any virtual shared disks defined on this global volume group.

Parameters

global_group_name

Specifies the volume group that you no longer want to be global to the system.

Security

You must have root authority to run this command.

Exit Status

0

Indicates the successful completion of the command.

nonzero

Indicates that an error occurred.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **starttrpdomain** command. To bring a particular node online in an existing peer domain, use the **starttrpnode** command. For more information on creating and administering an RSCT peer domain, refer to the *RSCT: Administration Guide*.

Standard Output

Current RVSD subsystem run level.

Examples

To remove the global volume group **vg1n1**, enter:

```
vsdelvg vg1n1
```

Location

/opt/rsct/vsd/bin/vsdelvg

vsdnode Command

Purpose

Define virtual shared disk information for a node or series of nodes.

Syntax

vsdnode

node_number... adapter_name min_buddy_buffer_size

max_buddy_buffer_size max_buddy_buffers

vsd_max_ip_msg_size [cluster_name]

Description

Use this command to make the specified nodes virtual shared disk nodes and to assign their virtual shared disk operational parameters. If this information is the same for all nodes, run this command once. If the information is different for the nodes, run this command once for each block of nodes that should have the same virtual shared disk information.

You can use the System Management Interface Tool (SMIT) to run the **vsdnode** command. To use SMIT, enter:

```
smit vsd_data
```

and select the **virtual shared disk Node Information** option.

Flags

-f

Forces the removal of any virtual shared disks defined on this global volume group.

Parameters

node_number

Specifies the node or nodes whose virtual shared disk information is to be set. The value you specify for *node_number* must match a valid RSCT remote peer domain node number.

adapter_name

Specifies the adapter name to be used for virtual shared disk communications for the nodes specified. The adapter name must already be defined to the nodes. Note that the nodes involved in virtual shared disk support must be fully connected so that proper communications can take place. Use **ml0** to specify that the virtual shared disk device driver transmits data requests over the SP Switch. The **ml0** adapter will be used the next time the virtual shared disk device driver is loaded.

min_buddy_buffer_size

Specifies the smallest buddy buffer a server uses to satisfy a remote request to a virtual shared disk. This value must be a power of 2 and greater than or equal to 4096. The suggested value is 4096 (4 KB). For a 512 byte request, 4 KB is excessive. However, recall that a buddy buffer is only used for the short period of time while a remote request is being processed at the server node.

max_buddy_buffer_size

Specifies the largest buddy buffer a server uses to satisfy a remote noncached request. This value must be a power of 2 and greater than or equal to the *min_buddy_buffer_size*. The suggested value is 262144 (256 KB). This value depends on the I/O request size of applications using the virtual shared disks and the network used by the virtual shared disk software.

max_buddy_buffers

Specifies the number of *max_buddy_buffer_size* buffers to allocate. The virtual shared disk device driver will have an initial size when first loaded, and then will dynamically allocate and reclaim additional space as needed. The suggested value is 2000 256 KB buffers.

Buddy buffers are only used on the servers. On client nodes you may want to set *max_buddy_buffers* to 1.

Note: The `statvsd` command will indicate if remote requests are queueing waiting for buddy buffers.

vsd_max_ip_msg_size

Specifies the maximum message size in bytes for virtual shared disks. This value must not be greater than the maximum transmission unit (MTU) size of the network. The recommended values are:

- 61440 (60KB) for a switch
- 8192 (8KB) for jumbo frame Ethernet
- 1024 (1KB) for 1500-byte MTU Ethernet

cluster_name

A cluster name must be specified for server nodes that will be serving concurrently accessed shared disks. The cluster name can be any user provided name. A node can only belong to one cluster. For example, when you have a concurrent access environment, the two servers for the CVSD must both specify the same cluster name.

Note: The *cluster_name* is required only for SSA (Serial Storage Architecture) disks.

Security

You must have `root` authority to run this command.

Restrictions

The node specified on this command must already belong to a peer domain, and you must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **starttrpdomain** command. To bring a particular node online in an existing peer domain, use the **starttrpnode** command. For more information on creating and administering an RSCT peer domain, refer to *RSCT Administration Guide*.

Examples

The following example defines information for a switch network and nodes 1 through 8.

```
vsdnode 1 2 3 4 5 6 7 8 m10 4096 262144 128 61440
```

Location

/opt/rsct/vsd/bin/vsdnode

vsdsklst Command

Purpose

Produces output that shows you the disk resources used by the virtual shared disk subsystem across a peer domain.

Syntax

```
vsdsklst [-v] [-d] [-a | -n node_number [, node_number2, ...]]
```

Description

Use this command to check disk utilization across a peer domain.

Flags**-v**

Displays only disk utilization information about volume groups and the virtual shared disks associated with them.

-d

Displays only disk utilization information about volume groups and the physical disks associated with them.

-a

Displays specified information for all nodes in the system or system partition.

-n *node_number*

Lists one or more node numbers for which information is to be displayed.

Parameters

node_number

Specifies the node or nodes whose virtual shared disk information is to be set. The value you specify for *node_number* must match a valid RSCT remote peer domain node number.

adapter_name

Specifies the adapter name to be used for virtual shared disk communications for the nodes specified. The adapter name must already be defined to the nodes. Note that the nodes involved in virtual shared disk support must be fully connected so that proper communications can take place. Use **m10** to specify that the virtual shared disk device driver transmits data requests over the SP Switch. The **m10** adapter will be used the next time the virtual shared disk device driver is loaded.

min_buddy_buffer_size

Specifies the smallest buddy buffer a server uses to satisfy a remote request to a virtual shared disk. This value must be a power of 2 and greater than or equal to 4096. The suggested value is 4096 (4 KB). For a 512 byte request, 4KB is excessive. However, recall that a buddy buffer is only used for the short period of time while a remote request is being processed at the server node.

max_buddy_buffer_size

Specifies the largest buddy buffer a server uses to satisfy a remote noncached request. This value must be a power of 2 and greater than or equal to the *min_buddy_buffer_size*. The suggested value is 262144 (256 KB). This value depends on the I/O request size of applications using the virtual shared disks and the network used by the virtual shared disk software.

max_buddy_buffers

Specifies the number of *max_buddy_buffer_size* buffers to allocate. The virtual shared disk device driver will have an initial size when first loaded, and then will dynamically allocate and reclaim additional space as needed. The suggested value is 2000 256KB buffers.

Buddy buffers are only used on the servers. On client nodes you may want to set *max_buddy_buffers* to 1.

Note: The `statvsd` command will indicate if remote requests are queueing waiting for buddy buffers.

vsd_max_ip_msg_size

Specifies the maximum message size in bytes for virtual shared disks. This value must not be greater than the maximum transmission unit (MTU) size of the network. The recommended values are:

- 61440 (60KB) for a switch
- 8192 (8KB) for jumbo frame Ethernet
- 1024 (1KB) for 1500-byte MTU Ethernet

cluster_name

A cluster name must be specified for server nodes that will be serving concurrently accessed shared disks. The cluster name can be any user provided name. A node can only belong to one cluster. For example, when you have a concurrent access environment, the two servers for the CVSD must both specify the same cluster name.

Note: The *cluster_name* is required only for SSA (Serial Storage Architecture) disks.

Security

You must have `root` authority to run this command.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **startpdomain** command. To bring a particular node online in an existing peer domain, use the **startpnode** command. For more information on creating and administering an RSCT peer domain, refer to *RSCT Administration Guide*.

Examples

This command:

```
vsdsklst -dv -a
```

displays the following information on a system that has volume groups and virtual shared disks defined on nodes 1 and 2.

```
c164n12.ppd.pok.ibm.com: Node Number:2; Node Name:c164n12.ppd.pok.ibm.com
c164n12.ppd.pok.ibm.com: Volume group:rootvg; Partition Size:32; Total:271; Free:168
c164n12.ppd.pok.ibm.com: Physical Disk:hdisk0; Total:271; Free:168
c164n12.ppd.pok.ibm.com: Volume group:testvg is not varied on.
c164n12.ppd.pok.ibm.com: Physical Disk:hdisk5;
c164n12.ppd.pok.ibm.com: Volume group:testlvvg; Partition Size:4; Total:537; Free:534
c164n12.ppd.pok.ibm.com: Physical Disk:hdisk2; Total:537; Free:534
c164n12.ppd.pok.ibm.com: VSD Name:vsd1n2[testnewlv21n2]; Size:1
c164n12.ppd.pok.ibm.com: VSD Name:vsd2n2[testlv1n2]; Size:346112.25
c164n12.ppd.pok.ibm.com: VSD Name:vsd3n2[testlv2n2]; Size:346112.25
c164n12.ppd.pok.ibm.com: Volume group:vg1 is not varied on.
c164n12.ppd.pok.ibm.com: Physical Disk:hdisk9;
c164n12.ppd.pok.ibm.com: Volume group:sharkvg is not varied on.
c164n12.ppd.pok.ibm.com: Physical Disk:hdisk7;
c164n12.ppd.pok.ibm.com: Physical Disk:hdisk10;
c164n12.ppd.pok.ibm.com: Volume group:bdhclvg; Partition Size:32; Total:134; Free:102
c164n12.ppd.pok.ibm.com: Physical Disk:hdisk13; Total:134; Free:102
c164n12.ppd.pok.ibm.com: Volume group:gpfs0vg; Partition Size:8; Total:536; Free:0
c164n12.ppd.pok.ibm.com: Physical Disk:hdisk12; Total:536; Free:0
c164n12.ppd.pok.ibm.com: VSD Name:gpfs0vsd[gpfs0lv]; Size:352256.75
c164n12.ppd.pok.ibm.com: Not allocated physical disks:
c164n12.ppd.pok.ibm.com: Physical disk:hdisk1
c164n12.ppd.pok.ibm.com: Physical disk:hdisk3
c164n12.ppd.pok.ibm.com: Physical disk:hdisk4
c164n12.ppd.pok.ibm.com: Physical disk:hdisk6
c164n12.ppd.pok.ibm.com: Physical disk:hdisk11
c164n12.ppd.pok.ibm.com: Physical disk:hdisk15
c164n11.ppd.pok.ibm.com: Node Number:1; Node Name:c164n11.ppd.pok.ibm.com
c164n11.ppd.pok.ibm.com: Volume group:rootvg; Partition Size:32; Total:271; Free:172
c164n11.ppd.pok.ibm.com: Physical Disk:hdisk0; Total:271; Free:172
c164n11.ppd.pok.ibm.com: Volume group:bdhclvg; Partition Size:32; Total:134; Free:102
c164n11.ppd.pok.ibm.com: Physical Disk:hdisk9; Total:134; Free:102
c164n11.ppd.pok.ibm.com: VSD Name:bdhcvsd1n1[lvbdhcvsd1n1]; Size:45056
c164n11.ppd.pok.ibm.com: Volume group:testvg; Partition Size:16; Total:134; Free:70
c164n11.ppd.pok.ibm.com: Physical Disk:hdisk13; Total:134; Free:70
c164n11.ppd.pok.ibm.com: Not allocated physical disks:
c164n11.ppd.pok.ibm.com: Physical disk:hdisk1
c164n11.ppd.pok.ibm.com: Physical disk:hdisk2
c164n11.ppd.pok.ibm.com: Physical disk:hdisk3
```

Location

/opt/rsct/vsd/bin/vsdsklst

vsvdvg Command

Purpose

Defines a virtual shared disk global volume group.

Syntax

vsdvg

```
[-g global_volume_group ] {-l server_list local_group_name / local_group_name primary_node  
[secondary_node [eio_recovery]]}
```

Description

Use this command to define volume groups for use by the Virtual shared disk subsystem. This is done by specifying the local volume group name, the node on which it resides, and the name by which the volume group will be known throughout the cluster.

You can use the System Management Interface Tool (SMIT) to run the **vsdvg** command. To use SMIT, enter the following command and select the **Virtual Shared Disk Global Volume Group Information** option:

```
smit vsd_data
```

Flags

-g *global_volume_group*

Specifies a unique name for the new global volume group. This name must be unique across the system partition. It should be unique across the SP, to avoid any naming conflicts during future system partitioning operations. The suggested naming convention is **vgxxn**yy, where yy is the node number, and xx uniquely numbers the volume groups on that node. If this is not specified, the local group name is used for the global name. The length of the name must be less than or equal to 31 characters.

-l *server_list*

Define the list of servers for CVSD. More than one server indicates the *global_volume_group* is a concurrent volume group.

Parameters

local_group_name

Specifies the name of a volume group that you want to indicate as being used for virtual shared disks. This name is local to the host upon which it resides. The length of the name must be less than or equal to 15 characters.

primary_node

Specifies the primary server node number on which the volume group resides. The length of the name must be less than or equal to 31 characters.

secondary_node

Specifies the secondary server node number on which the volume group resides. The length of the name must be less than or equal to 31 characters.

eio_recovery

Specifies how the Recoverable virtual shared disk subsystem will respond to EIO errors. If *eio_recovery* is set to the value 1 (the default), an EIO error will cause the Recoverable virtual shared disk system to flip the current primary node and the secondary node and perform one more retry on the new primary node.

Security

You must have root authority to run this command.

Exit Status

0

Indicates the successful completion of the command.

nonzero

Indicates that an error occurred.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **startpdomain** command. To bring a particular node online in an existing peer domain, use the **startpnode** command. For more information on creating and administering an RSCT peer domain, refer to the *RSCT: Administration Guide*.

Standard Output

Current RVSD subsystem run level.

Examples

1. The following example defines gpfs1vg as a virtual shared disk global volume group with the local volume group gpfs1vg accessed from node1 as the *primary_node* and node2 as the *secondary_node*.

```
vsdvg -g gpfs1vg gpfs1vg 1 2
```

2. The following example defines gpfs3vg as a virtual shared disk global volume group with the local volume group gpfs3vg concurrently accessed from node1 and node2.

```
vsdvg -g gpfs3vg -l 1:2 gpfs3vg
```

Location

/opt/rsct/vsd/bin/vsdvg

vsdvgts Command

Purpose

Updates the timestamp used by the Recoverable virtual shared disk subsystem by reading the timestamp from the volume group descriptor area (VGDA) of the physical disks.

Syntax

```
vsdvgt [-a] [ volgrp]
```

Description

Use this command to update the timestamp that the Recoverable virtual shared disk subsystem uses to determine if a twin-tailed volume group has changed. When the subsystem detects a change, the recovery scripts export the volume group and then import the volume group.

This command can be used to avoid exporting the volume group and then importing the volume group during recovery in situations where the export and import operations are not really necessary. This command should be used very carefully.

Flags**-a**

Specifies that the timestamps for this volume group for both primary and secondary nodes should be updated. If this flag is not specified, the timestamp is updated on the local node only.

Parameters

volgrp

Specifies a volume group. If this operand is not specified, the timestamps for all the volume groups on this node are updated.

Security

You must have root authority to run this command.

Exit Status

0

Indicates the successful completion of the command.

1

Indicates that the program was unable to read one or more timestamps.

Restrictions

You must issue this command from a node that is online in the peer domain. To bring a peer domain online, use the **startpdomain** command. To bring a particular node online in an existing peer domain, use the **startpnode** command. For more information on creating and administering an RSCT peer domain, refer to the *RSCT: Administration Guide*.

Standard Output

Current RVSD subsystem run level.

Examples

To update the timestamp associated with the virtual shared disk volume group vsdvg1 for just this node, enter:

```
vsdvgtts vsdvg1
```

Location

/usr/lpp/vsd/bin/vsdvgtts

W

The following AIX commands begin with the letter w.

w Command

Purpose

Prints a summary of current system activity.

Syntax

w [-h] [-u] [-w] [-l | -s [-X] [-@ [*WPAR*]] [*User*]

Description

The **w** command prints a summary of the current activity on the system. The summary includes the following:

Item	Description
WPAR	Workload partition name.
User	Who is logged on.
tty	Name of the tty the user is on.
login@	Time of day the user logged on.
idle	Number of minutes since a program last attempted to read from the terminal. Note: The idle time is taken from the global terminal when you log into wpar using the cllogin command.
JCPU	System unit time used by all processes and their children on that terminal.
PCPU	System unit time used by the currently active process.
What	Name and arguments of the current process.

The heading line of the summary shows the current time of day, how long the system has been up, the number of users logged into the system, and the load average. The load average is the number of runnable processes over the preceding 1-, 5-, 15-minute intervals.

The following examples show the different formats used for the login time field:

Item	Description
10:25am	The user logged in within the last 24 hours.
Tue10am	The user logged in between 24 hours and 7 days.
12Mar91	The user logged in more than 7 days ago.

If a user name is specified with the *User* parameter, the output is restricted to that user.

Flags

Item	Description

-@	Prints the System activity tagged with a workload partition name: • providing the -@ option without a WPAR name indicates the global environment in addition to all WPARs active in the system, and the heading line indicates values for the global environment only • providing the -@ option with a WPAR name indicates the activity, and the heading line indicates values for only that WPAR • providing -@ Global indicates the activity, and the heading line indicates values for the global environment only. Note: Not providing the -@ option indicates that the current WPAR or global environment, wherever the w command is running.
-h	Suppresses the heading.
-l	Prints the summary in long form. This is the default.
-s	Prints the summary in short form. In the short form, the tty is abbreviated, and the login time, system unit time, and command arguments are omitted.
-u	Prints the time of day, amount of time since last system startup, number of users logged on, and number of processes running. This is the default. Specifying the -u flag without specifying the -w or -h flag is equivalent to the uptime command.
-w	The equivalent of specifying the -u and -l flags, which is the default.
-X	Prints all available characters of each user name instead of truncating to the first 8 characters. The user name is also moved to the last column of the output.

Files

Item	Description
<u>/etc/utmp</u>	Contains the list of users.

wait Command

Purpose

Waits until the termination of a process ID.

Syntax

wait [*ProcessID* ...]

Description

The **wait** command waits (pauses execution) until the process ID specified by the *ProcessID* variable terminates. If the *ProcessID* variable is not specified, the **wait** command waits until all process IDs known to the invoking shell have terminated and exit with a 0 exit status. If a *ProcessID* variable represents an unknown process ID, the **wait** command treats them as known process IDs that exited with exit status 127. The **wait** command exits with the exitstatus of the last process ID specified by the *ProcessID* variable.

Flag

Item	Description
<i>ProcessID</i>	Specifies an unsigned decimal integer process ID of a command, which the wait command waits on until termination.

Exit Status

If one or more operands were specified, all of the operands terminated or were not known by the invoking shell, and the status of the last operand specified is known, then the exit status of the **wait** command is the same as the exit status information of the command indicated by the last operand specified. If the process terminated abnormally due to the receipt of a signal, then the exit status is greater than 128 and distinct from the exit status information generated by other signals, although the exact status value is unspecified (see the **kill -l** command option). Otherwise, the **wait** command exits with one of the following values:

Item	Description
0	The wait command was invoked with no operands and all process IDs known by the invoking shell have terminated.
1-126	The wait command detected an error.
127	The command identified by the last <i>ProcessID</i> operand specified is unknown.

File

Item	Description
<i>/usr/bin/wait</i>	Contains the wait command.

wall Command

Purpose

Writes a message to all users that are logged in.

Syntax

```
wall [ -a ] [ -g Group ][ Message ]
```

Description

The **wall** command writes a message to all users that are logged in. If the *Message* parameter is not specified, the **wall** command reads the message from standard input until it reaches an end-of-file character. The message is then sent to all logged in users. The following heading precedes the message:

```
Broadcast message from  
user@node  
  
(tty) at hh:mm:ss ...
```

hh:mm:ss represents the hours, minutes, and seconds when the message was sent.

To override any protections set up by other users, you must operate with root user authority. Typically, the root user uses the **wall** command to warn all other users of an impending system shutdown.

Note:

- The **wall** command only sends messages to the local node.

- Messages can contain multibyte characters.

Flags

Item	Description
-a	Performs the default operation. This flag is provided for System V compatibility. It broadcast messages to the console and pseudo-terminals.
-g <i>Group</i>	Broadcasts to a specified group only.

Files

Item	Description
/dev/tty	Specifies a device.

wallevent Command

Purpose

Broadcasts an event or a rearm event to all users who are logged in.

Syntax

wallevent [-c] [-h]

Description

The wallevent script broadcasts a message on an event or a rearm event to all users who are currently logged in to the host when the event or the rearm event occurs. Event or rearm event information is captured and posted by the event response resource manager in environment variables that are generated by the event response resource manager when an event or a rearm event occurs. This script can be used as an action that is run by an event response resource. It can also be used as a template to create other user-defined actions. The language in which the messages of the wallevent script are returned depend on the locale settings.

Messages are displayed in this format at the consoles of all users who are logged in when an event or a rearm event occurs for which this script is a response action :

```
Broadcast message from user@host (tty) at hh:mm:ss...

severity event_type occurred for Condition condition_name
on the resource resource_name of resource_class_name at hh:mm:ss mm/dd/yy
The resource was monitored on node_name and resided on {node_names}.
```

Event information is returned about the ERRM environment variables, and also includes the following:

Local Time

Time when the event or rearm event is observed. The actual environment variable supplied by ERRM is ERRM_TIME. This value is localized and converted to readable form before being displayed.

This script captures the environment variable values and uses the wall command to write a message to the currently logged-in user consoles.

Flags

- c**
Instructs wallevent to broadcast the ERRM_VALUE of an ERRM event. When the -c flag is specified, wallevent broadcasts the SNMP trap message.

-h

Writes the script's usage statement to standard output.

Parameters

log_file

Specifies the name of the file where event information is logged. An absolute path for the *log_file* parameter should be specified.

The *log_file* is treated as a circular log and has a fixed size of 64KB. When *log_file* is full, new entries are written over the oldest existing entries.

If *log_file* already exists, event information is appended to it. If *log_file* does not exist, it is created so that event information can be written to it.

Exit Status

0

Script has run successfully.

1

Error occurred when the script was run.

Restrictions

1. This script must be run on the node where the ERRM is running.
2. The `wall` command is used to write a message to currently logged-in user consoles. Refer to the `wall` man page for more information on the `wall` command.

Standard Output

When the `-h` flag is specified, the script's usage statement is written to standard output.

Examples

1. Suppose the `wallevent` script is a predefined action in the critical-notification response, which is associated with the `/var space used` condition on the resource `/var`. The threshold of the event expression defined for this condition is met, and an event occurs. The critical-notification response takes place, and `wallevent` is run. The following message is displayed on the consoles of all users who are logged in:

```
Broadcast message from joe@neverland.com (pts/6) at 18:42:03...

Critical event occurred for Condition /var space used
on the resource /var of filesys of IBM.FileSystem at 18:41:50 03/28/02
The resource was monitored on c174n05 and resided on {c174n05}.
```

2. When a rearm event occurs for the `/var space used` condition on the resource `/var`, the following message is displayed on the consoles of all users who are logged in:

```
Broadcast message from joe@neverland.com (pts/6) at 18:42:03...

Critical rearm event occurred for Condition /var space used
on the resource /var of filesys of IBM.FileSystem at 18:41:50 03/28/02
The resource was monitored on c174n05 and resided on {c174n05}.
```

Location

/opt/rsct/bin/wallevent

watch Command

Purpose

Observes a program that might be untrustworthy.

Syntax

watch [**-e** *Events*] [**-o** *File*] [**-X**] *Command* [*Parameter ...*]

Description

The **watch** command allows the root user or a member of the audit group to observe the actions of a program that are thought to be untrustworthy. The **watch** command starts the program you specify with the *Command* parameter, with or without any *Parameter* fields, and records all audit events or the audit events you specify with the **-e** flag.

The **watch** command observes all the processes that are created while the program runs, including any child process. The **watch** command continues until all processes exit, including the process it created, to observe all the events that occur.

The **watch** command formats the audit records and writes them to standard output or to a file you specify with the **-o** flag.

For the **watch** command to work, the auditing subsystem is not configured and enabled.

Flags

Item	Description
-e <i>Events</i>	Specifies the events to be audited. The <i>Events</i> parameter is a comma-separated list of audit events that are defined in the /etc/security/audit/events file. The default value is all events.
-o <i>File</i>	Specifies the path name of the output file. If the -o flag is not used, output is written to standard output.
-X	Prints long user names when used with other flags that display user names. The upper limit is determined by the <code>max_logname</code> object data manager (ODM) attribute in the predefined attribute (PdAt) and customized attributes (CuAt) object classes. If a user name is greater than the <code>max_logname</code> attribute, it is truncated to the number of characters as specified by the <code>max_logname</code> attribute, minus 1 character.

Security

Access Control: This command grants execute (x) access to the root user and members of the audit group. The **setuid** command is set for the root user. This setting allows access to other audit subsystem commands and files, and to the **trusted computing base** attribute.

Files Accessed:

Mode	File
r	/dev/audit
x	/usr/sbin/auditstream
x	/usr/sbin/auditselect
x	/usr/sbin/auditpr

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To watch all files opened by the **bar** command, enter:

```
watch -e FILE_Open /usr/lpp/foo/bar -x
```

This command opens the audit device and issues the **/usr/lpp/foo/bar** command. It then reads all records and selects and formats the files with the event type of FILE_Open.

2. To watch the installation of the xyzproduct program, that might be untrustworthy, enter:

```
watch /usr/sbin/installp xyzproduct
```

This command opens the audit device and issues the **/usr/sbin/installp** command. It then reads all records and formats them.

Files

Item	Description
/usr/sbin/watch	Contains the watch command.
/dev/audit	Specifies the audit device from which the audit records are read.

wc Command

Purpose

Counts the number of lines, words, bytes, or characters in a file.

Syntax

```
wc [ -c | -m ] [ -l ] [ -w ] [ File ... ]
```

```
wc -k [ -c ] [ -l ] [ -w ] [ File ... ]
```

Description

By default, the **wc** command counts the number of lines, words, and bytes in the files specified by the *File* parameter. The command writes the number of newline characters, words, and bytes to the standard output and keeps a total count for all named files.

When you use the *File* parameter, the **wc** command displays the file names as well as the requested counts. If you do not specify a file name for the *File* parameter, the **wc** command uses standard input.

The **wc** command is affected by the **LANG**, **LC_ALL**, **LC_CTYPE**, and **LC_MESSAGES** environment variables.

The **wc** command considers a word to be a string of characters of non-zero length which are delimited by a white space (for example SPACE , TAB).

Flags

Item	Description
-c	Counts bytes unless the -k flag is specified. If the -k flag is specified, the wc command counts characters.
-k	Counts characters. Specifying the -k flag is equivalent to specifying the -klwc flag. If you use the -k flag with other flags, then you must include the -c flag. Otherwise, the -k flag is ignored. For more information, see examples 4 and 5. Note: This flag is to be withdrawn in a future release.
-l	Counts lines.
-m	Counts characters. This flag cannot be used with the -c flag.
-w	Counts words. A word is defined as a string of characters delimited by spaces, tabs, or newline characters.

Note: If no flag is specified, **wc** by default counts the lines, words, bytes in a file or from standard input.

Exit Status

This command returns the following exit values:

Item	Description
0	The command ran successfully.
>0	An error occurred.

Examples

1. To display the line, word, and byte counts of a file, enter:

```
wc chap1
```

The **wc** command displays the number of lines, words, and bytes in the chap1 file.

2. To display only byte and word counts, enter:

```
wc -cw chap*
```

The **wc** command displays the number of bytes and words in each file that begins with chap. The command also displays the total number of bytes and words in these files.

3. To display the line, word, and character counts of a file, enter:

```
wc -k chap1
```

The **wc** command displays the number of lines, words, and characters in the chap1 file.

4. To display the word and character counts of a file, enter:

```
wc -kcw chap1
```

The **wc** command displays the number of characters and words in the chap1 file.

5. To use the **wc** command on standard input, enter:

```
wc -klw
```

The **wc** command displays the number of lines and words in standard input. The **-k** flag is ignored.

6. To display the character counts of a file, enter:

```
wc -m chap1
```

The **wc** command displays the number of characters in the chap1 file.

7. To use the **wc** command on standard input, enter:

```
wc -m1w
```

The **wc** command displays the number of lines, words, and characters in standard input.

Files

Item	Description
<code>/usr/bin/wc</code> , <code>/bin/wc</code>	Contains the wc command.
<code>/usr/ucb/wc</code>	Contains the symbolic link to the wc command.

what Command

Purpose

Displays identifying information in files.

Syntax

what [**-s**] *Pathname/File*.

Description

The **what** command searches specified files for all occurrences of the pattern that the **get** command substitutes for the **@(#)** keyletter (see the **get** or **prs** command for a description of identification keywords). By convention, the value substituted is "**@(#)**" (double quotation marks, at sign, left parenthesis, pound sign, right parenthesis, double quotation marks). If no file is specified, the **what** command reads from standard input.

The **what** command writes to standard output whatever follows the pattern, up to but not including the first double quotation mark (**"**), greater than symbol (**>**), new-line character, backslash (****), or null character.

The **what** command should be used in conjunction with the **get** command, which automatically inserts the identifying information. You can also use the **what** command on files where the information is inserted manually.

The **what** command accommodates the compiler inserted command line options in a binary file. The command line options saved in a binary file by AIX compilers may contain the backslash (****), the greater than symbol (**>**), or the double quotation mark (**"**), within the macro definitions. The **what** command behaves in the following manner to write the command line options saved by the compiler.

At the start of a line, if the pattern **@(#)** is followed by "opt" and is with or without a blank space (**" "**) in between, then the **what** command writes the character till **"\n"** that is the end of a line.

For example,

- For C and FORTRAN AIX compilers, use **@(#) opt (...)**.
- For C++ AIX compilers, use **@(#) opt (...)**.

The whole line is printed after "**@(#)**".

Note: The **what** command may fail to find SCCS identification strings in executable files.

Flags

Item	Description

-s	Searches for only the first occurrence of the @(#) pattern.

Exit Status

This command returns the following exit values:

Item	Description

0	Any matches were found.
1	Otherwise.

Examples

Suppose that the file `test.c` contains a C program that includes the line:

```
char ident[ ] = "@(#)Test Program";
```

If you compile `test.c` to produce `test.o`, then the command:

```
what test.c test.o
```

displays:

```
test.c:
Test Program
test.o:
Test Program
```

Note: The full file path names `usr/bin/test.c` and `usr/bin/test.o` are required if the files are not in the current directory.

Files

Item	Description
/usr/bin/what	Contains the what command.

whatis Command

Purpose

Describes what function a command performs.

Syntax

whatis [**-M** *PathName*] *Command* ...

Description

The **whatis** command looks up a given command, system call, library function, or special file name, as specified by the *Command* parameter, from a database you create using the **catman -w** command. The **whatis** command displays the header line from the manual section. You can then issue the **man** command to obtain additional information.

The **whatis** command is equivalent to using the **man -f** command.

Note: When the **/usr/share/man/whatis** database is built from the HTML library using the **catman -w** command, section 3 is equivalent to section 2 or 3. See the **man** command for further explanation of sections.

Flags

Item	Description
-M <i>PathName</i>	Specifies an alternative search path. The search path is specified by the <i>PathName</i> parameter, and is a colon-separated list of directories in which the whatis command expects to find the standard manual subdirectories.

Examples

To find out what the **ls** command does, enter:

```
whatis ls
```

This produces the following output:

```
ls(1)  -Displays the contents of a directory.
```

Files

Item	Description
/usr/share/man/whatis	Contains the whatis database.

whatnow Command

Purpose

Starts a prompting interface for draft disposition.

Syntax

```
whatnow [{ -draftfolder +folder | -nodraftfolder | file } { -draftmessage message | file } ]  
[ -editor editor | -noedit ] [ -prompt string ]
```

Description

The **whatnow** command provides an interface for the disposition of messages. By default, the interface operates on the current draft message. When you enter the **whatnow** command, the system places you in the interface and returns the following prompt:

```
What now?
```

Within the interface you can manipulate message drafts using the **whatnow** subcommands. To see a listing of the subcommands, press the Enter key at the What now? prompt. To exit the interface, press q.

If you do not specify the **-draftfolder** flag or if the Draft-Folder: entry in the **\$HOME/.mh_profile** file is undefined, the **whatnow** command searches your MH directory for a **draft** file. Specifying a message after the **-draftfolder** +*folder* flag is the same as specifying the **-draftmessage** flag.

To change the default editor for the **whatnow** command, use the **-editor** flag or define the Editor: entry in the *UserMhDirectory/.mh_profile* file.

Note: The **comp**, **dist**, **forw**, or **repl** commands use the same interface as the **whatnow** command.

Flags

Item	Description
-draftfolder <i>+folder</i>	Specifies the folder containing the message. By default, the system uses the <i>UserMhDirectory/draft</i> file. Specifying a message after the -draftfolder +folder is the same as using the -draftmessage flag.
-draftmessage <i>message</i>	Specifies the draft message.
-editor <i>editor</i>	Specifies that the value of the <i>editor</i> variable is the initial editor for composing or revising the message.
-help	Lists the command syntax, available switches (toggles), and version information.
	Note: For MH, the name of this flag must be fully spelled out.
<i>file</i>	User selected draft file.
<i>message</i>	Specifies the message. Use the following references to specify messages:
	Number Number of the message.
	cur or . (period) Current message. This is the default.
	first First message in a folder.
	last Last message in a folder.
	next Message following the current message.
	prev Message preceding the current message.
	-nodraftfolder Places the draft in the <i>UserMhDirectory/draft</i> file.
	-noedit Suppresses the initial edit.
	-prompt <i>string</i> Uses the specified string as the prompt. The default string is What now?.

whatnow Subcommands

The **whatnow** subcommands enable you to edit the message, direct the disposition of the message, or end the processing of the **whatnow** command.

Item	Description
display [<i>flags</i>]	Displays the message being redistributed or replied to. You can specify any <i>flags</i> parameter that is valid for the listing program. (Use the <code>lproc :</code> entry in the \$HOME/.mh_profile file to set a default listing program.) If you specify flags that are invalid for the listing program, the whatnow command does not pass the path name of the draft.
edit [<i>commandstring</i>]	Specifies with the <i>commandstring</i> parameter an editor for the message. You can specify the editor and any valid flags to that editor. If you do not specify an editor, the whatnow command uses the editor specified by the <code>Editor :</code> entry in your <i>UserMhDirectory/.mh_profile</i> file. If your <code>Editor :</code> entry is undefined, the whatnow command starts the editor used in the previous editing session.
list [<i>flags</i>]	Displays the draft. You can specify any <i>flags</i> parameter that is valid for the listing program. (To specify a default listing program, set a default <code>lproc :</code> entry in the \$HOME/.mh_profile file.) If you specify any flags that are invalid for the listing program, the whatnow command does not pass the path name of the draft.
push [<i>flags</i>]	Sends the message in the background. You can specify any valid flag for the send command.
quit [-delete]	Ends the whatnow session. If you specify the -delete flag, the whatnow command deletes the draft. Otherwise, the whatnow command stores the draft.
refile [<i>flags</i>] + <i>folder</i>	Files the draft in the specified folder and supplies a new draft having the previously specified form. You can specify any <i>flags</i> parameter that is valid for the command serving as the fileproc . (You can set a default <code>fileproc :</code> entry in the \$HOME/.mh_profile file.)
send [<i>flags</i>]	Sends the message. You can specify any valid flags for the send command.
whom [<i>flags</i>]	Displays the addresses to which the message would be sent. You can specify any valid flags for the whom command.

Profile Entries

The following entries are entered in the *UserMhDirectory/.mh_profile* file:

Item	Description
Draft-Folder:	Sets the default folder for drafts.
Editor:	Sets the default editor.
fileproc:	Specifies the program used to refile messages.
LastEditor-next:	Specifies the editor used after exiting the editor specified by the <i>LastEditor</i> variable.
lproc:	Specifies the program used to list the contents of a message.
Path:	Specifies the <i>UserMhDirectory</i> .
sendproc:	Specifies the program used to send messages.

Item	Description
whomproc :	Specifies the program used to determine the users to whom a message would be sent.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To display the original message when you are replying to a message, enter the following at the What now? prompt:

display

The system displays the original message. If you enter the **display** subcommand from a command other than the **dist** or **repl** command, you will receive a system message stating that there is no alternate message to display.

2. To edit the draft message with the vi editor, enter the following at the What now? prompt:

edit vi

3. To edit the draft message with the default editor specified in your **.mh_profile** file, enter the following at the What now? prompt:

edit

4. To list the contents of the draft message you have composed, enter the following at the What now? prompt:

list

The draft message you are composing is displayed.

5. To send the draft message in the background and get a shell prompt immediately, enter the following at the What now? prompt:

push

The draft message is sent and you immediately receive the shell prompt.

6. To quit composing a draft message and save it to a file so that you can later finish composing the message, enter the following at the What now? prompt:

quit

The system responds with a message similar to the following.

```
whatnow: draft left on /home/dale/Mail/draft
```

In this example, user dale's draft message is saved to the **/home/dale/Mail/draft** file.

7. To quit composing a draft message and delete the message, enter the following at the What now? prompt:

quit -delete

The shell prompt is displayed when the draft message is deleted.

8. To file the draft message you are composing before you send it, enter the following at the What now? prompt:

refile +tmp

The system responds with a message similar to the following:

```
Create folder "home/dale/Mail/tmp"?
```

In this example, if you answer yes, the draft message is filed in user dale's folder tmp.

9. To send the draft message you have composed, enter the following at the What now? prompt:

send

The shell prompt is displayed when the message is sent.

10. To verify that all addresses in the draft message are recognized by the mail delivery system, enter the following at the What now? prompt:

whom

The system responds with a message similar to the following:

```
jeanne... User unknown  
dale@venus... deliverable
```

In this example, the mail delivery system recognized dale@venus as a correct address, but did not recognize jeanne as a correct address.

Files

Item	Description
\$HOME/.mh_profile	Specifies the MH user profile.
<i>UserMhDirectory/draft</i>	Contains the current message draft.
/usr/bin/whatnow	Contains the whatnow command.

whereis Command

Purpose

Locates source, binary, or manual for program.

Syntax

whereis [-s] [-b] [-m] [-u] [{ -S | -B | -M } *Directory ...*]... -f] *File ...*

Description

The **whereis** command locates the source, binary, and manuals sections for specified files. The supplied names are first stripped of leading path name components and any (single) trailing extension of the form *.ext* (for example, *.c*). Prefixes of **s.** resulting from use of the Source Code Control System (see **SCCS**) are also dealt with. The command then attempts to find the desired program from a list of standard locations.

A usage message is returned if a bad option is entered. In other cases, no diagnostics are provided.

Flags

If any of the **-b**, **-s**, **-m** or **-u** flags are given, the **whereis** command searches only for binary, source, manual, or unusual sections respectively (or any two thereof).

Item	Description

-b	Searches for binary sections of a file.
-m	Searches for manual sections of a file.
-s	Searches for source sections of a file.
-u	Searches for unusual files. A file is said to be unusual if it does not have one entry of each requested type. Entering <code>whereis -m -u *</code> asks for those files in the current directory which have no documentation.

The **-B**, **-M**, and **-S** flags can be used to change or otherwise limit the places where the **whereis** command searches. Since the program uses the **chdir** subroutine to run faster, path names given with the **-M**, **-S** and **-B** flag directory list must be full; for example, they must begin with a **/** (slash).

Item	Description

-B	Like -b , but adds a directory to search. Change or limit the places where the whereis command searches for binaries.
-M	Like -m , but adds a directory to search. Change or limit the places where the whereis command searches for manual sections.
-S	Like -s , but adds a directory to search. Change or limit the places where the whereis command searches for sources
-f	Terminates the last -M , -S or -B directory list and signal the start of file names.

Examples

To find all of the files in the **/usr/ucb** directory that either are not documented in the **/usr/man/man1** directory or do not have source in the **/usr/src/cmd** directory, enter:

```
cd /usr/ucb
whereis -u -M /usr/man/man1 -S /usr/src/cmd -f *
```

Files

Item	Description

/usr/share/man/*

Directories containing manual files.

/sbin, /etc, /usr/{lib,bin,ucb,lpp}

Directories containing binary files.

/usr/src/*

Directories containing source code files.

which Command

Purpose

Locates a program file, including aliases and paths.

Syntax

which [*Name ...*]

Description

The **which** command takes a list of program names and looks for the files that run when these names are given as commands. The **which** command expands each argument, if it is aliased, and searches for it along the user's path. The aliases and paths are taken from the **.cshrc** file in the user's home directory. If the **.cshrc** file does not exist, or if the path is not defined in the **.cshrc** file, the **which** command uses the path defined in the user's environment.

A diagnostic is given if a name is aliased to more than a single word or if an executable file with the argument name is not found in the path.

In the Korn shell, you can use the **whence** command to produce a more verbose report.

Examples

To find the executable file associated with a command name of `lookup`:

```
which lookup
```

Files

Item	Description
\$HOME/.cshrc	Contains the source of aliases and path values.

which_fileset Command

Purpose

Searches the **/usr/lpp/bos/AIX_file_list** file for a specified file name or command.

Syntax

which_fileset [*File*]

Description

The **which_fileset** command searches the **/usr/lpp/bos/AIX_file_list** file for a specified file name or command name, and prints out the name of the fileset that the file or command is shipped in.

The **/usr/lpp/bos/AIX_file_list** file is large and not installed automatically. You must install the **bos.content_list** fileset to receive this file.

The *File* parameter can be the command name, the full path name, or a regular expression search pattern.

Examples

1. To display which fileset the `dbx` command is shipped in, enter:

```
which_fileset dbx
```

The screen displays the following:

```
/usr/bin/dbx > /usr/ccs/bin/dbx      bos.adt.debug 4.2.1.0
/usr/ccs/bin/dbx                    bos.adt.debug 4.2.1.0
```

2. To display all commands and paths containing the *sendmail* string, enter:

```
which_fileset sendmail.*
```

The screen displays the following:

```

/usr/ucb/mailq > /usr/sbin/sendmail      bos.compat.links 4.2.0.0
/usr/ucb/newaliases > /usr/sbin/sendmail bos.compat.links 4.2.0.0
/usr/lib/nls/msg/Ca_ES/sendmail87.cat    bos.msg.Ca_Es.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/ca_ES/sendmail87.cat     bos.msg.ca_Es.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/cs_CZ/sendmail87.cat     bos.msg.cs_CZ.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/De_DE/sendmail87.cat     bos.msg.De_DE.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/de_DE/sendmail87.cat     bos.msg.de_DE.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/En_US/sendmail87.cat     bos.msg.En_US.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/en_US/sendmail87.cat     bos.msg.en_US.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/Es_ES/sendmail87.cat     bos.msg.Es_ES.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/es_ES/sendmail87.cat     bos.msg.es_ES.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/Fr_FR/sendmail87.cat     bos.msg.Fr_FR.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/fr_FR/sendmail87.cat     bos.msg.fr_FR.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/hu_HU/sendmail87.cat     bos.msg.hu_HU.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/It_IT/sendmail87.cat     bos.msg.It_IT.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/it_IT/sendmail87.cat     bos.msg.it_IT.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/Ja_JP/sendmail87.cat     bos.msg.Ja_JP.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/ja_JP/sendmail87.cat     bos.msg.ja_JP.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/ko_KR/sendmail87.cat     bos.msg.ko_KR.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/pl_PL/sendmail87.cat     bos.msg.pl_PL.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/ru_RU/sendmail87.cat     bos.msg.ru_RU.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/Sv_SE/sendmail87.cat     bos.msg.Sv_SE.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/sv_SE/sendmail87.cat     bos.msg.sv_SE.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/ZH_CN/sendmail87.cat     bos.msg.ZH_CN.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/zh_CN/sendmail87.cat     bos.msg.zh_CN.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/Zh_TW/sendmail87.cat     bos.msg.Zh_TW.net.tcp.client 4.2.0.0
/usr/lib/nls/msg/zh_TW/sendmail87.cat     bos.msg.zh_TW.net.tcp.client 4.2.0.0
/etc/sendmail.cf                          bos.net.tcp.client.4.2.1.0
/usr/lib/sendmail > /usr/sbin/sendmail    bos.net.tcp.client.4.2.1.0
/usr/sbin/mailq > /usr/sbin/sendmail      bos.net.tcp.client.4.2.1.0
/usr/sbin/newaliases > /usr/sbin/sendmail bos.net.tcp.client.4.2.1.0
/usr/sbin/sendmail                         bos.net.tcp.client.4.2.1.0

```

3. To find where the **/usr/sbin/which_fileset** command is shipped, enter:

```
which_fileset /usr/bin/which_fileset
```

The screen displays:

```
/usr/sbin/which_fileset      bos.rte.install 4.2.1.0
```

who Command

Purpose

Identifies the users currently logged in.

Syntax

who [**-a** | **-b** **-d** **-i** **-l** **-m** **-p** **-q** **-r** **-s** **-t** **-u** **-w** **-A** **-H** **-T** **-X**] [*File*]

who am { **i** | **I** }

Description

The **who** command displays information about all users currently on the local system. The following information is displayed: login name, tty, date and time of login. Typing **who am i** or **who am I** displays your login name, tty, date and time you logged in. If the user is logged in from a remote machine, then the host name of that machine is displayed as well.

The **who** command can also display the elapsed time since line activity occurred, the process ID of the command interpreter (shell), logins, logoffs, restarts, and changes to the system clock, as well as other processes generated by the initialization process.

The general output format of the **who** command is as follows:

```
Name [State] Line Time [Activity] [Pid] [Exit] (Hostname)
```

where:

Item	Description
Name	Identifies the user's login name.
State	Indicates whether the line is writable by everyone (see the -T flag).
Line	Identifies the line name as found in the /dev directory.
Time	Represents the time when the user logged in.
Activity	Represents the hours and minutes since activity last occurred on that user's line. A . (dot) here indicates line activity within the last minute. If the line has been quiet more than 24 hours or has not been used since the last system startup, the entry is marked as old.
Pid	Identifies the process ID of the user's login shell.
Term	Identifies the process termination status (see the -d flag). For more information on the termination values, refer to the wait subroutine or to the /usr/include/sys/signal.h file.
Exit	Identifies the exit status of ended processes (see the -d flag).
Hostname	Indicates the name of the machine the user is logged in from.

To obtain information, the **who** command usually examines the **/etc/utmp** file. If you specify another file with the *File* parameter, the **who** command examines that file instead. This new file is usually the **/var/adm/wtmp** or **/etc/security/failedlogin** file.

If the *File* parameter specifies more than one file name, only the last file name will be used.

Note: This command only identifies users on the local node.

Flags

Item	Description
-a	Processes the /etc/utmp file or the named file with all information. Equivalent to specifying the -bdlprrTu flags.
-b	Indicates the most recent system startup time and date.
-d	Displays all processes that have expired without being regenerated by init. The exit field appears for dead processes and contains the termination and exit values (as returned by wait) of the dead process. (This flag is useful for determining why a process ended by looking at the error number returned by the application.)
-l	Lists any login process.
-m	Displays information about the current terminal only. The who -m command is equivalent to the who am i and who am I commands.
-p	Lists any active process that is currently active and has been previously generated by init.
-q	Prints a quick listing of users and the number of users on the local system.
-r	Indicates the current run-level of the process.
-s	Lists only the name, line, and time fields. This flag is the default; thus, the who and who -s commands are equivalent.
-t	Indicates the last change to the system clock by the root user using the date command. If the date command has not been run since system installation, the who -t command produces no output.
-u or -i	Displays the user name, tty, login time, line activity, and process ID of each current user.
-A	Displays all accounting entries in the /etc/utmp file. These entries are generated through the acctwtmp command.

Item	Description
-H	Displays a header (title).
-T or -w	Displays the state of the tty and indicates who can write to that tty as follows: + Writable by anyone. - Writable only by the root user or its owner. ? Bad line encountered.
-X	Prints all available characters of each user name instead of truncating to the first 8 characters. The user name is also moved to the last column of the output.

Exit Status

This command returns the following exit values:

Ite m	Description
0	Successful completion.
>0	An error occurred.

Examples

1. To display information about who is using the local system node, type:

```
who
```

Information similar to the following is displayed:

```
pts/1      Nov  9 00:20   long_username_greater_than_eight_characters  (localhost)
```

2. To display your user name, type:

```
who am i
```

Information similar to the following is displayed:

```
george lft/0 Jun 8 08:34
```

3. To display a history of logins, logouts, system startups, and system shutdowns, type:

```
who /var/adm/wtmp
```

Information similar to the following is displayed:

```
hank  lft/0   Jun   8   08:34  (ausnix5)
john  lft/0   Jun   8   08:34  (JIKey)
mary  lft/0   Jun   8   08:22  (machine.austin.ibm)
jan   pts4     Jun   8   09:19  (puff.wisc.edu)
```

4. To display the run-level of the local system node, type:

```
who -r
```

Information similar to the following is displayed:

```
. run-level 2 Jun 8 04:15 2 0 s
```

5. To display any active process that is currently active and has been previously generated by init, type:

```
who -p
```

Information similar to the following is displayed:

```
srcmstr . Jun 8 04:15 old 2896
cron . Jun 8 04:15 old 4809
uprintfd . Jun 8 04:15 old 5158
```

6. To process the `/var/adm/wtmp` file with the `-bdlpTtu` flags specified, type:

```
who -a /var/adm/wtmp
```

Information similar to the following is displayed:

```
.      system boot Jun 19 10:13
.      run-level 2  Jun 19 10:13
.      .           Jun 19 10:14    old
.      .           Jun 19 10:14    old
.      .           Jun 19 10:14    old
rc      -          Jun 19 10:13    old
.      .           Jun 19 10:16    old
.      .           Jun 19 10:14    old
srcmstr -          Jun 19 10:14    old
rctcpip -          Jun 19 10:14    old
rcdce   -          Jun 19 10:14    old
rccm    -          Jun 19 10:15    old
dceupdt -          Jun 19 10:15    old
rcnfs   -          Jun 19 10:15    old
cron    -          Jun 19 10:16    old
pioebe  -          Jun 19 10:16    old
qdaemon -          Jun 19 10:16    old
writesrv -         Jun 19 10:16    old
uprintfd -         Jun 19 10:16    old
.      .           Jun 19 10:16    old
LOGIN   - lft0     Jun 19 10:16    old
.      .           Jun 19 10:16    old
.      .           Jun 19 10:16    old
```

Files

Item	Description
/etc/utmp	Contains user and accounting information.
/etc/security/failedlogin	Contains the history of all invalid logins.
/var/adm/wtmp	Contains the history of all logins since the file was last created.
/usr/include/sys/signal.h	Contains a list of termination values.

whoami Command

Purpose

Displays your login name.

Syntax

```
whoami
```

Description

The **whoami** command displays your login name. Unlike using the command **who** and specifying **am i**, the **whoami** command also works when you have root authority since it does not examine the `/etc/utmp` file.

Files

Item	Description
<code>/etc/passwd</code>	Contains user IDs.

whodo Command

Purpose

Lists the jobs being performed by users on the system.

Syntax

whodo [-h] [-l] [-X] [User]

Description

Prints information on all processes for a terminal, as well as the child processes.

By default, the output generated by the command for each active logged user will contain name of the terminal, user ID, date login time. The output is headed by the date, time and machine name. This information is followed by a record of active processes associated with that user ID. Each record shows the terminal name, process-ID, CPU minutes and seconds used, and process name.

Flags

Item	Description
-h	Suppress the heading that is printed on the output.
-l	Produce a long form of output. A summary of the current activity on the system is printed. The summary includes the following: User Who is logged on. tty Name of the tty the user is on. login@ Time of day the user logged on. idle Number of minutes since a program last attempted to read from the terminal. JCPU System unit time used by all processes and their children on that terminal. PCPU System unit time used by the currently active process. what Name and parameters of the current process. The heading line of the summary shows the current time of day, how long the system has been up, the number of users logged into the system.
-X	Prints all available characters of each user name instead of truncating to the first 8 characters. The user name is also moved to the last column of the output.

Parameters

Item	Description
<i>User</i>	Limits output to all sessions pertaining to the user specified with <i>User</i> . More than one user name cannot be specified at a time.

Exit Status

- 0
The command completed successfully.
- >0
An error occurred.

Examples

1. When the **whodo** command is invoked on host "linguist" without any flags or parameters, the output looks similar to the following:

```
Sun Jul 28 16:27:12 2002
linguist

lft0    jeffg      8:15
?       4136      0:00    dtlogin
?       3408      4:55    dtsession
?       2072      4:37    dtwm
?       17310     0:00    dtexec
?       20904     5:53    dtterm
pts/0   22454     0:00    ksh
pts/0   4360      0:07    ksh
pts/0   25788     0:00    whodo
?       23672     0:00    dtexec
?       27536     0:00    dtterm
pts/3   21508     0:00    ksh
?       23888     0:00    dtexec
?       24384     2:49    dtterm
pts/2   24616     0:00    ksh
pts/2   25002     0:04    ksh
pts/2   26110     0:00    ksh
?       25276     0:00    dtexec
?       27090     0:31    dtterm
pts/1   24232     0:00    ksh
pts/1   23316     0:01    ksh
?       12566     4:23    dtfile
?       21458     1:35    dtfile

pts/0   jeffg      8:16
pts/0   22454     0:00    ksh
pts/0   4360      0:07    ksh
pts/0   25788     0:00    whodo

pts/1   jeffg      17:8
pts/1   24232     0:00    ksh
pts/1   23316     0:01    ksh

pts/2   jeffg      17:20
pts/2   24616     0:00    ksh
pts/2   25002     0:04    ksh
pts/2   26110     0:00    ksh

pts/3   root      16:26
pts/3   21508     0:00    ksh
```

2. The command **whodo -l** on the host linguist produces the following output:

```
04:33PM up 20 day(s), 22 hr(s), 51 mins(s) 5 user(s)
User  tty      login@      idle      JCPU      PCPU      what
jeffg lft0      08Jul02    21day(s)          /usr/sbin/getty /de
jeffg pts/0     08Jul02          14:00      7 whodo -l
jeffg pts/1     16Jul02    10day(s)         44      9 /usr/bin/ksh
jeffg pts/2     12Jul02          11      8:39      4 /usr/bin/ksh
root  pts/3     04:26PM          7          -ksh
```

3. The command `whodo -lX` on the host `kq11` produces the following output:

```
12:50AM    up 3 day(s),  1 hr(s), 41 mins(s)  4 user(s)
tty        login@        idle      JCPU    PCPU    what      User
tty0       Wed11PM    2day(s)          -ksh     root
pts/0      12:12AM          tn 0       root
pts/1      12:20AM          whodo -lX
long_username_greater_than_eight_characters
pts/2      Fri05AM    2day(s)          -ksh     root
```

Files

Item	Description
<code>/usr/sbin/whodo</code>	Contains the whodo command.
<code>/etc/utmp</code>	Contains the list of users.

whois Command

Purpose

Identifies a user by user ID or alias.

Syntax

`whois [ -h HostName ] [ . | ! ] [ * ] Name [ ... ]`

`whois ?`

Description

The `/usr/bin/whois` command searches a user name directory and displays information about the user ID or nickname specified in the *Name* parameter. The **whois** command tries to reach ARPANET host `internic.net` where it examines a user-name database to obtain information. The **whois** command should be used only by users on ARPANET. Refer to RFC 812 for more complete information and recent changes to the **whois** command.

Note: If your network is on a national network, such as ARPANET, the host name is hard-coded as `internic.net`.

The *Name* [...] parameter represents the user ID, host name, network address, or nickname on which to perform a directory search. The **whois** command performs a wildcard search for any name that matches the string preceding the optional ... (three periods).

Flags

Item	Description
.	Forces a name-only search for the name specified in the <i>Name</i> parameter.
!	Displays help information for the nickname or handle ID specified in the <i>Name</i> parameter.
*	Displays the entire membership list of a group or organization. If there are many members, this can take some time.
?	Requests help from the ARPANET host.
-h <i>HostName</i>	Specifies an alternative host name. The default host name on the ARPANET is <code>internic.net</code> . You can contact the other major ARPANET user-name database, <code>nic.ddn.mil</code> , by specifying the -h <i>HostName</i> flag.

Examples

1. To display information about ARPANET registered users by the name of Smith, enter:

```
whois Smith
```

2. To display information about ARPANET registered users that use the handle Hobo, enter:

```
whois !Hobo
```

3. To display information about ARPANET registered users with the name of John Smith, enter:

```
whois .Smith, John
```

4. To display information about ARPANET registered users whose names or handles begin with the letters HEN, enter:

```
whois HEN ...
```

5. To get help information for the **whois** command, enter:

```
whois ?
```

whom Command

Purpose

Manipulates Message Handler (MH) addresses.

Syntax

```
whom [ -alias File ... ] [ -nocheck | -check ] [ { -draftfolder +Folder | -nodraftholder | File }  
{ -draftmessage Message | -draftFile } ]
```

Description

The **whom** command does the following:

- Expands the headers of a message into a set of addresses.
- Lists the addresses of the proposed recipients of a message.
- Verifies that the addresses are deliverable to the transport service.

Note: The **whom** command does not guarantee that addresses listed as being deliverable will actually be delivered.

A message can reside in a draft folder or in a file. To specify where a message resides, use the **-draft**, **-draftfolder**, or **-draftmessage** flag.

If you do not specify the **-draftfolder** flag or if the `Draft-Folder:` entry in the `$HOME/.mh_profile` file is undefined, the **whom** command searches your MH directory for a **draft** file. Specifying a message after the **-draftfolder** *+Folder* flag is the same as specifying the **-draftmessage** flag.

Flags

Item	Description
-alias <i>File</i>	Specifies a file to search for mail aliases. By default, the system searches the <code>/etc/mh/MailAliases</code> file.
-draft	Uses the header information in the <code>UserMhDirectory/draft</code> file if it exists.

Item	Description
-draftfolder <i>+Folder</i>	Uses the header information from the draft message in the specified folder. If you specify a draft folder that doesn't exist, the system creates one for you.
-draftmessage <i>Message</i>	Uses the header information from the specified draft message.
-help	Lists the command syntax, available switches (toggles), and version information.
	Note: For MH, the name of this flag must be fully spelled out.
<i>Message</i>	Specifies the message draft. Use the following to specify messages: Number Number of the message. cur or . (period) Current message. This is the default. first First message in a folder. last Last message in a folder. next Message following the current message. prev Message preceding the current message.
-nodraftfolder	Undoes the last occurrence of the -draftfolder <i>+Folder</i> flag.
	Note: Two other flags, -check and -nocheck , are also available. These flags have no effect on how the whom command performs verification. The -check and -nocheck flags are provided for compatibility only.

Profile Entries

The following entries are entered in the *UserMhDirectory/.mh_profile* file:

Item	Description
Draft-Folder:	Sets your default folder for drafts.
postproc:	Specifies the program used to post messages.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

To list and verify the addresses of the proposed recipients of a message, enter the addressees and subject of the message at the respective prompt, as follows:

```
To: d77@nostromo
Subject: a test
```

When prompted again, enter the text of the message:

```
-----Enter initial text
test
-----
```

After the whatnow prompt, enter the **whom** command:

```
whatnow>>> whom
```

The address of the proposed recipients of the message is then displayed:

```
lance...
d77@nostromo... deliverable
```

Files

Item	Description
\$HOME/.mh_profile	Specifies the MH user profile.
/usr/bin/whom	Contains the whom command.

wlmassign command

Purpose

Manually assigns processes to a workload management class or cancels prior manual assignments for processes.

Syntax

```
wlmassign [ -s | -S ] [ -u | Class_Name ] [ pid_list ] [ -g pgid_list ]
wlmassign [ -t { tag [ -i inheritance ] | -r } [ pid_list ] [ -g pgid_list ]
```

Description

The **wlmassign** command:

- Assigns a set of processes specified by a list of process identifiers (*PID*) or process group identifiers (*PGIDs*) to a specified superclass or subclass or both, thus overriding the automatic class assignment or a prior manual assignment.
- Cancels a previous manual assignment for the processes specified in *pid_list* or *pgid_list*.
- Assigns the Workload Manager (WLM) tag process attribute to a set of processes specified by a list of *pids* or *pgids*.
- Removes the WLM tag process attribute from a set of processes specified by a list of *pids* or *pgids*.

In addition to the tag, additional inheritance suboptions can be specified and these suboptions indicates to the WLM if a child process must inherit the tag from its parent after **fork** or **exec** subroutine.

The process requires at least the **SIGPRIV** privileges or higher for tagging another process.

The WLM tag assignment remains in effect until one or more of the following conditions are true:

- The tag is removed by using the **-r** flag.
- The tagged process is ended.
- The tag is overwritten with a new tag.

When a WLM tag is assigned to a process that belongs to a class that has the inheritance property turned off, it is automatically reclassified according to the current assignment rules and the new tag is taken into account during reclassification. The WLM tag is effective if class inheritance attribute is not specified for the current process class. To override the class inheritance attribute in favor of reclassification based on tag rules, the **/usr/samples/kernel/wlmtune** command available in the **bos.adt.samples** PTF, can be used to modify the behavior of WLM. The related tunables follow:

tag_override_super

Indicates to the WLM that superclass inheritance is bypassed in favor of the rule-based classification if a rule matches the process tag. The default value is **0**.

tag_override_sub

Indicates to the WLM that subclass inheritance is bypassed in favor of the rule-based classification if a rule matches the process tag. The default value is **0**.

For the tag assignment rules-based classification to be effective, the tunable values must be set prior to a WLM reclassification update.

The interactions between automatic assignment (inheritance and rules), inheritance, and manual assignment are detailed in the [Workload management in Operating system and device management](#).

The **wlmassign** command allows to specify processes using a list of PIDs, a list of PGDIDs, or both. The formats of these lists follow:

```
pid[,pid[,pid[...]]]
```

```
pgid[,pgid[,pgid[...]]]
```

The name of a valid superclass or subclass must be specified to manually assign the target processes to a class. If the target class is a superclass, each process is assigned to one of the subclasses of the specified superclass according to the assignment rules for the subclasses of this superclass.

A manual assignment remains in effect (and a process remains in its manually assigned class) until:

- The process terminates
- Workload Management (WLM) is stopped. When WLM is restarted, the manual assignments in effect when WLM was stopped are lost.
- The class the process has been assigned to is deleted
- A new manual assignment overrides a prior one.
- The manual assignment for the process is canceled using the **-u** flag.
- The process calls the **exec()** routine.

The name of a valid superclass or subclass must be specified to manually assign the target processes to a class. The assignment can be done or canceled at the superclass level, the subclass level or both. When a manual assignment is canceled for a process, or the process calls **exec()**, the process is then subject to automatic classification; if inheritance is enabled for the class that the process is in, it will remain in that class, otherwise the process will be reclassified according to the assignment rules.

For a manual assignment:

- If the *Class_Name* is the name of a superclass, the processes in the list are assigned to the superclass. The subclass is then determined, for each process, using the assignment rules for the subclasses of the target superclass.
- If the class name is a subclass name (*supername.subname*), the processes by default are assigned to both the superclass and the subclass. The processes can be assigned to the superclass only by specifying the **-S** flag or the subclass only by specifying the **-s** flag.

```
wlmassign  super1.sub2  -S  pid1
```

is equivalent to:

```
wlmassign super1 pid1
```

To assign a process to a class or cancel a prior manual assignment, the user must have authority both on the process and on the target class. These constraints translate into the following:

- The root user can assign any process to any class.
- A user with administration privileges on the subclasses of a given superclass (that is, the user or group name matches the user or group names specified in the attributes **adminuser** and **admingroup** of the superclass) can manually reassign any process from one of the subclasses of this superclass to another subclass of the superclass.
- Users can manually assign their own processes (same real or effective user ID) to a class, for which they have manual assignment privileges (that is, the user or group name matches the user or group names specified in the attributes **authuser** and **authgroup** of the superclass or subclass).

This defines 3 levels of privilege among the persons who can manually assign processes to classes, root being the highest. For a user to modify or terminate a manual assignment, they must have at least the same level of privilege as the person who issued the last manual assignment.

Note: The **wlmassign** command works with currently loaded WLM configuration. If the current configuration is a set, and the assignment is made to a class which does not exist in all configurations in the set, the assignment will be lost when a configuration that does not contain the class becomes active (class is deleted).

Flags

Item	Description
-g <i>pgid_list</i>	Indicates that the following list is a list of PGID.
-S	Specifies that the assignment is to be done or canceled at the superclass level only. This flag is used with a subclass name of the form <i>supername.subname</i> .
-s	Specifies that the assignment is to be done or canceled at the subclass level only. This flag is used with a subclass name of the form <i>supername.subname</i> .
-u	Cancel any manual assignment in effect for the processes in the pid_list or the pgid_list . If none of the -s or -S flags are used, this cancels the manual assignments for both the superclass and the subclass level.
-r	Removes a WLM tag from the specified process or the process group list.
-t <i>tag</i>	Sets a WLM tag for the specified process or the process group list.
-i <i>inheritance</i>	Specifies one or both tag inheritance sub-options in a comma-separated list. The following tag inheritance sub-options can be specified: fork Specifies that the children of this process should inherit the parent tag across fork. exec Specifies that the process retains its tag after a call to exec.

wlmcheck command

Purpose

Check automatic assignment rules and/or determines the Workload Manager class a process with a specified set of attributes would be classified in.

Syntax

wlmcheck [**-d** *Config*] [**-a** *Attributes*] [**-q**]

Description

The **wlmcheck** command with no arguments, gives the status of Workload Manager (WLM) and makes some coherency checks:

- Displays the current status of WLM (running/non running, active/passive, rsets bindings active, total limits enabled).
- Displays the status files that report the last loading errors, if any. If 'current' configuration is a set, this applies to all configurations in the set, and messages logged by the WLM daemon are reported.
- Checks the coherency of the attributes and assignment rules file(s) (such as, the existence of the classes, validity of user and group names, existence of application file names, etc).

If the **-d** *Config* flag is not specified, the checks are performed on the 'current' configuration.

The **wlmcheck** command can apply to a configuration set. In this case, the checks mentioned above are performed on all configurations of the set, after checking the set itself. Superclass names are reported in the form 'config/superclass' to indicate the regular configuration which they belong to.

Specifying a configuration with **-d** *Config* performs the checks on the *Config* configuration or set instead of 'current'. This does not change the reporting of status files and of the WLM daemon log, which only applies to the active configuration.

With the **-a** flag, **wlmcheck** displays the class that the process with attributes specified by *Attributes* would be assigned to, according to the rules for the current or specified configuration or configuration set. The format of the *Attributes* string is similar to an entry in the *rules* file, with the following differences:

- The class field is omitted (it is actually an output of **wlmcheck**)
- Each field can have at most one value. Exclusion (!), attribute groupings (\$), comma separated lists, and wild cards are not allowed. For the *type* field, the AND operator "+" is allowed, since a process can have several of the possible values for the type attribute at the same time. For instance a process can be a 32 bit process and call plock, or be a 64 bit fixed priority process.
- At least one field must be specified (have a value different from a hyphen (-)).

In addition, the first 2 fields are mandatory. The other fields, if not present default to a hyphen (-) which mean that any value in the corresponding field of an assignment rule is a match. When one or more of the fields in the attribute string are either not present or specified as a hyphen (-), the string is likely to match more than one rule. In this case, **wlmcheck** displays all the classes corresponding to all the possible matches.

Example of valid attribute strings:

```
$ wlmcheck -a "- root system /usr/lib/frame/framemaker - -"
$ wlmcheck -a "- - staff - 32bit+fixed"
$ wlmcheck -a "- bob"
```

Flags

Item	Description
-a <i>Attributes</i>	Passes a set of values for the classification attributes of the process in order to determine which class the process would be put into. This is a way to check that the assignment rules are correct and classify processes as expected.
-d <i>Config</i>	Uses the WLM property files in /etc/wlm/Config (which may indicate a set of time-based configurations) instead of /etc/wlm/current .
-q	Suppresses the output of the status of the latest activation/update of WLM and of messages logged by the WLM daemon (quiet mode).

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Files

Item	Description
classes	Contains the names and definitions of the classes.
limits	Contains the resource limits enforced on the classes.
rules	Contains the automatic assignment rules.
shares	Contains the resource shares allocated to the classes.

wlmcntrl Command

Purpose

Starts or stops the Workload Manager.

Syntax

wlmcntrl [[**-a** | **-c** | **-p**] [**-T** [**class** | **proc**] [**-g**] [**-d** *Config_Dir*] [**-o** | **-q**]

wlmcntrl **-u** [**-S** *Superclass* | **-d** *Config_Dir*]

Description

The **wlmcntrl** command stops, starts, updates or queries the state of Workload Manager (WLM). When starting or updating WLM, the WLM property files for the target configuration are pre-processed, and the data is loaded into the kernel. WLM can be started in two different modes:

- An active mode where WLM monitors and regulates the processor, memory and disk I/O utilization of the processes in the various classes.
- A passive mode where WLM only monitors the resource utilization without interfering with the standard operating system resource allocation mechanisms.

The active mode is the usual operating mode of WLM.

The classes, their limits and shares are described respectively in the **classes**, **limits**, and **shares** files. The automatic assignment rules are taken from the **rules** file. The class properties files for the superclasses of the WLM configuration **Config** are located in the subdirectory **/etc/wlm/Config**. The class properties files for the subclasses of the superclass **Super** of the configuration **Config** are located in **/etc/wlm/Config/Super**. The standard configuration shipped with the operating system is in **/etc/wlm/standard**. The current configuration is the one in the directory pointed to by the symbolic link **/etc/wlm/current**.

When the **-d** *Config_dir* flag is not used, **wlmcntrl** uses the configuration files in the directory pointed to by the symbolic link **/etc/wlm/current**.

When the **-d** *Config_dir* flag is used, **wlmcntrl** uses the configuration files in **/etc/wlm/Config_dir** and updates the **/etc/wlm/current** symbolic link to point to **/etc/wlm/Config_dir**, making **/etc/wlm/Config_dir** the current configuration. This is the recommended way to make **/etc/wlm/Config_dir** the current configuration.

When updating WLM using the **-u** flag, an empty string can be passed as *Config_dir* with the **-d** flag:

```
wlmcntrl -u -d ""
```

will simply refresh (reload) the assignment rules of the current configuration into the kernel without reloading the class definitions. This can be useful when a prior activation of WLM detected that some application files could not be accessed. After the system administrator has fixed the problems with either the rules or the files, this command can be used to reload only the rules.

The WLM configuration **Config** may also be a set of time-based configurations, in which case the subdirectory **/etc/wlm/Config** does not contain the properties files, but a list of configurations and the times of the week when they apply. The properties files are still in the subdirectory of each regular configuration of the set. When WLM is started or updated with such a set, a daemon is responsible for switching regular configurations of the set when the applicable one changes.

Note: This command is not supported when executed within a workload partition.

Flags

Item	Description
-a	Starts WLM in active mode or switches from passive to active mode. This is the default when no flag other than -d , -g , or -T is specified.
-c	Starts WLM in processor-only mode or switches from any mode to processor-only mode. In this mode, the WLM accounts for all resources, but only processor resource is regulated.
-d <i>Config_dir</i>	Uses /etc/wlm/Config_dir as an alternate directory for the WLM configuration (containing the classes, limits, shares and rules files) or configuration set (containing the list of configurations and the time ranges when they apply). This makes /etc/wlm/Config_dir the current configuration. This flag is effective when starting the WLM in active, processor-only or passive mode, or when updating the WLM. This flag cannot be used in conjunction with the -o and -q flags or when switching from a mode (among active, processor-only and passive) to another.
-g	Instructs WLM to ignore any potential resource set bindings. This means that all classes have access to the whole resource set of the system, regardless of whether or not they use a restricted resource set.
-o	Stops Workload Manager.
-p	Start WLM in passive mode or switches from any mode to passive mode. In this mode, the WLM accounts for all resources, but no resource is regulated.

Item	Description
-q	Queries the current state of WLM. Returns: 0 WLM is running in active mode. 1 WLM is not started. 2 WLM is running in passive mode. 3 WLM is running in active mode with no rset bindings. 4 WLM is running in passive mode with no rset bindings. 5 WLM is running in active mode for processor only 6 WLM is running in active mode for processor only with no rset bindings. 16 WLM is running in active mode, process total accounting is off. 18 WLM is running in passive mode, process total accounting is off. 19 WLM is running in active mode with no rset bindings, process total accounting is off. 20 WLM is running in passive mode with no rset bindings, process total accounting is off. 21 WLM is running in active mode for processor only, process total accounting is off. 22 WLM is running in active mode for processor only with no rset bindings, process total accounting is off. 32 WLM is running in active mode, class total accounting is off. 34 WLM is running in passive mode, class total accounting is off. 35 WLM is running in active mode with no rset bindings, class total accounting is off. 36 WLM is running in passive mode with no rset bindings, class total accounting is off. 37 WLM is running in active mode for processor only, class total accounting is off. 38 WLM is running in active mode for processor only with no rset bindings, class total accounting is off. 48 WLM is running in active mode, class and process total accounting are off. 50 WLM is running in passive mode, class and process total accounting are off. 51 WLM is running in active mode with no rset bindings, class and process total accounting are off. 52 WLM is running in passive mode with no rset bindings, class and process total accounting are off. 53 WLM is running in active mode for processor only, class and process total accounting are off. 54 WLM is running in active mode for processor only with no rset bindings, class and process total accounting are off. A message indicating the current state of WLM is printed on STDOUT.
-S Superclass	Requests an update of WLM that is limited to the subclasses of the Superclass. Use this flag with the -u flag. If the running configuration is a set of time-based configurations, Superclass must be given in the form "config/Superclass" where "config" is the regular configuration of the set which the Superclass belongs to. If "config" is the currently active configuration of the set, the changes will take effect immediately, else they will take effect at the next time "config" will be made active.
-T	Disables both class and process total limits accounting and regulation.
-T class	Disables only class total limits accounting and regulation.
-T proc	Disables only process total limits accounting and regulation.

Item	Description
-u	Updates the WLM. A single update operation can change the attributes, limits and shares of existing classes and/or add or remove classes. If the running configuration is a set, this operation refreshes the set description along with the content of all configurations of the set. Update can be used by a user with root authority to switch to an alternate configuration or configuration set. Update can also be used by a superclass administrator to update only the subclasses of the superclass he has administrative access to (using the -S flag).

Security

Access Control: Starting, stopping, switching from one mode to another, and updating superclasses or a configuration set requires root privileges. Updating the subclasses of a given superclass requires only admin user or admin group privileges (superclass administrator). Any user can query the state of WLM.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see Privileged Command Database in Security. For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Files

Item	Description
classes	Contains the names and definitions of the classes.
limits	Contains the resource limits enforced on the classes.
rules	Contains the automatic assignment rules.
shares	Contains the resource shares allocated to the classes.
description	Contains the description text for each configuration.
groupings	Contains attribute value groupings for the configuration

wlmstat Command

Purpose

Shows Workload Manager (WLM) per class resource utilization statistics.

Syntax

wlmstat [**-l** *Class* | **-t** *Tier*] [**-S** | **-s**] [**-@**] [**-c**] [**-m**] [**-b**] [**-B** *Device*] [**-T**] [**-a**] [**-w**] [**-v**] [*Interval*] [*Count*]

wlmstat [**-l** *Class* | **-t** *Tier*] [**-@**] [**-c**] [**-m**] [**-b**] [**-u**] [*Interval*] [*Count*]

wlmstat [**-l** *Class* | **-t** *Tier*] [**-@**] [**-M**] [**-S** | **-s**] [**-w**] [**-v**] [*Interval*] [*Count*]

Description

The **wlmstat** command symbolically displays the contents of WLM data structures retrieved from the kernel. If a *Count* is specified, **wlmstat** loops *Count* times and sleeps *Interval* seconds after each block is displayed. If *Interval* and *Count* are not specified, one output report is produced. If *Interval* is specified but no *Count* is given, **wlmstat** outputs results continuously at the given interval until stopped by a signal (SIGINTR, SIGQUIT, and SIGKILL). By default, **wlmstat** displays the statistics for all the resources for every superclass and subclass. You can specify flags to narrow the focus of the statistics to a type of resource, tier, superclass, or subclass and alter the output format.

Note: The following should be considered when viewing the **wlmstat** output:

1. Starting with AIX 5.3, the WLM processor usage values and process priority adjustments are updated 10 times per second by default.
2. The value displayed for processor usage is not the current instantaneous usage from the last second, but is instead an average of the last *N* readings (starting with AIX 5.3, the default value for *N* is 15).

3. The Unmanaged class is used to report system interrupt time and for tracking memory usage for all of the pinned pages in the system that are not managed by the WLM. No processes are assigned to this class.

It is possible for a process with a hard limit of 50 percent to use more than 50 percent of the processor between two consecutive WLM usage updates. Each tenth of a second, every process is assigned a priority, and the scheduler then schedules all processes based on their assigned priorities. A process might receive more of the processor resources than the process hard limit between WLM updates.

By default, each instantaneous value of processor usage from each update is kept for the following 15 readings and is averaged with the other 14 readings before being displayed by **wlmstat**. This can potentially result in a value of greater than 50 percent due to a single instance of more than 50 percent usage between WLM updates.

The priority of a process will be greatly reduced and the process will be unable to run if the process consistently reaches or exceeds its hard limit. Over the long term, the resource utilization of the process must be at or under the process hard maximum. Over a short time interval, **wlmstat** may show the process using more than the process hard limit. The **/usr/samples/kernel/wlmtune** command that is available in the bos.adt.samples PTF can be used to modify the behavior of WLM in such an instance. The related tunables are:

schedhz

The frequency at which the WLM scheduler recalculates class consumption and priority for processor. The default is 10. Modifying this value changes the responsiveness of WLM. Increasing this value causes WLM to update more frequently, thereby reducing the possibility of a process exceeding its hard limit during a short time interval. The trade-off for this is increased overhead, since more WLM processing occurs. This can potentially affect overall system performance.

cpuhist

The number of consecutive processor consumption values used in the average calculation. The default is 15. Increasing this value further smooths the reported processor usage values by averaging over a longer period.

To make WLM more responsive so that classes do not exceed their maximums over long periods, it is recommended that you first try modifying **schedhz** until the **wlmstat** output displays the desired results. You may want to also modify **cpuhist** so that **wlmstat** averages over the same time interval. For example, if **schedhz** is 20 and **cpuhist** is 15, **wlmstat** will average over a period of 0.75 seconds (15/20), so you may want to change **cpuhist** to 30 so that **wlmstat** still averages over 1.5 seconds.

On systems with no contention for processor, an *Interval* of 5 for **wlmstat** is recommended in order to adhere to WLM limits.

Flags

Item	Description
-@	Displays workload partition resource information.
-a	Displays subclass consumption in absolute terms. By default, the subclass consumption percentages are shown relative to the superclass consumption. With this option, subclass consumption is displayed relative to the total amount of resource available on the system (as is done for superclasses). All values are displayed with 1% precision. For instance, if a superclass has a processor target of 20% and the processor percentage shown by wlmstat without -a for a subclass is 10%, wlmstat with -a shows the processor percentage for the subclass as 2%.
-b	Displays only disk I/O statistics.
-B Device	Displays disk I/O device statistics. Passing an empty string (-B "") displays the statistics for all the disks accessed by the class.
-c	Shows only processor statistics.

Item	Description
-l <i>Class</i>	Displays statistics for <i>Class</i> name. If not specified, all classes display along with a summary for appropriate fields.
-m	Shows only physical memory statistics.
-M	Displays the Real/Virtual Memory statistics. Use of the -M option adds the following columns in the output: RMSIZ Utilized real memory size for the class VMSIZ Utilized virtual memory size for the class RMLIM Real memory limit for the class VMLIM Virtual memory limit for the class LGPGSIZ Utilized large pages in the class LGPG LIM Large page limit for the class Note: A - will be displayed for the RMLIM, VMLIM, and LGPG LIM fields if the limit is undefined. When the -M and -w options are used together, RMSIZ and VMSIZ fields contain the high watermarks for these attributes instead of the actual utilized values. In addition, the LGPGSIZ and LGPG LIM fields is turned off.
-s	Displays only subclass statistics.
-S	Displays only superclasses statistics.
-t <i>Tier</i>	Displays statistics only for the specified <i>Tier</i> .
-T	Displays the total numbers for resource utilization since WLM was started or the class was created, whichever is the latter. The units are: CPU The total processor time, in milliseconds, consumed by a class MEM Unused DKIO The total number of 512 byte blocks sent/received by a class for all the disk devices accessed.

Item	Description
-v	Specifies verbose mode. This flag, intended for trouble shooting, also displays some class attributes, resource shares and limits and other WLM parameters, including internal parameter values intended for AIX support personnel. The following information can be of interest for users: Column Header Description CLASS Class name. tr tier number (0 to 9) i Value of the inheritance attribute: 0 = no, 1 = yes. #pr Number of processes in the class. If a class has no (0) process assigned to it, the values shown in the other columns might not be significant. CPU Processor utilization of the class (%). MEM Physical memory utilization of the class (%). DKIO Disk IO bandwidth utilization for the classs (%). sha Number of shares ('-' is represented as -1) min Resource minimum limit (%) smx Resource soft maximum limit (%) hmx Resource hard maximum limit (%) des (desired): percentage goal (target) calculated by WLM using the shares numbers (%) npg Number of memory pages owned by the class. The other columns are for internal use only and bear no meaning for administrators and end users. This format is better used with a resource selector (-c, -m, or -b), otherwise the lines might be too long to fit into a line of a display terminal.
-w	Displays the memory <i>high water mark</i> , that is the maximum number of pages that a class had in memory at any given time since WLM was started or the class was created (whichever happened last).
-u	Displays per-tier and total unused resources.

Display

Results are tabulated, with the following fields:

Name	Class name
CPU	Percentage of total processor time consumed by the class.

Name	Class name
MEM	Percentage of physical memory consumed by the class.
DKIO	Percentage of the disk IO bandwidth consumed by the class. This number is the average of the disk bandwidth on all the disk devices accessed by the class, and is usually not significant. For instance if a class consumes 80% of the bandwidth of one disk and 5% of the bandwidth of two other disks, the DKIO column shows 30%. For details on the per device utilization, use the -B device option.

Examples

1. To get a printout of WLM activity right now, enter:

```
wlmstat
```

This produces the following output:

```

      CLASS CPU MEM DKIO
Unclassified  0  0  0
  Unmanaged  0  0  0
    Default  0  0  0
    Shared   0  0  0
    System   0  0  0
    class1  12  0  0
  class1.Default  4  0  0
  class1.Shared   0  0  0
class1.subclass1  4  0  0
class1.subclass2  4  0  0
    class2  12  0  0
  class2.Default  4  0  0
  class2.Shared   0  0  0
class2.subclass1  4  0  0
class2.subclass2  4  0  0

```

2. To get a report for superclass **class1**, enter:

```
wlmstat -l class1
```

This produces the following output:

```

      CLASS CPU MEM DKIO
    class1  12  0  0
  class1.Default  4  0  0
  class1.Shared   0  0  0
class1.subclass1  4  0  0
class1.subclass2  4  0  0

```

3. To get a report for subclass **sclass1.subclass2** updated every 10 seconds, for one minute, enter:

```
wlmstat -l class1.subclass2 10 6
```

This produces the following output:

```

      CLASS CPU MEM DKIO
class1.subclass2  4  0  0
class1.subclass2  4  0  0
class1.subclass2  4  0  0
class1.subclass2  4  0  0
class1.subclass2  4  0  0
class1.subclass2  4  0  0

```

4. To display virtual/real memory statistics, enter:


```
wlmstat -M
```

This produces the following output:

CLASS	RMSIZ	RMLIM	VMSIZ	VMLIM	LGPGSIZ	LGPGLIM
Unmanaged	1024	4096	4096	8192	0	-
Default	0	-	0	-	0	-
Shared	0	-	0	-	0	-
System	23567	50000	819234	1000000	0	-

5. To display the memory high water mark, enter:

```
wlmstat -M -w
```

This produces the following output:

CLASS	RMSIZ	RMLIM	VMSIZ	VMLIM
Unmanaged	1024	4096	4096	8192
Default	0	-	0	-
Shared	0	-	0	-
System	23567	50000	819234	1000000

Errors

A warning message is issued by **wlmstat** if WLM is not started.

wol command

Purpose

Wakes up one or more hosts that are connected to a network in suspend mode by sending a Magic Packet.

Syntax

To send a Magic Packet to a subnet-directed broadcast address:

```
wol { -m MACAddress [ -h Host -s SubnetMask ] | -i Interface ] | -f File } [ -v ]
```

To send a Magic Packet to a multicast address:

```
wol { -m MACAddress -M MulticastAddress [ -p Port ] | -i Interface ] | -f File } [ -v ]
```

Description

The **wol** command wakes up one or more hosts that are connected to a network in suspend mode by sending a Magic Packet to the specified address or addresses on the specified subnet.

If the user doesn't specify either the **-h**, nor **-s** flag, the **wol** manager will broadcast the Magic Packet as follows:

- If the user specifies the interface name (**-i** *Interface*), the Magic Packet will be broadcast from the specified interface.
- If the user doesn't specify the interface name, then the **wol** manager will loop through each network interface installed on the machine. If an interface is up, it will broadcast the Magic Packet from that interface, and then continue to the next interface until it goes through the entire interface list on the machine.

The file specified with **-f** *File* contains the list of hosts which need to be awakened. This file consists of one or more lines, each line containing the following information in this format:

MacAddress; Hostname/IPaddress; SubnetMask; Multicast; Port; Interface

For example, the file might look like this:

```
00:20:35:7a:7:89a;      9.41.86.19;      255.255.255.0  ; ; ;
00:04:ac:17:c0:9f ;      obiwan.aoot.austin.ibm.com;      255.255.255. 224; ; ;
00:07:be:4a:2:394; ; ; ; en0
00:06:38:6b:7e:8f ;      ; ; 234.5.6.7; 12345 ;
```

A line starting with a "#" character is a comment and is ignored. Each line contains 6 tokens separated by ";" character. The MAC address is mandatory. The other tokens are optional, but the ";" character must be used to separate unused tokens.

Flags

Item	Description
-i <i>Interface</i>	Specifies the interface to use on the host where the wol command is being run
-f <i>File</i>	Specifies the name of a file containing a group list. This allows the user to wake a specified group of hosts.
-h <i>Host</i>	Specifies a host to wake, either as a hostname or as an IPv4 address in dot string representation (for example, 10.0.0.3).
-m <i>MACAddress</i>	Specifies the a 48 bits MAC address of the host in hex representation (for example, 00:20:35:7a:78:9a).
-M <i>MulticastAddress</i>	Specifies an IPv4 multicast address.
-p <i>Port</i>	Specifies the port to use on the multicast machine.
-s <i>SubnetMask</i>	Specifies an IPv4 subnet mask in dot string representation (for example, 255.255.255.0).
-v	Specifies verbose mode.

Exit Status

Item	Description
0	The command completed successfully.
>0	An error occurred.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Location

/usr/sbin/wol

wparerr Command

Purpose

Logs an error message for a specific Workload partition (WPAR).

Syntax

From global

```
wparerr -w WparName [{-c <cat file> -s <set no> -n <msg no>}] -S default_string args
```

From WPAR

```
wparerr [-c <cat file> -s <set no> -n <msg no>}] -S default_string args
```

Description

The **wparerr** command provides a mechanism to log error messages for a given WPAR. Each WPAR can hold up to 1 KB of error messages. If there is enough space to log a new message, the command logs the message; otherwise, it fails. The *-w* option should not be used inside a WPAR. Everything after *-S* flag is treated as arguments for the message.

Flags

Item	Description
<i>-w</i>	Specifies the name of the workload partition for which the message should be logged.
<i>-c</i>	Specifies the catalog file name to be used for translation.
<i>-s</i>	Specifies the message set number of the error message in the catalog file.
<i>-n</i>	Specifies the message number of the error message.
<i>-S</i>	Specifies the default message string. Follows the same syntax as the printf subroutine <i>Format</i> parameter. Floating point is not supported.
<i>args</i>	Specifies the arguments to the message if any.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database](#) in Security. For a list of privileges and the authorizations that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Example

- To log a message for the WPAR *mywpar* from global:

```
wparerr -w mywpar -c wparerrs.msg -s 1 -n 12 -S " %s failed with return value %d\n" lswpar -1
```

- To log a command failure from inside a WPAR:

```
wparerr -c wparerrs.msg -s 1 -n 5 -S " %s application does not allow checkpoint\n" myapplication
```

wparexec Command

Purpose

Creates an application workload partition or specification file.

Syntax

wparexec [**-a**] [**-c**] [**-1**] [**-F**] [**-h** *hostname*] [**-H** *architecture*] [**-i**] [**-I** *attribute=value ...*] ... [**-M** *attribute=value...*] ... [**-N** *attribute=value...*] ... [**-R** *attribute=value...*] [**-u** *userscript*] [**-v**] [**-x**] { **-n** *wparname* [**-e** *existingwpar* | **-f** *infile*] [**-o** *outfile* [**-w**]] | **-f** *infile* [**-n** *wparname*] [**-o** *outfile* [**-w**]] | **-w** **-o** *outfile* [**-n** *wparname*] [**-e** *existingwpar* | **-f** *infile*] } [**--**] [*var=value ...*] /*path/to/command* [*arg ...*]

Note:

Regardless of locale, only ASCII characters are allowed as arguments to the **wparexec** command.

In addition to this, there are more restrictions for a WPARs name:

- May not be more than 25 bytes.
- May not contain white space or any of the following symbols:

= : / ! ; ' " < > ~ & () * + [] , . ^ 0 { } | \

- May not start with '-' or '0'.

Description

The **wparexec** command builds and starts an application workload partition, or creates a specification file to simplify the creation of future application workload partitions.

An application workload partition is an isolated execution environment that might have its own network configuration and resource control profile. Although the partition shares the global file system space, the processes that are running therein are only visible to other processes in the same partition. This isolated environment allows process monitoring, gathering of resource, accounting, and auditing data for a predetermined cluster of applications.

The **wparexec** command starts and monitors a single application within this isolated environment. The **wparexec** command returns synchronously with the return code of this tracked process only when all of the processes in the workload partition terminate. For example, if the tracked process creates a daemon and exits with the 0 return code, the **wparexec** command will block until the daemon and all of its children terminate, and then exit with the 0 return code, regardless of the return code of the daemon or its children.

Flags

Item	Description
-1	Creates the configuration only. Causes the wparexec command to stop after creating the configuration of the application WPAR. The startwpar command must then be used to start the WPAR. Only advanced users can use the -1 option.
-a	Automatically resolves conflicting static settings if required. Resolvable settings are name, hostname, and network configuration.
-c	Enables this workload partition to be checkpointed. This option is only valid when additional checkpoint-restart software is installed and configured. When this option is used, any file systems associated with this workload partition (for example, with the -M option) must be remote (for example, vfs=nfs).
-e <i>existingwpar</i>	Uses an existing application workload partition as the source for specification data. Do not use the -e flag with the -f flag. Any values specified by other wparexec flags override those values from the existing workload partition.
-f <i>infile</i>	Indicates the specification file to read default values from. Do not use with the -e flag. Any values specified by other wparexec flags override those values from the loaded specification file.
-F	Suppresses or overrides most error conditions. With the -F flag, the wparexec command continues with a warning.
-h <i>hostname</i>	Specifies a host name for this workload partition. If not specified, the wparexec command uses the workload partition name as host name.

Item	Description
-H <i>architecture</i>	Creates an architecture-compatible workload partition. Valid architecture values are <code>pow4</code>, <code>ppc970</code>, <code>pow5</code>, <code>pow6</code>, and <code>pow7</code>. The architecture value must be earlier, or equal to, the system hardware version. The applications in the workload partition are presented with the lowest common denominator of the specified architecture. If the workload partition can create a checkpoint, the workload partition is able to migrate between systems with hardware levels greater than, or equal to, the workload partition architecture. Note: The POWER5 processor-based systems and BladeCenter JS21 Express systems are not compatible with each other. You cannot create a JS21-compatible (ppc970) WPAR on a POWER5 processor-based system even though the JS21 using an earlier processor than a POWER5 processor-based systems.
-i	Enables WPAR-specific routing for the workload partition. By default, outgoing network traffic from a workload partition is routed like it is being sent from the global environment, notably in the following ways: - Traffic between addresses that were hosted on the same global system is sent through the loopback interface. - Routing table entries that are configured in the global system, including the default route, are used to transmit workload partition traffic. If you enable WPAR-specific routing by specifying the <code>-i</code> flag, the workload partition creates and uses its own routing table for outgoing traffic. Routing entries are created automatically for each of the network addresses of the workload partition to accommodate broadcast, loopback, and subnet routes. For more information about the network attributes, see the <code>-N</code> flag. You can create explicit additions to the routing table of the workload partition using the <code>-I</code> flag. In particular, you can use the <code>-I</code> flag to configure the default route, as no default route is created automatically.
-I <i>attribute=value ...</i>	Adds routing table entries to the entries that are automatically created when WPAR-specific routing is in effect. You can specify more than one <code>-I</code> flag to configure multiple routes. Using the <code>-I</code> flag automatically enables WPAR-specific routing as described under the <code>-i</code> flag. You can specify the following attributes with the <code>-I</code> flag. The <code>rtdest</code> attribute and the <code>rtgateway</code> attribute are required to be specified. rtdest=destination Identifies the host or network to which you are directing the route. You can specify the value using either a symbolic name or a numeric address. You can use the keyword default to specify a default route. For more information about the route rtdest attribute, see the <i>Destination</i> parameter of the route command. rtgateway=gateway Identifies the gateway to which packets are addressed. You can specify the value using either a symbolic name or a numeric address. rtnetmask=A.B.C.D Specifies the network mask to the destination address. rtprefixlen=n Specifies the length of a destination prefix, which is the number of bits in the netmask. The value must be a positive integer. rttype={net host} Forces the rtdest attribute to be interpreted as the specified type. rtinterface=if Specifies the interface, for example, <code>en0</code>, to associate with the route so that packets are sent using the interface when the route is chosen. 61Vrtfamily={inet inet6} Specifies the address family. For information about the parameters of the rtfamily flag, see the parameter section of the route command.
-M <i>directory=dir [vfs=type] [dev=devicepath] [host=remotehost] [mountopts=mountopts]</i>	Specifies file system dependencies only. Attributes must be space-separated. By default, an application workload partition has the same level of access to all of the global file systems and mounts as the user who created the workload partition. Use the <code>-M</code> flag with the <i>directory</i> attribute that is set to the file system name to specify additional file systems. More than one <code>-M</code> flag can be specified. Note: All of the mounts and all of the directories are created and available at global level. File systems that are based on disk, such as the <code>vfs=jfs</code> and the <code>vfs=jfs2</code>, will not be created for application workload partitions. A local file system dependency can be added by defining only the <i>directory</i> attribute. However, the <i>directory</i> specified must exist in the <code>/etc/filesystems</code>. If an error occurs during the process of creating the workload partition, any file systems mounted by the <code>wparexec</code> command are unmounted. After the creation succeeds, the file systems are not unmounted, regardless of the return status of the user application. The following are the valid values for the <i>vfs</i> attribute for application workload partitions: nos The <i>directory</i> specified by the <i>dev</i> attribute on the system specified by the <i>host</i> attribute is mounted at the location that is specified by the <i>directory</i> attribute. If the mount point does not exist, it will be created. The only other attributes that are applicable to an <i>nfs</i> mount are the <i>mountopt</i> attributes, corresponding to the <code>-o</code> option of the mount command or the <i>options</i> attribute in an <code>/etc/filesystems</code> stanza. If not specified, no mount options are used by default. Acceptable option values correspond to the <code>-o</code> options to the mount command. namefs The global <i>directory</i> specified by the <i>dev</i> attribute is mounted over the <i>directory</i> specified by the <i>directory</i> attribute. The only other attributes that are applicable to a <i>namefs</i> mount are the <i>mountopt</i> attributes. directory The global <i>directory</i> specified by the <i>directory</i> attribute is created if it does not exist. No mounting is performed.
-n <i>wparname</i>	Specifies the name for the workload partition to be created. If no name is supplied, a name is generated.

Item

-N *attribute=value*

Description

Allows specification of the following network configuration attributes:

- **interface**= *if* or **interface**=*namemappedif*
- **address**=*A.B.C.D*
- **netmask**=*A.B.C.D*
- **broadcast**=*A.B.C.D*
- **address6**=*S:T:U:V:W:X:Y:Z*
- **prefixlen**=*n*

The name-mapped interface is defined in the **/etc/wpars/devmap** file. You can specify the mapping between the name-mapped interface and the system interface as follows:

```
# The comments start with '#'
# Each line contains a pair of name-mapped interface
# and real interface separated by tab or blank spaces.
foo en0
goo en1
soo en2
```

The *attribute=value* pairs must be separated by spaces. More than one **-N** flag can be used to configure multiple IP addresses. At minimum, the **address** or the **address6** attribute must be specified. The **wparexec** command collects any other values that are not specified from the global system settings. If no **-N** flag is specified, the **wparexec** command will attempt to discover an appropriate IP address for the workload partition by running the `gethostbyname` subroutine on the workload partition name specified with the **-n** flag. If an address is found on the same subnet as any global interface, the settings of that interface will be used with the resolved IP address to create the default network entry.

To define an IPv6 network configuration, specify the **-N** flag with the **address6** attribute, the **prefixlen** attribute, and the **interface** attribute:

- The **address6** attribute is a 128-bit address. The address is represented by eight 16-bit integers that are separated by colons. Each integer is represented by four hex digits. Leading zeros can be skipped, and consecutive null 16-bit integers can be replaced by two colons (one time per address).
- The **prefixlen** attribute is the number of high-order bits that are used to mask the IPv6 address and to comprise the prefix. The value of the **prefixlen** attribute ranges from 0 through 128. Each **-N** flag can accept either IPv4 attributes, or IPv6 attributes, but not both.

-o *outfile*

Indicates an output path and file name to write specification data to. This specification file can be used to create an application workload partition later with the **-f** flag.

Item

-R *attribute=value*

Description

Allows specification of resource control attributes. Only one -R flag can be specified. Most resource controls are similar to those supported by the Workload Manager (WLM). See the listed WLM pages for descriptions of these attributes. Valid attributes are as follows:

active

Allows resource control definitions to be retained, but rendered inactive. This attribute can take the *yes* or *no* values.

rset

Configures this workload partition to use a resource set created by the **mkcrset** command.

shares_CPU

The number of processor shares available to this workload partition. For more information about processor shares, see [Workload Manager shares File](#) in *Files Reference*.

CPU

The percentage of processor limits for this workload partition's processes. This attribute uses the following format to define the limits values:

```
CPU=<m>% - <SM>%, <HM>%
```

The *m* value represents the minimum limit. The *SM* value represents the soft maximum limit. The *HM* value represents the hard maximum limit. For more information about limited values, see the [Workload Manager limits File](#) in *Files Reference*.

shares_memory

The number of memory shares available to this workload partition. For more information about memory shares, see [Workload Manager shares File](#) in *Files Reference*.

memory

The percentage of memory limits for this workload partition's processes. For more information about memory limit, see the [Workload Manager limits File](#) in *Files Reference*.

procVirtMem

The maximum amount of virtual memory that a single process can use. Processes that exceed the specified limit are terminated. The valid units are megabytes (M or MB), gigabytes (G or GB), and terabytes (T or TB). The minimum limit allowed is 1M. The maximum limit that can be specified is 8796093022207M, 8589934591G, or 8388607T. If the value is set to -1 (no units), the limit is disabled. For more information about limit values, see [Workload Manager limits File](#) in *Files Reference*.

totalVirtMem

The maximum amount of virtual memory that can be used by the WPAR as a whole. Processes that cause the specified limit to be exceeded will be terminated. The valid range and units are the same as for `procVirtMem`. If the value is set to '-1' (no units), the limit is disabled. See [Workload Manager limits File](#) in *Files Reference*.

totalProcesses

The total number of processes that are allowed in this workload partition. For more information about allowed processor number, see [Workload Manager limits File](#) in *Files Reference*.

(Attributes for the -R flag, continued):

totalPTYs=n

The total number of pseudo terminals that are allowed in the workload partition. For more information about the allowed pseudo terminals, see [pty Special File](#).

totalLargePages=n

The number of large pages that can be allowed for the workload partition. For more information about the allowed large pages, see [Large Pages](#).

pct_msgIDs=n%

The percentage of the maximum number of message queue IDs of the system that are allowed in the workload partition. For more information about the allowed number of message queue IDs, see [Message Queue Kernel Services](#).

pct_semIDs=n%

The percentage of the maximum number of semaphore IDs of the system that are allowed in the workload partition.

pct_shmIDs=n%

The percentage of the maximum number of shared memory IDs of the system that are allowed in the workload partition. For more information about the allowed number of shared memory IDs, see [Shared Memory](#).

pct_pinMem=n%

The percentage of the maximum pinned memory of the system that can be allocated to the workload partition. For more information about pinned memory, see [Support for pinned memory](#).

totalThreads

The total number of threads that are allowed in this workload partition. For more information about allowed number of threads, see [Workload Manager limits File](#) in *Files Reference*.

Item	Description
-u <i>userscript</i>	Specifies the path to a user script to be run by the workload partition commands at various administration points. The parameter of the -u flag can be a quoted string including additional arguments to be passed to the script. In all cases, the first component of the parameter of the -u flag must be an absolute path to an existing executable file. The script is invoked as follows: <pre>/path/to/userScript<action><WAPR></pre> The <i>action</i> argument indicates the administrative action being performed, as follows: - WPAR_LOAD: A script runs in the global environment after kernel configuration, before the tracked process is created. If the script returns a value of non-zero, the workload partition will not be started. - WPAR_START: A script runs in the global environment after the workload partition becomes active. For application workload partitions, the script runs once the tracked process is started. Note: This code path can be run asynchronously by a dissociated process with its standard I/O streams closed or redirected. Internal messaging must be handled accordingly, and the script must account for the fact that short-lived workload partitions might be stopped or stopping at any point during the execution of the script. If the script returns a value of nonzero, a warning is logged, but no other behavior changes. - WPAR_STOP: A script runs in the global environment after all of the workload partition processes finish, and before the kernel is unconfigured. Note: This code path can be executed by a dissociated process with its standard I/O streams closed. Internal messaging should be handled accordingly. If the script returns a value of non-zero, a warning will be logged, but no other behavior will change.
-v	Specifies the command to run in the verbose mode.
-x	Allows access to cross-WPAR semaphores and shared memory segments.
-w	Writes the specification file. When it is used with the -o flag, the -w flag causes the <code>wparexec</code> command to quit after writing the new specification file without actually creating the workload partition.
[--] [<i>var=value</i>] <i>/path/to/command</i> [<i>arg ...</i>]	Specifies the application (tracked process) to be run within the workload partition, along with any necessary environment variable settings and arguments. The command is required, either by this command-line syntax or the general <code>.application</code> attribute in the specification file, unless the command is only creating a specification file (with -w flag). When it is started, the command line provided is always shell-expanded within the workload partition. When using the command line, shell metacharacters should be escaped appropriately to prevent premature expansion. The special double-minus separator (--) is used to signify that all subsequent command-line arguments comprise the tracked process. For example, use this separator to remove ambiguity between attributes to the -N flag and assignment of environment variables to the tracked process. Only one tracked process per workload partition is supported, but this application might create other processes. The workload partition is automatically stopped and removed when all of the processes therein terminate. A workload partition might be stopped and removed prematurely by the <code>stopwpar</code> command.

Security

Access Control: Only the root user can run this command.

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

1. To create an application workload partition that is running a benchmark program, enter:

```
wparexec -n tpcc -N address=192.168.0.51 /u/tpcc/benchmark -f /tmp/logfile
```

Note: The **-f** flag is passed to the `/u/tpcc/benchmark` file and is not processed by the `wparexec` command as a flag.

2. To create a workload partition based on an existing specification file, enter:

```
wparexec -f /tmp/wparexec1.spec
```

3. To override the default minimal *PATH* variable provided by the `wparexec` command, enter:


```
wparexec PATH=/usr/opt/bin:/usr/bin:/usr/sbin /home/joe/runapp
```

Files

Item	Description
/usr/samples/wpars/sample.spec	An annotated workload partition specification file.

wpar_reg_script Command

Purpose

Allows a user to register scripts to be run during different phases of a system WPAR live application mobility. Currently, only the WPAR restart phase is supported.

Syntax

```
/usr/lib/wpars/wpar_reg_script [ -r | -u ] [ -p phase ] [ -s /path/to/script ]
```

Description

The `wpar_reg_script` command allows a user to register scripts to be run during specific phases of WPAR live application mobility.

Registering a script will place an entry in a new ODM class called `CuWscr` that contains the full path to the script and its arguments.

This script can be used with kernel extensions and mobility to load kernel extensions on the arrival node before other processes start.

Restrictions

The `wpar_reg_script` command can only be run from inside a WPAR.

Only the file systems that are part of the WPARs config are available at the time of execution (not privately mounted file systems). These file systems are only available read-only.

No processes can be left running when the script exits.

Loading kernel extensions with this script for use with mobility is restricted to `SYS_SINGLELOAD`.

Flags

Item	Description
<code>-r</code>	Register an entry including its script and phase to the <code>CuWscr</code> class.
<code>-u</code>	Unregister an entry from the <code>CuWscr</code> class.
<code>-p</code>	The phase that the command will be executed during. Value of 1 corresponds to WPAR restart. More may be added in the future. If no value is specified, it will use a default value of 1.
<code>-s</code>	The path to the script and its arguments that will be executed. Surround the full command with <code>"</code> to include command line parameters.

Examples

1. To register the script `/usr/sbin/foo/` to execute:

```
/usr/lib/wpars/wpar_reg_script -r -s /usr/sbin/foo
```

2. To register the script `/usr/sbin/foo/` with arguments:

```
/usr/lib/wpars/wpar_reg_script -r -s "/usr/sbin/foo args"
```

3. To Unregister all instances of the script `/usr/sbin/foo/` foo:

```
/usr/lib/wpars/wpar_reg_script -u -s /usr/sbin/foo
```

4. To unregister all scripts that run during phase 1:

```
/usr/lib/wpars/wpar_reg_script -u -p 1
```

wparprnterr Command

Purpose

Displays error messages specific to a Workload partition (WPAR).

Syntax

From global

```
wparprnterr WparName
```

Description

The **wparprnterr** command prints all the error messages that are logged for a WPAR by using the **wparerr** command, **wpar_log_err**, and **kwpar_err** subroutines on the standard output.

Security

Attention RBAC users: This command can perform privileged operations. Only privileged users can run privileged operations. For more information about authorizations and privileges, see [Privileged Command Database in Security](#). For a list of privileges and the authorizations that are associated with this command, see the [lssecattr](#) command or the [getcmdattr](#) subcommand.

Examples

To display messages of the WPAR *mywpar*:

```
wparprnterr mywpar
```

write Command

Purpose

Opens a line of communication to send messages to other users on the system in real time.

Syntax

To query all messages awaiting replies from users on a host and display them with their handles, type the following:

write -q [**-n** *Host*]

To Reply to a Message Sent by a Utility or a Shell Script, or Redisplay the Message Associated with a Given handle, type the following:

write -h*Handle*, { ok | cancel | query } [**-n** *Host*]

To send messages to a user, optionally on another host or a particular device, type the following:

write [**-r**] { [**-n** *Host*] *User* | *User@Host* } [*Line*]

Description

The **write** command enables message sending over the system in real time. It provides conversation-like communication with another logged-in user. Each user alternately sends and receives short messages from the other workstation. Long messages can be sent by putting the complete message in a file and then redirecting that file as input to the **write** command.

For another user (as specified by the *User* parameter) to receive a message, that user must be logged in and must not have refused message permission. When a message is sent to a user who is not logged in, the message user not logged in appears. If the message is sent to a user who has refused message permission by setting the **mesg** command to no, the message write: permission denied appears.

When the **write** command is issued, it immediately sends the following message, along with an attention-getting sound (the ASCII BEL character) to the message recipient or target:

```
Message from SenderID on SenderHostname (ttynn) [Date] ...
```

With a successful connection, the **write** command sends two ASCII BEL characters to both workstations. The beep alerts the sender that the message can begin and it alerts the receiving user that a message is coming.

Sending occurs one line at a time as the Enter key is pressed. The communication link from the sender to the receiver remains open and sending continues until the Ctrl-D key sequence ends the sending link. Then an end-of-text character (<EOT>) is sent to the target workstation and the **write** command mode is terminated.

The receiving or target user can respond by sending a **write** command to the originating user. This opens a line of communication from the receiver back to the sender, enabling message responses in return. For this type of exchange, the following convention is useful: When you first write to others, wait for a response before sending any text. End a message with a signal such as o (over) to alert the other person to reply. Use oo (over and out) when the conversation is finished.

If the character ! (exclamation point) is found at the beginning of a line, the **write** command calls the shell to execute the rest of the line as a command.

When you write to a user who is logged in at more than one workstation or multi-using more than one process, the **write** command uses the first login instance found in the **/etc/utmp** file as the message delivery point (usually the login or console shell), and you get the message:

```
UserID is logged on more than one place.  
You are connected to "Workstation".  
Other locations are:  
Workstation
```

When this message is received, if you wish to send the message to a location other than the initial login location, the target user can be contacted at a different location by specifying the *Line* of the location (tty00, for example).

Permission to write to another user is granted or denied by the individual user with the **mesg** command. Some commands deny message permission while they are running to prevent interference with their output. A user with root user authority can write to any workstation regardless of the workstation's message permission.

You can use the **write** command to converse with users on other hosts. You can identify a user on a remote host by using the **-nHostName** flag or the *User@Host* parameter. In order to write to a user on a remote host, the **writesrv** daemon must be running on both the current host and the remote host.

The **write** command is also used by the **qdaemon** daemon to send messages to users on other hosts and to wait for replies. There are only three valid replies:

Item	Description
ok	The original write exits with a status of 0.
cancel	The original write exits with a status of 1.
query	The message associated with the given handle is displayed.

Parameters

Item	Description
<i>User</i>	Specifies the user ID of the person to receive the message text.
<i>User@Host</i>	Specifies the user ID and remote host of the person to receive the message text.
<i>Line</i>	Contacts the target user at another location (tty00, for example).

Flags

Item	Description
-h <i>Handle,Reply</i>	Replies to a message sent by a utility or shell script using write with the reply option. The value to be used for the <i>Handle</i> variable is generated internally and supplied to the user in the text of the original message. The reply can be ok, cancel, or query.
-nHost	Specifies a remote host. The <i>Host</i> variable may be a nickname or an internet address.
-q	Queries all messages awaiting replies from users on a host and displays them with their handles.
-r	Generates a message handle, places it in the message header, sends the message, and waits for a reply. This flag is used by the qdaemon daemon for operator messages and can be put in shell scripts. It is not used for interactive conversations. An exit status of 0 indicates that the reply was ok, a status of 1 indicates that the reply was cancel, and an exit status of 2 indicates that the user could not be contacted.

Requirements:

- The **writesrv** daemon must be running on the target host in order for any of the flags to work. If you are not using either the **-n** flag or *@Host*, but using **-h**, **-q**, or **-r**, the **writesrv** daemon must be running on your host.
- If TCP/IP is not installed on your machine but the *HostName* is set, in order to converse with users on the local host using the **write** command with the **-h**, **-q**, or **-r** flag, you must append your host name to the end of the loopback entry in the **/etc/hosts** file. The original entry should read:

```
127.0.0.1 loopback LocalHostName
```

The new entry should read:

```
127.0.0.1 loopback LocalHostName HostName
```

Exit Status

This command returns the following exit values:

Item	Description

0	Successful completion.
>0	The addressed user either is not logged on or denies permission.

Examples

1. To write a message to a user who is logged in, enter:

```
write june
```

Press the Enter key and type,

```
I need to see you! Meet me in the computer room at 12:30.
```

Then press the Ctrl-D key sequence to terminate the **write** command mode.

If your user ID is karen and you are using workstation tty3, june's workstation displays:

```
Message from karen on trek tty3 Aug 17 11:55:24 ...
I need to see you! Meet me in the computer room at 12:30.
<EOT>
```

2. To hold a conversation, enter:

```
write june
```

Press the Enter key and type,

```
Meet me in the computer room at 12:30.
o
```

This starts the conversation. The o at the beginning of the next line means the message is over. It tells June that you are waiting for a response. Do not press Ctrl-D if you wish to continue.

Now June replies by typing:

```
write karen
```

Presses the Enter key and types,

```
I'm running tests at 12:30. Can we meet at 3?
o
```

And you might respond:

```
OK--the computer room at 3.
oo
```

The oo means *over and out*, telling June that you have nothing more to say. If June is also finished oo, then you both press Ctrl-D to end the conversation.

3. To write someone a prepared message, enter:

```
write june < message.text
```

This writes the contents of the **message.text** file to june's workstation.

4. To write to the person using a certain workstation, enter:

```
write -n console
```

Press the Enter key and type,

```
The printer in building 998 has jammed.  
Please send help.
```

Then press the Ctrl-D key sequence.

This writes the message to the person logged in at the workstation /dev/console.

5. To send a message to user spuds at host partya, enter:

```
write -n partya spuds
```

Press the Enter key and type,

```
Your new tape has just arrived,  
come see me to pick it up.  
Thanks!
```

Then press the Ctrl-D key sequence.

OR

```
write spuds@party
```

Press the Enter key and type,

```
Your new tape has just arrived,  
come see me to pick it up.  
Thanks!
```

Then press the Ctrl-D key sequence.

6. Here is an example of a message sent by the **qdaemon** daemon:

```
Message from mary on trek (tty10) Aug 17 10:03:34 ...  
Use "write -h 6398492,reply" to reply  
Please insert tape number 5 into rmt0.  
<EOT>
```

To reply in the affirmative, enter:

```
write -h 6398492,ok
```

Then press the Ctrl-D key sequence.

To reply in the negative, enter:

```
write -h 6398492,cancel
```

Then press the Ctrl-D key sequence.

With the **-h** flag, there is no need to supply the host name or user ID. This information is tracked with the handle.

Files

Item	Description
<u>/etc/hosts</u>	Contains TCP/IP host information.
<u>/etc/utmp</u>	Contains user and accounting information for the <u>who</u> , write , and login commands.

writesrv Daemon

Purpose

Allows users to send messages to and receive messages from a remote system.

Syntax

writesrv

Description

The **writesrv** daemon allows users to send messages to users on a remote system and receive responses from users on a remote system with the **write** command.

The **writesrv** utility receives incoming requests from a **write** command and creates a server process to handle the request. This server process communicates with the client process (**write**) and provides whatever services are requested.

To perform these services, the **writesrv** daemon creates a socket that is attached to the port defined in the **/etc/services** file. All requests for service are sent as messages to this socket.

Note: If the **writesrv** daemon terminates abnormally (such as a system crash, power failure, or the **kill -9** command), the **/var/spool/writesrv** directory must be manually cleaned out to remove any files left behind by the **writesrv** daemon.

Examples

1. To start the **writesrv** daemon from the **/etc/rc** script, enter:

```
/usr/sbin/writesrv
```

The **writesrv** daemon is started from the **/etc/rc** script. This is the usual way the daemon is started.

2. To start the **writesrv** daemon using the System Resource Controller (SRC), enter:

```
startsrc -s writesrv &
```

The **writesrv** daemon is started using SRC.

Files

Item	Description
/etc/services	Contains the Network Services directory.

wtmpfix Command

Purpose

Manipulates connect-time accounting records by correcting date and time stamp inconsistencies.

Syntax

/usr/sbin/acct/wtmpfix [*File ...*]

Description

The **wtmpfix** command is called by the **runacct** procedure to examine standard input or *Files* that contain records in **wtmp** format, and correct problems that could make the **acctcon1** or **acctcon2** commands fail. The **wtmpfix** command corrects date and time stamp inconsistencies, and writes the corrected records to standard output. If the date and time stamps are not consistent when the **acctcon1** command runs, the **acctcon1** command generates an error and stops.

The **wtmpfix** command also checks the validity of the name field to ensure that it consists only of alphanumeric characters, a \$ (dollar sign), or spaces. If the name is invalid, the **wtmpfix** command changes the login name to **INVALID** and writes a diagnostic message to standard error. In this way, the **wtmpfix** command reduces the chance that the **acctcon2** command will fail.

Each time the date is set (on system startup or with the **date** command), a pair of date change records is written to the **/var/adm/wtmp** file. The first record is the old date, denoted by the *old time* string. The *old time* string is placed in the line field and the **OLD_TIME** flag is placed in the type field. The second record is the new date, denoted by the string *new time*. The *new time* string is placed in the line field and the **NEW_TIME** flag is placed in the type field. The **wtmpfix** command uses these records to synchronize all date and time stamps in the file.

Flags

None.

Parameters

Item	Description

<i>File</i>	Specifies the file to examine that contains records in wtmp format.

Security

Access Control: These commands should grant execute (x) access only to members of the **adm** group.

Examples

1. To correct date and time stamp inconsistencies in **/var/adm/wtmp** and write the corrected records to **dummy.file**, enter:

```
/usr/sbin/acct/wtmpfix /var/adm/wtmp > /tmp/dummy.file
```

Files

Item	Description
/usr/sbin/acct/wtmpfix	Contains the wtmpfix command.
/var/adm/wtmp	Contains records of date changes that include an old date and a new date.
/usr/include/utmp.h	Contains history records that include a reason, date, and time.

wump Command

Purpose

Starts the hunt the wumpus game.

Syntax

wump

Description

A wumpus is a creature living in a cave with many rooms interconnected by tunnels. You move among the rooms trying to shoot the wumpus with an arrow and trying to avoid being eaten by the wumpus or falling into bottomless pits. There are also super bats that may pick you up and drop you in some randomly selected room. For moving among the rooms and shooting arrows, the game prompts you with appropriate questions and follows your instructions. For example:

```
You are in room 14.  
I feel a draft.  
There are tunnels to 1 13 18.  
Move or shoot? (m-s) m  
Which room? 1  
You are in room 1.  
I feel a draft.  
There are tunnels to 14 17 18.  
Move or shoot? (m-s) m  
Which room? 17  
You are in room 17.  
You fell into a pit!  
Another game? (y-n)
```

In the above example, you start out in room 14. The computer displays `I feel a draft`. This is the hint that a pit is nearby. You choose to move to room 1. Again you are warned of the pit. You then choose to move to room 17 where you fall into a pit and die.

At the beginning of the game, you are prompted `Instructions? (y-n)`. Choosing `y` provides an explanation of the warnings, how to move, and how to shoot.

The game ends and you are prompted `Another game? (y-n)` if:

- You kill the wumpus.
- The wumpus eats you.
- You fall into a bottomless pit.
- You run out of arrows.

To quit the game at any time, press the interrupt (Ctrl-C) key sequence.

Files

Item	Description
<code>/usr/games</code>	Contains the location of the system's games.

X

The following AIX commands begin with the letter x.

X Command

Purpose

Starts the X Server.

Syntax

```
X [ -a Number ] [ -auth FileName ] [ -bc | +bc ] [ -bp Color ] [ -broadcast ] [ -bs | -nobs ] [ -c Volume ]  
[ -cc VisualType [ :Display ] ] [ -class DisplayClass ] [ -co File ] [ -cookie XDMAAuthenticationBit ]  
[ -D File ] [ -d Depth [ :Display ] ] [ -displayID DisplayID ] [ -damage | +damage ] [ -evie | +evie ]  
[ -f Number ] [ -fc Font ] [ -fixes | +fixes ] [ -fn Font ] [ -fp Font ] [ -help ] [ -I ] [ -indirect HostName ]  
[ -layer # [ :Display ] ] [ -logo | nologo ] [ -n :Number ] [ -once ] [ -P RowColumn Display } ] [ -pbuffer  
level [ :display name | :display number ] ] [ -p Number ] [ -port PortNumber ] [ -query HostName ] [ -r | r ]  
[ +render | -render ] [ -s Number ] [ -secIP [PermissionCode] ] [ -secLocal [PermissionCode] ] [ -secSMT  
[PermissionCode] ] [ -stereo [ :Display ] ] [ -su ] [ -T ] [ -t Number ] [ -to Number ] [ -v ] [ -vfb ] [ -wm ]  
[ -wp Color ] [ -wrap ] [ -wrapx ] [ -wrapy ] [ -x ExtensionName ] [ -xkbdir Directory ] [ -xkbmap  
FileName ] [ [+|-]accessx ] [ -ar1 Milliseconds ] [ -ar2 Milliseconds ] [ -sp FileName ] [ +/- xinerama  
FileName ]
```

Description

The **X** command starts the X Server, a display server that runs on bitmapped terminals. The X Server distributes input and output requests to or from programs located on either the host system or systems connected to it through a network.

End an Enhanced X-Windows session by using the Ctrl+Alt+Backspace key sequence.

You can specify one or more display devices. If none are specified, the default is all. The default configuration order is determined by the adapter slot order. The adapter in the first slot is initialized as the left most screen, the adapter in the second slot is the next screen to the right. To rearrange the layout of the screens, use the **-P** flag. The **-P** flag associates the row and column of the device with the device name. You can determine the device name by using the **lsdisp** command.

The two displays are arranged either vertically or horizontally. The following example shows **-P** flags specifying a horizontal arrangement:

```
-P11 ppr0 -P12 ppr1
```

The 2 in the right position of the second **-P** flag indicates that the second monitor view is along the x-axis. This produces the horizontal arrangement:

Display	1	Display	2

To see two monitors in a vertical arrangement, the **-P** flags should read:

```
-P11 ppr0 -P21 ppr1
```

The 2 in the first position indicates that the monitors are in a vertical configuration along the y-axis:

Display	1

In the horizontal configuration, when a mouse is traveling from left to right in Display 1 and reaches the border of Display 1 and 2, the cursor continues into Display 2 at the same y-axis position. When it reaches the edge of Display 2 and the **-wrapx** flag is set, it appears at the leftmost edge of Display 1 in the same y-axis position. If the **-wrapx** flag is not set, the mouse stops at the far edge of Display 2.

In a vertical configuration, when the mouse is traveling from top to bottom in Display 1 and reaches the border of Display 1 and Display 2, the cursor continues into Display 2 at the same x-axis position. When the cursor reaches the bottom of the display 2 and the **-wrapy** flag is set, the cursor appears at the top edge of Display 1 in the same x-axis position. If the **-wrapy** flag is not set, the mouse stops at the bottom of Display 2.

In addition, information and error messages (for example, a message indicating that an extension not able to load) are listed in the **/tmp/xlogfile** file. This file can provide useful information in cases when the X Server encounters a problem. This file is re-written every time the X Server is instantiated. This file provides additional error and non-error information but is not a complete error log for the X Server.

When X-Server is started, it comes up using the default color class. Depending on the driver, the X-Server may default to using the PseudoColor or TrueColor class.

The PseudoColor class uses a colormap to display the colors on the screen. Many graphic adapters only support one hardware colormap. In this case, if the default color class is PseudoColor and an application is using a colormap that is different from the default colormap, incorrect colors may be displayed on the screen. Only the window that has focus will display the correct colors. It is advisable to run the X-Server in the TrueColor mode to prevent wrong colors from being displayed on the screen.

The **-cc X-Server** command flag can be used to bring the X-Server up using the TrueColor class. The **/usr/lpp/X11/defaults/xserverrc** file can be modified to allow this as shown in the following example.

As a root user, edit the **/usr/lpp/X11/defaults/xserverrc** file. Update the **EXTENSIONS=""** variable as shown in the following example:

```
#-----
# Start the X server in True Color mode
#-----
EXTENSIONS="$EXTENSIONS -cc 4"
```

Restart X Server by logging out of CDE and clicking reset.

Note: The **xdpyinfo** command can be used to verify the default color class.

Flags

Item	Description
-a <i>Number</i>	Specifies the acceleration multiplier for mouse movement. For example, a value of 5 causes the cursor to move five times as fast as the mouse. The default is 4 pixels; any value specified must be a positive value greater than 0.
-auth <i>FileName</i>	Specifies to X the file from which to read the MIT (Massachusetts Institute of Technology) magic cookie.
-bc	Turns off backward compatibility with Enhanced X-Windows version 1.1.
+bc	Turns on backward compatibility with Enhanced X-Windows version 1.1. This is the default.

Item	Description														
-bp <i>Color</i>	Specifies a black pixel color for the display. The default is display dependent.														
-bs	Enables backing store support on all screens. Backing store support is disabled by default.														
-c <i>Volume</i>	Specifies key click volume.														
-cc <i>VisualType</i> [: <i>Display</i>]	Specifies the type of visual to use for the root window of the screen specified by the display name. Not all visual types are available on all adapters at all depths. The <i>:Display</i> parameter is optional, but useful when using the multihead option. The <i>:Display</i> parameter is the name of the display as shown in the lsdisp command. If no display number or name is supplied, the specified visual is selected for all screens. To specify the visual type and depth for the default visual, use the -cc and -d flags, respectively. Values for the <i>VisualType</i> parameter are specified as a string or a number as follows: <table> <tr> <th>String</th><th>Numeric equivalent</th></tr> <tr> <td>StaticGray</td><td>0</td></tr> <tr> <td>GrayScale</td><td>1</td></tr> <tr> <td>StaticColor</td><td>2</td></tr> <tr> <td>PseudoColor</td><td>3</td></tr> <tr> <td>TrueColor</td><td>4</td></tr> <tr> <td>DirectColor</td><td>5</td></tr> </table>	String	Numeric equivalent	StaticGray	0	GrayScale	1	StaticColor	2	PseudoColor	3	TrueColor	4	DirectColor	5
String	Numeric equivalent														
StaticGray	0														
GrayScale	1														
StaticColor	2														
PseudoColor	3														
TrueColor	4														
DirectColor	5														
-co <i>File</i>	Sets the name of the red, green, and blue (RGB) color database. This is the default flag for the color database.														
-D <i>File</i>	Specifies the full path name of the color definition database file. The default is /usr/lib/X11/rgb .														
-d <i>Depth</i> [: <i>Display</i>]	Specifies the root depth for the screen specified by the display name. Not all visual types will be available on all adapters at all depths. The <i>:Display</i> parameter is optional, but useful when using the multihead option and must correspond to the values passed with the -P flag. The <i>:Display</i> parameter is the name of the display as shown in the lsdisp command. In the absence of the <i>:Display</i> parameter, the specified depth is selected for all the selected displays in the multihead option, as specified in the -P flag.														
-damage	Disables the X Damage extension.														
+damage	Enables the X Damage extension.														
-evie	Disables the X Event Interception extension.														
+evie	Enables the X Event Interception extension.														

Item

-f *Number*

Description

Specifies the beep volume. The default is -1 or medium. The supported values are as follows:

Value	Setting
0	Off
1-33	Low
-1 or 34-66	Medium
67-100	High

-fc *Font*

Specifies the cursor font for cursor glyphs and cursor masks. The default depends on the operating system and the display.

-fixes

Disables the X Fixes extension.

+fixes

Enables the X Fixes extension.

-fn *Font*

Specifies the default text font. The default depends on the operating system and the display.

-fp *Font*

Specifies the font path.

-I

Causes all remaining command line arguments to be ignored. (Uppercase i)

-help

Prints a usage message.

-layer *#[:Display]*

Specifies that the default visual should be in the *#* layer. The *:Display* parameter is the name of the display as shown in the **lsdisp** command. Specifying this flag for an adapter that does not have overlays, or has less than 8 bits of overlay, has no effect. Specifying this flag with a *#* higher than the number of supported layers results in the default visual residing in the default layer of the screen (as if no **-layer** flag had been used).

-logo

Turns on the X Window System logo display in the screen saver. There is currently no way to change this from a client.

-n *:Number*

Specifies the connection number. Valid values for the *Number* parameter are 0 to 255. The default is the next available number. The *Number* parameter is used by programs to communicate with a specific X Server. For example, the command:

```
X -n :18
```

specifies that communication to the activated X Server takes place by *unix:18* or by *Hostname:18*.

-nobs

Disables backing store support on all screens. This is the default.

nologo

Turns off the X Window System logo display in the screen saver. There is currently no way to change this from a client.

-once

Instructs the server to exit after the first session ends. Normally, the server starts sessions automatically.

Item

-P*RowColumn Display*

Description

Specifies the physical positioning of the displays in a multihead configuration. The *Row* parameter indicates the row in which the display is located. The *Column* parameter indicates the column in which the display is located.

The *Display* parameter is the device name of the display as shown in the first column of output from the **lsdisp** command. The first **-P***RowColumn Display* occurrence on the command line describes screen 0 to the X Server, the second describes screen 1, and so on.

The **-P** flag is for use with multiple head support.

-pbuffer *level* [**:display** *name* | **:display** *number*]

Specifies the **pbuffer** memory allocation level for the screen specified by **:display**. This flag is only useful when used in conjunction with the GLX extension.

The *level* parameter indicates the relative amount of frame buffer memory to be reserved for pbuffers. Specified values must be in the range of [0..2]. A value of 0 indicates that no memory should be reserved for pbuffers. A value of 1 indicates that a low amount of memory should be reserved. A value of 2 indicates that a high amount of memory should be reserved. Not all adapters support pbuffers. For those that do, not all screen configurations support pbuffers. The actual amount of frame buffer memory reserved for pbuffers is device-dependent, and may be influenced by other factors, such as screen resolution or default pixel depth.

The **:display** parameter is optional, but useful when using the multihead option. The **:display** parameter is the name of the display as shown in the **lsdisp** command. If no display *number* or *name* is supplied, the specified **pbuffer** width is selected for all screens.

-p *Number*

Specifies the time interval, in minutes, between changes of the X Window System logo position. This flag is used with the **-s** (screen saver timeout) flag to control the blanking of the screen.

-r

Disables autorepeat. The default is autorepeat enabled.

r

Turns on autorepeat.

+render

Enables the X Render extension. By default, the X Render extension is disabled.

Note: X Render Extension can be activated on the X Server only with GXT135P, GXT145, GXT4500P, and GXT6500P graphics adapters. To check the available adapter on the system, run the **lsdisp** command. Use the **-vfb** flag with the virtual frame buffer along with the **+render** flag.

Item	Description
-render	Disables the X Render extension.
-s <i>Number</i>	Specifies the number of minutes to wait before blanking the screen. The default is 10 minutes. If this value is set to 0, the screen-saver is disabled.
-secIP [<i>PermissionCode</i>]	Sets local access control on the internet socket. The <i>PermissionCode</i> is 3 octal digits which can set read, write, and execute bits. If no <i>PermissionCode</i> is specified after a security flag, then permission is defaulted to 0 for that socket.
-secLocal [<i>PermissionCode</i>]	Sets access control on the unix socket. The <i>PermissionCode</i> is 3 octal digits which can set read, write, and execute bits. If no <i>PermissionCode</i> is specified after a security flag, then permission is defaulted to 0 for that socket.
-secSMT [<i>PermissionCode</i>]	Sets access control on the shared memory transport socket. The <i>PermissionCode</i> is 3 octal digits which can set read, write, and execute bits. If no <i>PermissionCode</i> is specified after a security flag, then permission is defaulted to 0 for that socket.
-stereo [: <i>Display</i>]	Configures the graphics adapter for optimum stereo support for the screen specified by <i>Display</i>. Supported screens will configure the adapter to provide the best available support for stereo. This may decrease other resources such as texture memory. The actual amount of memory affected is device-dependent, and may be influenced by other factors, such as screen resolution or default pixel depth. The <i>Display</i> parameter is optional, but useful when using the multihead option. The <i>Display</i> parameter is the name of the display as shown in the lsdisp command. If no display number or name is supplied, the -stereo flag pertains to all supported screens. Unsupported screens will ignore the -stereo flag.
-su	Disables save under support on all screens.
-T	Disables the Ctrl+Alt+Backspace key sequence that, by default, ends the AIXwindows session and all windows opened from it.
-t <i>Number</i>	Specifies the mouse threshold. The default is 2 pixels. Acceleration takes effect only if the mouse is moved beyond the mouse threshold in one time interval and only applies to the amount beyond the threshold.
-to <i>Number</i>	Specifies the number of minutes to elapse between connection checks. The default is 60 minutes. A specified value must be greater than 0.

Item	Description
-v	Specifies that the display be replaced with the current background color after the time specified by the -s flag expires. By default, if the -v flag is not used, the entire display is painted with the background tile after the time specified by the -s flag expires.
-vfb	Starts the X Server with Virtual Frame Buffer (VFB), without initializing any graphics adaptor.
-wm	Forces the default backing store of all windows to have the WhenMapped value. This is a convenient way of applying backing store to all windows.
-wp <i>Color</i>	Specifies a white pixel display color. The default depends on the display.
-wrap	Specifies the behavior of the mouse when its hotspot reaches the left or right border or the top or bottom of any root window. If this flag is set and the hotspot of the mouse reaches the left border of the leftmost root window, the mouse is automatically positioned at the right border of the rightmost root window at the same y position. Conversely, if this flag is set and the hotspot of the mouse reaches the right border of the rightmost root window, the mouse is automatically positioned at the left border of the leftmost root window at the same y position. If this flag is not set, the mouse stops at the left or right border of any root window. If this flag is set and the hotspot of the mouse reaches the top border of the topmost root window, the mouse is positioned at the bottom border of the bottommost root window at the same x position. Conversely, if this flag is set and the hotspot of the mouse reaches the bottom border of the bottommost root window, the mouse is positioned at the top border of the topmost root window at the same x position. The -wrap flag is for use with multiple head support.

Item	Description
-wrapx	Specifies the behavior of the mouse when its hotspot reaches the left or right border of any root window. If this flag is set and the hotspot of the mouse reaches the left border of the leftmost root window, the mouse is positioned at the right border of the rightmost root window at the same y position. Conversely, if this flag is set and the hotspot of the mouse reaches the right border of the rightmost root window, the mouse is positioned at the left border of the leftmost root window at the same y position. If this flag is not set, the mouse stops at the left or right border of any root window. The -wrapx flag is for use with multiple head support.
-wrapy	Specifies the behavior of the mouse when its hotspot reaches the top or bottom border of any root window. If this flag is set and the hotspot of the mouse reaches the top border of the topmost root window, the mouse is positioned at the bottom border of the bottommost root window at the same x position. Conversely, if this flag is set and the hotspot of the mouse reaches the bottom border of the bottommost root window, the mouse is positioned at the top border of the topmost root window at the same x position. If this flag is not set, the mouse stops at the top or bottom border of any root window. The -wrapy flag is for use with multiple head support.
-x <i>ExtensionName</i>	Specifies that the extension name should be loaded when the server is initialized. This is particularly useful for large extensions, such as the Display PostScript Level 2 (dps). This flag can be specified more than once with multiple extension names.
-query <i>HostName</i>	Enables Enhanced X-Windows Display Manager Control Protocol (XDMCP) and sends a Query packet to the specified host. The -query flag is for use with XDMCP.
-broadcast	Enables XDMCP and broadcasts BroadcastQuery packets to the network. The first responding display manager is chosen for the session. The -broadcast flag is for use with XDMCP.
-indirect <i>HostName</i>	Enables XDMCP and sends IndirectQuery packets to the specified host. The -indirect flag is for use with XDMCP.

Item	Description
-port <i>PortNumber</i>	Specifies an alternative port number for XDMCP . This flag must be specified before any -query , -broadcast , or -indirect flags. Normally, the server starts sessions one after another. This flag causes the server to exit after the first session ends. The -port flag is for use with XDMCP .
-class <i>DisplayClass</i>	Sets the value for an additional display qualifier used by XDMCP in resource lookup for display-specific options. The -class flag is for use with XDMCP .
-cookie <i>XDMAAuthenticationBits</i>	Specifies a private key to be shared between the server and the manager when testing XDM-AUTHENTICATION-1. The -cookie flag is for use with XDMCP .
-displayID <i>DisplayID</i>	Allows the display manager to identify each display so that it can locate the shared key specified by the -cookie flag. The -displayID flag is for use with XDMCP .
+/- xinerama	Enable/Disable panoramic screen or Virtual Large Screen (VLS). Allows users to treat all heads in a multihead environment as a large screen.

Xkeyboard Flags

Item	Description
-xkbdir <i>Directory</i>	Specifies the base directory for the keyboard layout files.
-xkbmap <i>FileName</i>	Specifies the keyboard description to load on startup.
[+ -]accessx	Enables (+) or disables (-) AccessX key sequences.
-ar1 <i>Milliseconds</i>	Sets the length of time in milliseconds that a key must be pressed before autorepeat starts.
-ar2 <i>Milliseconds</i>	Sets the length of time in milliseconds that should elapse between autorepeat generated keystrokes.

Security Extension Flags

Item	Description

-sp

FileName

Causes the server to attempt to read and interpret *FileName* as a security policy file with the format described below. The file is read at server startup and reread at each server reset.

The syntax of the security policy file is as follows. Notation: "*" means zero or more occurrences of the preceding element, and "+" means one or more occurrences. To interpret *foo/bar*, ignore the text after the /; it is used to distinguish between instances of *foo* in the next section.

```

policy file ::= version line other line*
version line ::= string/v '\n'
other line  ::= comment | access rule | site policy | blank line
comment    ::= # not newline* '\n'
blank line ::= space '\n'
site policy ::= sitepolicy string/sp '\n'
access rule ::= property property/ar window perms '\n'
property    ::= string
window      ::= any | root | required property
required property ::= property/rp | property with value
property with value ::= property/rpv = string/rv
perms       ::= [ operation | action | space ]*
operation   ::= r | w | d
action      ::= a | i | e
string      ::= dbl quoted string | single quoted string | unquoted string
dbl quoted string ::= space " not dqoute* " space
single quoted string ::= space ' not squote* ' space
unquoted string ::= space not space+ space
space       ::= [ ' ' | '\t' ]*
```

Character sets:

```

not newline ::= any character except '\n'
not dqoute  ::= any character except "
not squote  ::= any character except '
not space   ::= any character except those in space
```

Item	Description

The semantics associated with the previously described syntax are as follows.

version line

The first line in the file, specifies the file format version. If the server does not recognize the version *string/v*, it ignores the rest of the file. The version string for the file format described here is *version-1*.

Once past the *version line*, lines that do not match the above syntax are ignored.

comment

Lines are ignored.

sitepolicy

Lines are currently ignored. They are intended to specify the site policies used by the XC-QUERY-SECURITY-1 authorization method.

access rule

Lines specify how the server should react to untrusted client requests that affect the X Window property named *property/ar*. The rest of this section describes the interpretation of an *access rule*.

For an *access rule* to apply to a given instance of *property/ar*, *property/ar* must be on a window that is in the set of windows specified by *window*. If *window* is **any**, the rule applies to *property/ar* on any window. If *window* is **root**, the rule applies to *property/ar* only on root windows.

If *window* is *required property*, the following apply. If *required property* is a *property/rp*, the rule applies when the window also has that *property/rp*, regardless of its value. If *required property* is a *property with value*, *property/rpv* must also have the value specified by *string/rv*. In this case, the property must have type STRING and format 8, and should contain one or more null-terminated strings. If any of the strings match *string/rv*, the rule applies.

The definition of string matching is simple case-sensitive string comparison with one elaboration: the occurrence of the character '*' in *string/rv* is a wildcard meaning "any string." A *string/rv* can contain multiple wildcards anywhere in the string. For example, *x** matches strings that begin with **x**, **x* matches strings that end with **x**, **x** matches strings containing **x**, and *x*y** matches strings that start with **x** and subsequently contain **y**.

There may be multiple *access rule* lines for a given *property/ar*. The rules are tested in the order that they appear in the file. The first rule that applies is used.

Item**Description*****perms***

Specify operations that untrusted clients may attempt, and the actions that the server should take in response to those operations.

operation

Can be **r** (read), **w** (write), or **d** (delete). The following information shows how X Protocol property requests map to these operations in the X Consortium server implementation.

GetProperty

r, or **r** and **d** if delete = True

ChangeProperty

w

RotateProperties

r and **w**

DeleteProperty

d

ListProperties

none, untrusted clients can always list all properties

action

Can be **a** (allow), **i** (ignore), or **e** (error).

Allow

Executes the request as if it had been issued by a trusted client.

Ignore

Treats the request as a no-op. In the case of GetProperty, ignore means return an empty property value if the property exists, regardless of its actual value.

Error

Specifies not to execute the request and return a BadAtom error with the atom set to the property name. Error is the default action for all properties, including those not listed in the security policy file.

An *action* applies to all *operations* that follow it, until the next *action*> is encountered. Thus, **i****r****w****a****d** means ignore read and write, allow delete.

GetProperty and RotateProperties might do multiple operations (**r** and **d**, or **r** and **w**). If different actions apply to the operations, the most severe action is applied to the whole request; there is no partial request execution. The severity ordering is: allow < ignore < error. Thus, if the *perms* for a property are **i****r****e****d** (ignore read, error delete), and an untrusted client attempts GetProperty on that property with delete = True, an error is returned, but the property value is not. Similarly, if any of the properties in a RotateProperties do not allow both read and write, an error is returned without changing any property values.

Item	Description

An example a security policy file follows:

```
version-1

# Allow reading of application resources, but not writing.
property RESOURCE_MANAGER      root      ar iw
property SCREEN_RESOURCES      root      ar iw

# Ignore attempts to use cut buffers.  Giving errors causes apps to crash,
# and allowing access may give away too much information.
property CUT_BUFFER0           root      irw
property CUT_BUFFER1           root      irw
property CUT_BUFFER2           root      irw
property CUT_BUFFER3           root      irw
property CUT_BUFFER4           root      irw
property CUT_BUFFER5           root      irw
property CUT_BUFFER6           root      irw
property CUT_BUFFER7           root      irw

# If you are using Motif, you probably want these.

property _MOTIF_DEFAULT_BINDINGS      rootar iw
property _MOTIF_DRAG_WINDOW            root  ar iw
property _MOTIF_DRAG_TARGETS           any   ar iw
property _MOTIF_DRAG_ATOMS             any   ar iw
property _MOTIF_DRAG_ATOM_PAIRS        any   ar iw

# The next two rules let xwininfo -tree work when untrusted.
property WM_NAME                       any     ar

# Allow read of WM_CLASS, but only for windows with WM_NAME.
# This might be more restrictive than necessary, but demonstrates
# the required property facility, and is also an attempt to
# say "top level windows only."
property WM_CLASS                      WM_NAME  ar

# These next three let xlsclients work untrusted.  Think carefully
# before including these; giving away the client machine name and command
# may be exposing too much.
property WM_STATE                      WM_NAME  ar
property WM_CLIENT_MACHINE             WM_NAME  ar
property WM_COMMAND                    WM_NAME  ar

# To let untrusted clients use the standard colormaps created by
# xstdcmap, include these lines.
property RGB_DEFAULT_MAP               root     ar
property RGB_BEST_MAP                  root     ar
property RGB_RED_MAP                   root     ar
property RGB_GREEN_MAP                 root     ar
property RGB_BLUE_MAP                  root     ar
property RGB_GRAY_MAP                  root     ar

# To let untrusted clients use the color management database created
# by xcmsdb, include these lines.
property XDCCC_LINEAR_RGB_CORRECTION   rootar
property XDCCC_LINEAR_RGB_MATRICES     rootar
property XDCCC_GRAY_SCREENWHITEPOINT   rootar
property XDCCC_GRAY_CORRECTION         rootar

# oddball property names and explicit specification of error conditions
property "property with spaces"        'property with " 'aw er ed

# Allow deletion of Woo-Hoo if window also has property OhBoy with value
# ending in "son".  Reads and writes will cause an error.
property Woo-Hoo                       OhBoy = "*son"ad
```

Example

To start the X Server with X Render extension, enter the following command:

```
$X -T -force :0 -vfb -d 32 +render
```

In this example, the X Server will use the Virtual Frame Buffer (VFB) for rendering instead of using the physical graphics adaptor.

x_add_fs_fpe Command

Purpose

Adds a network font server to a font path.

Syntax

x_add_fs_fpe *Host Port Position TypeName*

Description

The **x_add_fs_fpe** command adds a font path element to the font path of the selected network type name for a font server to access fonts.

Item	Description
<i>Host</i>	Specifies the name of the system where the font server resides.
<i>Port</i>	Specifies the number of the font server port. This number must be in the /etc/services file and specified in decimal.
<i>Position</i>	Specifies where to insert this element in the font path.
<i>TypeName</i>	Specifies the name of the network type. Each network type has a font path consisting of one or more font path elements. Specify the name of the network type to which the font path element will be added, or choose to have it added to all network type names by specifying All . If a font path element is added to All network types, will be placed at the end of each font path.

Security

Access Control: Only the root user should have execute (x) access to this command.

Example

To add the font server to the start of the font path for network type **x_st_mgr.ether**, enter:

```
x_add_fs_fpe winter 7500 1 x_st_mgr.ether
```

In this example, the font server on host **winter** has been added to the start of the font path for network type **x_st_mgr.ether**. The font server port is 7500.

Files

Item	Description
/usr/lpp/x_st_mgr/bin/x_add_fs_fpe	Contains the x_add_fs_fpe command.
/etc/x_st_mgr/ether.cf	Contains the network type x_st_mgr.ether configuration file (sample).

x_add_nfs_fpe Command

Purpose

Adds a NFS/TFTP accessed font directory to a font path.

Syntax

x_add_nfs_fpe *Host Directory Method Position TypeName*

Description

The **x_add_nfs_fpe** command adds a font path element to the font path of the selected network type name. This font directory will be accessed using Network File System (NFS) or Trivial File Transfer Protocol (TFTP).

Item	Description
<i>Host</i>	Specifies the system name to access for the font directory.
<i>Directory</i>	Specifies the complete path to the directory that contains the fonts.
<i>Method</i>	Specifies either <code>nfs</code> or <code>tftp</code> to be used to access the fonts.
<i>Position</i>	Specifies where to insert this element in the font path.
<i>TypeName</i>	Specifies the name of the network type. Each network type has a font path consisting of one or more font path elements. Specify the name of the network type to which the font path element will be added, or choose to have it added to all network type names by specifying <code>All</code> . If a font path element is added to <code>All</code> network types, it will be placed at the end of each font path.

Security

Access Control: Only the root user should have execute (x) access to this command.

Example

To add the fonts in `/usr/lib/X11/fonts/100dpi` to the network type `x_st_mgr.ether`, enter:

```
x_add_nfs_fpe cedar /usr/lib/X11/fonts/100dpi nfs Last \ x_st_mgr.ether
```

In this the font path element `/usr/lib/X11/fonts/100dpi` is added to the end of the font path for network type `x_st_mgr.ether`. The font directory is on the host `cedar`, which is accessed using NFS.

Files

Item	Description
<code>/usr/lpp/x_st_mgr/bin/x_add_nfs_fpe</code>	Contains the x_add_nfs_fpe command.
<code>/etc/x_st_mgr/ether.cf</code>	Contains the network type x_st_mgr.ether configuration file (sample).

x_rm_fpe Command

Purpose

Removes a font path element from a font path.

Syntax

x_rm_fpe *TypeName Position Method Host Port Directory*

Description

The **x_rm_fpe** command removes a font path element from the font path of the selected network type name.

Item	Description
<i>TypeName</i>	Specifies from which network type name the element is to be removed.
<i>Position</i>	Specifies where the element is in the font path.
<i>Method</i>	Specifies the method used to access the font path element. The valid options are: tcp for Network Font Server; default for initial default font path element; nfs for NFS; and tftp for TFTP.
<i>Host</i>	Specifies the name of the system specified in the font path element. For elements using the default method, specify None .
<i>Port</i>	Specifies the number of the server port specified in the font path element. For elements using the nfs or tftp method, specify None .
<i>Directory</i>	Specifies the complete path to the directory that contains the fonts. For a Network Font Server element, specify None .

Security

Access Control: Only the root user should have execute (x) access to this command.

Examples

To remove the font element `/usr/lib/X11/fonts/100dpi` from the font path for network type `x_st_mgr.ether`, enter:

```
x_rm_fpe x_st_mgr.ether 3 nfs waco None /usr/lib/X11/fonts/100dpi
```

In this example, the font path element `/usr/lib/X11/fonts/100dpi` that is accessed on host `waco` using NFS has been removed from the third position of the font path for network type `x_st_mgr.ether`. Because a port number is not used for NFS, this parameter was set to `None`.

Files

Item	Description
<code>/usr/lpp/x_st_mgr/bin/x_rm_fpe</code>	Contains the x_rm_fpe command.
<code>/etc/x_st_mgr/ether.cf</code>	Contains the network type x_st_mgr.ether configuration file (sample).

xargs Command

Purpose

Constructs parameter lists and runs commands.

Syntax

xargs [-p t x] [-e *EOFString*] [-E *EOFString*] [-i *ReplaceString*] [-I *ReplaceString*] [-L *Number*] [-n *Number*] [-l [*Number*]] [-s *Size*] [*Command* [*Argument* ...]]

Note: Do not put a blank space between the lowercase flags and the parameter.

Description

The generated command line length is the sum of the size, in bytes, of the *Command* and each *Argument* treated as strings, including a null byte terminator for each of these strings. The **xargs** command limits the command line length. When the constructed command line runs, the combined *Argument* and environment lists can not exceed **ARG_MAX** bytes. Within this constraint, if you do not specify the **-n** or the **-s** flags, the default command line length is at least the value specified by **LINE_MAX**.

Flags

Item	Description
-e[<i>EOFString</i>]	Obsolete flag. Use the -E flag. Uses the <i>EOFString</i> parameter as the logical EOF string. If you do not specify the -e or the -E flags, underscore (_) is assumed for the logical EOF string. If you do not specify the <i>EOFString</i> parameter, the logical EOF string capability is disabled, and underscores are taken literally. The xargs command reads from standard input until either EOF or the specified string is reached.
-E <i>EOFString</i>	Specifies a logical EOF string to replace the default underscore (_). The xargs command reads standard input until either EOF or the specified string is reached.
-i [<i>ReplaceString</i>]	Obsolete flag. Use the -I (uppercase i) flag. If you do not specify the <i>ReplaceString</i> parameter, the string {} is used. Note: The -I (uppercase i), i , -L (uppercase l), l , and -n flags are mutually exclusive. The last flag specified takes effect.
-I <i>ReplaceString</i>	(Uppercase i). Inserts each line of standard input as an argument for the <i>Command</i> parameter, inserting it in <i>Argument</i> for each occurrence of <i>ReplaceString</i> . <i>ReplaceStrings</i> can not be used in more than 5 arguments. Blank characters at the beginning of each standard input line are ignored. Each <i>Argument</i> can contain one or more <i>ReplaceStrings</i> , but may not be larger than 255 bytes. The -I flag also turns on the -x flag. The -I (uppercase i) flag means -L1 . Therefore, only one standard input line can be substituted as an argument at a time. If the replaced string appears more than once in the command parameter, the same standard input line is substituted for each occurrence of the replaced string. Note: The -I (uppercase i), i , -L (uppercase l), l , and -n flags are mutually exclusive. The last flag specified takes effect.
-l [<i>Number</i>]	(Lowercase l). Obsolete flag. Use the -L flag. If you do not specify the <i>Number</i> parameter, a value of 1 is used. The -l flag also turns on the -x flag. Note: The -I (uppercase i), i , -L (uppercase l), -l , and -n flags are mutually exclusive. The last flag specified takes effect.

Item	Description
-L <i>Number</i>	Runs the <i>Command</i> parameter with the specified number of nonempty parameter lines read from standard input. The last invocation of the <i>Command</i> parameter can have fewer parameter lines if fewer than the specified <i>Number</i> remain. A line ends with the first new-line character unless the last character of the line is a space or a tab. A trailing space indicates a continuation through the next nonempty line. Reads the <i>Number</i> lines from the standard input and places them at the end of the command line. Note: The -I (uppercase i), i, -L (uppercase l), -l, and -n flags are mutually exclusive. The last flag specified takes effect.
-n <i>Number</i>	Runs the <i>Command</i> parameter using as many standard input arguments as possible, up to the maximum specified by the <i>Number</i> parameter. Reads the maximum of <i>Number</i> arguments from the standard input and places them at the end of the command line. The xargs command uses fewer arguments if: • The accumulated command line length exceeds the bytes specified by the -s <i>Size</i> flag. • The last iteration has fewer value than the value specified by the <i>Number</i> argument, but not zero, arguments remaining. Note: The -I (uppercase i), i, -L (uppercase l), -l, and -n flags are mutually exclusive. The last flag specified takes effect.
-p	Asks whether to run the <i>Command</i> parameter. It displays the constructed command line, followed by a ? . . . (question mark, ellipsis) prompt. Enter an affirmative response specific to the locale to run the <i>Command</i> parameter. Any other response causes the xargs command to skip that particular invocation of the parameter. You are asked about each invocation. The -p flag also turns on the -t flag.
-s <i>Size</i>	Sets the maximum total size of the constructed <i>Command</i> line. The <i>Size</i> parameter must be a positive integer. Fewer arguments are used if: 1. The total number of arguments exceeds those specified by the -n flag. 2. The total number of lines exceeds those specified by the -L or -l (Lowercase l) flags. 3. EOF is reached before the number of bytes specified by the <i>Size</i> parameter are accumulated.
-t	Enables the trace mode and echoes the constructed <i>Command</i> line to standard error before running.
-x	Stops running the xargs command if any <i>Command</i> line is greater than the number of bytes specified by the -s <i>Size</i> flag. This -x flag is turned on if you specify either the -I (Uppercase i) or -l (Lowercase l) flag. If you do not specify the -i, -I (Uppercase i), -l (Lowercase l), -L, or -n flag, the total length of the <i>Command</i> line must be within the limit specified by the -s <i>Size</i> flag.

Exit Status

This command returns the following exit values:

Item	Description
0	All invocations of the <i>Command</i> parameter returned exit status 0.
1-125	A command line meeting the specified requirements could not be assembled, one or more of the invocations of the <i>Command</i> parameter returned a non-zero exit status, or some other error occurred.
126	<i>Command</i> was found but could not be invoked.
127	<i>Command</i> could not be found.

If a command line meeting the specified requirements cannot be assembled, the command cannot be invoked, an invocation of the command is terminated by a signal, or an invocation of the command exits with exit status 255. The **xargs** command will write a diagnostic message and exit without processing any remaining input.

Examples

1. To use a command on files whose names are listed in a file, type:

```
xargs lint -a <cfiles
```

If the *cfiles* file contains the following text:

```
main.c readit.c
gettoken.c
putobj.c
```

the **xargs** command constructs and runs the following command:

```
lint -a main.c readit.c gettoken.c putobj.c
```

If the *cfiles* file contains more file names than fit on a single shell command line (up to **LINE_MAX**), the **xargs** command runs the **lint** command with the file names that fit. It then constructs and runs another **lint** command using the remaining file names. Depending on the names listed in the *cfiles* file, the commands might look like the following:

```
lint -a main.c readit.c gettoken.c . . .
lint -a getisx.c getprp.c getpid.c . . .
lint -a fltadd.c fltmult.c fltdiv.c . . .
```

This command sequence is not quite the same as running the **lint** command once with all the file names. The **lint** command checks cross-references between files. However, in this example, it cannot check between the *main.c* and the *fltadd.c* files, or between any two files listed on separate command lines.

For this reason you may want to run the command only if all the file names fit on one line. To specify this to the **xargs** command use the **-x** flag by typing:

```
xargs -x lint -a <cfiles
```

If all the file names in the *cfiles* file do not fit on one command line, the **xargs** command displays an error message.

2. To construct commands that contain a certain number of file names, type:

```
xargs -t -n 2 diff <<EOF
starting chap1 concepts chap2 writing
chap3
EOF
```

This command sequence constructs and runs **diff** commands that contain two file names each (**-n 2**):

```
diff starting chap1
diff concepts chap2
diff writing chap3
```

The **-t** flag causes the **xargs** command to display each command before running it, so you can see what is happening. The <<EOF and EOF pattern-matching characters define a [here document](#), which uses the text entered before the end line as standard input for the **xargs** command.

3. To insert file names into the middle of command lines, type:

```
ls | xargs -t -I {} mv {} {}.old
```

This command sequence renames all files in the current directory by adding `.old` to the end of each name. The **-I** flag tells the **xargs** command to insert each line of the **ls** directory listing where {} (braces) appear. If the current directory contains the files `chap1`, `chap2`, and `chap3`, this constructs the following commands:

```
mv chap1 chap1.old
mv chap2 chap2.old
mv chap3 chap3.old
```

4. To run a command on files that you select individually, type:

```
ls | xargs -p -n 1 ar r lib.a
```

This command sequence allows you to select files to add to the `lib.a` library. The **-p** flag tells the **xargs** command to display each **ar** command it constructs and to ask if you want to run it. Type `y` to run the command. Press the any other key if you do not want to run the command.

Something similar to the following displays:

```
ar r lib.a chap1 ?...
ar r lib.a chap2 ?...
ar r lib.a chap3 ?...
```

5. To construct a command that contains a specific number of arguments and to insert those arguments into the middle of a command line, type:

```
ls | xargs -n6 | xargs -I{} echo {} - some files in the directory
```

If the current directory contains files `chap1` through `chap10`, the output constructed will be the following:

```
chap1 chap2 chap3 chap4 chap5 chap6 - some files in the directory
chap7 chap8 chap9 chap10 - some files in the directory
```

File

Item	Description
<code>/usr/bin/xargs</code>	Contains the xargs command.

xauth Command

Purpose

Edits and displays the authorization information used in connecting to the X server.

Syntax

```
xauth [ -f AuthFile ] [ -v | -q ] [ -i ] [ -b ] [ CommandArgument ... ]
```

Description

The **xauth** command is usually used to edit and display the authorization information used in connecting to the X server. This program extracts authorization records from one machine and merge them into another (for example, when using remote logins or granting access to other users).

The following commands can be entered interactively, on the **xauth** command line, or in scripts. Note that this program does not contact the X server.

Item

add *DisplayName ProtocolName Hexkey*

Description

An authorization entry is added to the authorization file for the indicated display using the given protocol and key data. The data is specified as an even-length string of hexadecimal digits, each pair representing one octet. The first digit of each pair gives the most significant 4 bits of the octet, and the second digit of the pair gives the least significant 4 bits. For example, a 32-character hexkey would represent a 128-bit value. A protocol name consisting of just a single period is treated as an abbreviation for **MIT-MAGIC-COOKIE-1**.

extract *FileName DisplayName...*

Authorization entries for each of the specified displays are written to the indicated file. The extracted entries can be read back in using the **merge** and **nmerge** commands. If the file name consists of just a single dash, the entries are written to the binary output.

Item

generate *DisplayName ProtocolName* [*trusted | untrusted*] [*timeout seconds*] [*group group-id*] [*data hexdata*]

Description

This command is similar to **add**. The main difference is that instead of requiring the user to supply the key data, it connects to the server specified in *displayname* and uses the SECURITY extension in order to get the key data to store in the authorization file. If the server cannot be contacted or if it does not support the SECURITY extension, the command fails. Otherwise, an authorization entry for the indicated display using the given protocol is added to the authorization file. A protocol name consisting of just a single period is treated as an abbreviation for MIT-MAGIC-COOKIE-1.

If the *trusted* option is used, clients that connect using this authorization will have full run of the display, as usual. If *untrusted* is used, clients that connect using this authorization will be considered untrusted and prevented from stealing or tampering with data belonging to trusted clients. See the SECURITY extension specification for full details on the restrictions imposed on untrusted clients. The default is *untrusted*.

The *timeout* option specifies how long in seconds this authorization will be valid. If the authorization remains unused (no clients are connected with it) for longer than this time period, the server purges the authorization, and future attempts to connect using it will fail. Note that the purging done by the server does not delete the authorization entry from the authorization file. The default timeout is 60 seconds.

The *group* option specifies the application group that clients connecting with this authorization should belong to. See the application group extension specification for more details. The default is to not belong to an application group.

The *data* option specifies data that the server should use to generate the authorization. Note that this is not the same data that gets written to the authorization file. The interpretation of this data depends on the authorization protocol. The *hexdata* is in the same format as the *hexkey* described in the **add** command. The default is to send no data.

list [*DisplayName...*]

Authorization entries for each of the specified displays (or all displays if none are named) are printed on the standard output in a textual format. Key data is always displayed in the hexadecimal format given in the description of the **add** command.

Item	Description
merge [<i>FileName...</i>]	Authorization entries are read from the specified files and are merged into the authorization database, superseding any matching existing entries. If a file name consists of just a single dash, the binary input is read if it has not been read before.
[n]extract <i>Filename DisplayName...</i>	Authorization entries for each of the specified displays are written to the indicated file. The entries are written in a numeric format suitable for non-binary transmission (such as secure electronic mail). The extracted entries can be read back in using the merge and nmerge commands. If the file name consists of just a single dash, the entries are written to the standard output.
[n]list [<i>DisplayName...</i>]	Authorization entries for each of the specified displays (or all displays if none are named) are printed on the standard output in the numeric format used by the nextract command. Key data is always displayed in the hexadecimal format given in the description of the add command.
[n]merge [<i>FileName...</i>]	Authorization entries are read from the specified files and are merged into the authorization database, superseding any matching existing entries. The numeric format given in the description of the extract command is used. If a file name consists of just a single dash, the standard input is read if it has not been read before.
remove <i>DisplayName...</i>	Authorization entries matching the specified displays are removed from the authority file.
source <i>FileName</i>	The specified file is treated as a script containing xauth commands to execute. Blank lines and lines beginning with a # (pound sign) are ignored. A single dash can be used to indicate the standard input, if it has not already been read.
info	Information describing the authorization file, whether or not any changes have been made, and from where xauth commands are being read is printed on the standard output.
exit	If any modifications have been made, the authority file is written out (if allowed), and the program exits. An end of file is treated as an implicit exit command.
quit	The program exits, ignoring any modifications. This may also be accomplished by pressing the interrupt character.
help [<i>String</i>]	A description of all commands that begin with the given string (or all commands if no string is given) is printed on the standard output.
?	A short list of the valid commands is printed on the standard output.

Display names for the **add**, **[n]extract**, **[n]list**, **[n]merge**, and **remove** commands use the same format as the **DISPLAY** environment variable and the common *display* command-line argument. Display-specific information (such as the screen number) is unnecessary and is ignored. Same-machine connections (such as local-host sockets, shared memory, and the Internet Protocol *HostName LocalHost*) are referred to as *HostName/unix:DisplayNumber* so that local entries for different machines can be stored in one authority file.

Note: Users that have unsecure networks should take care to use encrypted file transfer mechanisms to copy authorization entries between machines. Similarly, the MIT-MAGIC-COOKIE-1 protocol is not very useful in unsecure environments. Sites that are interested in additional security may need to use encrypted authorization mechanisms such as Kerberos. Spaces are currently not allowed in the protocol name. Quoting could be added.

Flags

The following options are used with the **xauth** command. They can be given individually (for example, **-q** **-i**) or combined (for example, **-qi**).

Item	Description
-f <i>AuthFile</i>	Specifies the name of the authority file to use. By default, xauth uses the file specified by the XAUTHORITY environment variable or <i>.xauthority</i> in the user's home directory.
-v	Indicates that xauth should operate verbosely and print status messages indicating the results of various operations (for example, how many records have been read in or written out). This is the default if xauth is reading commands from its standard input and its standard output is directed to a terminal.
-q	Indicates that xauth should operate quietly and not print unsolicited status messages. This is the default if an xauth command is given on the command line or if the standard output is not directed to a terminal.
-i	Indicates that xauth should ignore any authority file locks. Normally, xauth refuses to read or edit any authority files that have been locked by other programs (usually xdm or another xauth).
-b	Indicates that xauth should attempt to break any authority file locks before proceeding. Use this option only to clean up stale locks.

Example

The most common use for the **xauth** command is to extract the entry for the current display, copy it to another machine, and merge it into the user's authority file on the remote machine:

```
% xauth extract \- $DISPLAY | rsh otherhost xauth merge \-
```

Files

Item	Description
\$HOME/.Xauthority	Contains the default authority file if the XAUTHORITY environment variable is not defined.

xclock Command

Purpose

Continuously displays the current time of day.

Syntax

```
xclock [ -Xtoolkitoption ... ] [ -analog | -digital ] [ -chime ] [ -hd Color ] [ -help ] [ -hl Color ]  
[ -padding Number ] [ -update Seconds ]
```

Description

The **xclock** command gets the time from the system clock, then displays and updates it in the form of a digital or analog clock. Select the **-analog** or **-digital** flag to display the clock in analog or digital formats. You can also select flags to specify the presentation of the clock, including chime and update frequency, colors, and border width.

This command uses the Athena clock widget, which understands core resource names and classes. To specify these resources, you need to know the hierarchy of the widgets that comprise the **xclock** command. In the following example, the indented items indicate the hierarchical structure. The widget class name is given first, followed by the widget instance name:

```
XClock xclock  
Clock clock
```

The following examples demonstrate the possible ways to specify resources for this client:

```
xclock.clock.background
```

```
XClock*background
```

```
xclock*background
```

Note: Specifying resources as `xclock.background` which worked with the previous version of `xclock` will not work with this version.

Flags

Item	Description
<i>-Xtoolkitoption</i>	The xclock command accepts all of the standard X Toolkit command-line option flags in addition to the specific flags listed. You can view standard X Toolkit command-line option flag the in the custom command.
-analog	Sets the analog display mode, which is the default mode. Draws a conventional 12-hour clock face with ticks for each minute and stroke marks on each hour.
-chime	Specifies the sounding of a chime once on the half hour and twice on the hour.
-digital	Sets the 24-hour digital display mode. Displays the date and time in digital form.
-hd <i>Color</i>	Specifies the color of the hands in analog mode on color displays. The default is black.
-help	Prints a brief summary of the allowed options.
-hl <i>Color</i>	(lowercase HL) Specifies the highlight color of the edges of the hands of the analog clock. The default is black.
-padding <i>Number</i>	Specifies the width in pixels of the padding between the window border and the clock text or picture. The default is 8.

Item	Description
-update <i>Seconds</i>	Specifies the frequency in seconds that the xclock command updates its display. If the xclock window is obscured and then exposed, the xclock command redisplay immediately. The specification of an update frequency less than 30 seconds enables the second hand in the analog mode. The default update frequency is 60 seconds.

.Xdefaults Keywords

Use the following keywords to set the defaults for the **xclock** command.

Item	Description
analog (class Boolean)	Specifies an analog clock instead of a digital clock. The default is true.
chime (class Boolean)	Specifies whether a bell sounds on the hour and half hour.
fontSet (class FontSet)	Specifies the fontset for the digital clock. Variable-width fonts do not always display correctly.
foreground (class Foreground)	Specifies the color of tick marks on color displays. If reverseVideo is specified, the default is white, otherwise the default is black.
hands (class Foreground)	Specifies the color on the inside of the hands in the analog clock on color displays. If reverseVideo is specified, the default is white, otherwise the default is black.
highlight (class Foreground)	Specifies the color used to highlight the clock's hands. If reverseVideo is specified, the default is white, otherwise the default is black.
height (class Height)	Specifies the height of the clock. The default for the analog clock is 164 pixels. The default for the digital clock is whatever is required to hold the clock when displayed in the chosen font.
padding (class Margin)	Specifies the amount of internal padding in pixels. The default is 8.
update (class Interval)	Specifies the frequency in seconds in which the xclock command updates its display.
width (class Width)	Specifies the width of the clock. The default for the analog clock is 164 pixels. The default for the digital clock is whatever is needed to hold the clock when displayed in the chosen font.

Environment Variables

Item	Description
DISPLAY	Gets the default host and display number.
XENVIRONMENT	Gets the name of a resource file that overrides the global resources stored in the RESOURCE_MANAGER property.

Examples

1. To specify a digital clock display, enter:

```
xclock -digital
```

2. To specify red hands on an analog clock, enter:

```
xclock -hd red
```

File

Item	Description
<code>/usr/lib/X11/app-defaults/XClock</code>	Specifies the required resources.

xcmsdb Command

Purpose

Loads, queries, or removes Screen Color Characterization Data stored in properties on the root window of the screen.

Note: The **xcmsdb** command is only supported in X11R5 (AIXwindows Version 1.2.3).

Syntax

```
xcmsdb [ -display Display ] [ [ -query ] [ -remove ] [ -color ] ] [ -format 32 | 16 | 8 ] [ FileName ]
```

Description

The **xcmsdb** command is used to load, query, or remove Screen Color Characterization Data stored in properties on the root window of the screen. Screen Color Characterization Data is an integral part of **Xlib**, which is necessary for proper conversion between device-independent and device-dependent color specifications. **Xlib** uses the **XDCCC_LINEAR_RGB_MATRICES** and **XDCCC_LINEAR_RGB_CORRECTION** properties to store color characterization data for color monitors. It uses **XDCCC_GRAY_SCREENWHITEPOINT** and **XDCCC_GRAY_CORRECTION** properties for gray scale monitors. Because **Xlib** allows the addition of Screen Color Characterization Function Sets, added function sets may place their Screen Color Characterization Data on other properties. This utility is unaware of these other properties; therefore, you will need to use a similar utility provided with the function set, or use the example **xprop** utility.

The ASCII readable contents of the *FileName* parameter (or the standard input if no input file is given) are appropriately transformed for storage in properties, provided the **-query** or **-remove** flag options are not specified.

Note: The Xcms API in **libX11.a** is supported; however, the client side color name data base, **/usr/lib/X11/Xcms.txt**, and a device color characterization file, **/usr/lib/X11/XcmsIBM5081.dcc**, are provided as unsupported samples.

Flags

Item	Description
-display <i>Display</i>	Specifies the server to which you are converting.
-query	Reads or attempts to read the XDCCC properties off the screen's root window. If successful, it transforms the data into a more readable format, and then sends the data to standard output.

Item	Description
-remove	Removes or attempts to remove the XDCCC properties on the screen's root window.
-color	Sets the -query and -remove options to only check for the XDCCC_LINEAR_RGB_MATRICES and XDCCC_LINEAR_RGB_CORRECTION properties. If the -color option is not set, the -query and -remove options check for all the properties.
-format 32 16 8	Specifies the property format (32, 16, or 8 bits per entry) for the XDCCC_LINEAR_RGB_CORRECTION property. Precision of encoded floating-point values increases with the increase in bits per entry. The default is 32 bits per entry.

Parameter

Item	Description
<i>FileName</i>	Specifies the ASCII readable contents of a Screen Color Characterization Data file.

Examples

1. Use the following example to put Screen Color Characterization Data on the root window by telling the **xcmsdb** command to read it from a file:

```
xcmsdb /usr/lib/X11/XcmsIBM5081.dcc
```

2. Use the following example after you have already put Screen Color Characterization Data on the root window to tell the **xcmsdb** command to read the data back if it exists:

```
xcmsdb -query
```

xdm Command

Purpose

Manages a collection of X Displays with support for XDMCP.

Syntax

```
xdm [ -config ConfigurationFile ] [ -debug DebugLevel ] [ -nodaemon ] [ -error ErrorLogFile ]
[ -resources ResourceFile ] [ -server ServerEntry ] [ -udpPort PortNumber ] [ -session SessionProgram ]
[ -xrm ResourceSpecification ]
```

Description

The **xdm** (X Display Manager) command manages a collection of X displays, which may be on the local host or remote servers. The design of the **xdm** command was guided by the needs of X terminals as well as the X Consortium standard XDMCP, the *X Display Manager Control Protocol*. The **xdm** command provides services similar to those provided by the **init**, **getty**, and **login** commands on character terminals: prompting for login name and password, authenticating the user, and running a session.

A *session* is defined by the lifetime of a particular process; in the traditional character-based terminal world, it is the user's login shell. In the **xdm** context, it is an arbitrary session manager. This is because in a windowing environment, a user's login shell process does not necessarily have any terminal-like interface with which to connect. When a real session manager is not available, a window manager or terminal emulator is typically used as the *session manager*, meaning that ending this process ends the user's session.

When the session is ended, **xdm** resets the X server and (optionally) restarts the whole process.

When the **xdm** command receives an **Indirect** query by way of XDMCP, it can run a **chooser** process to perform an XDMCP **BroadcastQuery** (or an XDMCP Query to specified hosts) on behalf of the display and offer a menu of possible hosts that offer XDMCP display management. This feature is useful with X terminals that do not offer a host menu themselves.

Because the **xdm** command provides the first interface that users see, it is designed to be simple to use and easy to customize to the needs of a particular site.

Typical Usage

The **xdm** command is designed to operate in a wide variety of environments.

First, set up the **xdm** configuration file. Make a directory (usually **/usr/lib/X11/xdm**) to contain all of the relevant files. The following is a reasonable configuration file, which could be named **xdm-config**:

```
DisplayManager.servers:      /usr/lib/X11/xdm/Xservers
DisplayManager.errorLogFile: /usr/lib/X11/xdm/xdm-errors
DisplayManager*resources:    /usr/lib/X11/xdm/Xresources
DisplayManager*startup:      /usr/lib/X11/xdm/Xstartup
DisplayManager*session:      /usr/lib/X11/xdm/Xsession
DisplayManager.pidFile:      /usr/lib/X11/xdm/xdm-pid
DisplayManager._0.authorize:  true
DisplayManager*authorize:    false
```

This file contains references to other files. Some of the resources are specified with an * (asterisk) separating the components. These resources can be made unique for each display by replacing the * (asterisk) with the display name, but typically this is not useful. See the [Resources](#) section on the next page for a complete discussion.

The first file, **/usr/lib/X11/xdm/Xservers**, contains the list of displays to manage that are not using **XDMCP**. Most workstations have only one display, numbered 0 (zero), so the file looks something like this:

```
:0 Local local /usr/bin/X11/X -force
```

This keeps **/usr/bin/X11/X** running on this display and manages a continuous cycle of sessions.

The **/usr/lib/X11/xdm/xdm-errors** file contains error messages from **xdm** and anything output to standard error by **Xsetup**, **Xstartup**, **Xsession** or **Xreset** scripts. If you have trouble starting the **xdm** command, check the **/usr/lib/X11/xdm/xdm-errors** file to see if the **xdm** command has any clues to the trouble.

The next configuration entry, **/usr/lib/X11/xdm/Xresources**, is loaded onto the display as a resource database using the **xrdb** command. As the authentication widget reads this database before starting, it usually contains parameters for that widget.

Flags

All of these options (except **-config**) specify values that can also be specified in the configuration file as resources.

Item	Description
-config <i>ConfigurationFile</i>	Names the configuration file, which specifies resources to control the behavior of the xdm command. The /usr/lib/X11/xdm/xdm-config file is the default.
-debug <i>DebugLevel</i>	Specifies the numeric value for the DisplayManager.debugLevel resource. A nonzero value causes xdm to print debugging statements to the terminal and disables the DisplayManager.daemonMode resource, forcing xdm to run synchronously. These error messages may be unclear. To interpret them, check the X11R4 source code for the xdm command.

Item	Description
-nodaemon	Specifies False as the value for the DisplayManager.daemonMode resource. This suppresses the usual daemon behavior, in which the xdm command closes all file descriptors, disassociates itself from the controlling terminal, and puts itself in the background when it first starts up.
-error <i>ErrorLogFile</i>	Specifies the value for the DisplayManager.errorLogFile resource. This file contains errors from xdm as well as anything written to standard error by the various scripts and programs run during the progress of the session.
-resources <i>ResourceFile</i>	Specifies the value for the DisplayManager*resources resource. This file is loaded using the xrdb command to specify configuration parameters for the authentication widget.
-server <i>ServerEntry</i>	Specifies the value for the DisplayManager.servers resource. See the section Server Specification for a description of this resource.
-udpPort <i>PortNumber</i>	Specifies the value for the DisplayManager.requestPort resource. This sets the port number that the xdm command monitors for XDMCP requests. XDMCP uses the registered well-known UDP port 177. Do not change this resource except when debugging.
-session <i>SessionProgram</i>	Specifies the value for the DisplayManager*session resource. This indicates the program to run as the session after the user has logged in.
-xrm <i>ResourceSpecification</i>	Allows an arbitrary resource to be specified, as in most X Toolkit applications.

Resources

At many stages, the actions of **xdm** can be controlled through the use of its configuration file, which is in the X resource format. Some resources modify the behavior of **xdm** on all displays, while others modify its behavior on a single display. When actions relate to a specific display, the display name is inserted into the resource name between "DisplayManager" and the final resource name segment. For example, **DisplayManager.expo_0.startup** is the name of the resource that defines the startup shell file on the "expo:0" display. Because the resource manager uses colons to separate the name of the resource from its value and dots to separate resource name parts, **xdm** substitutes underscores for both dots and colons when generating the resource name.

Item	Description
DisplayManager.servers	Specifies either a file name full of server entries, one per line (if the value starts with a slash), or a single server entry. See the section Server Specification for details.
DisplayManager.requestPort	Indicates the UDP port number that the xdm command uses to listen for incoming XDMCP requests. Unless you need to debug the system, leave this with its default value of 177.
DisplayManager.errorLogFile	Redirects error messages to go to the named file rather than to the console. This file also contains any output directed to standard error by the Xsetup , Xstartup , Xsession , and Xreset files, so it will contain descriptions of problems in those scripts as well.

Item	Description
DisplayManager.debugLevel	If the integer value of this resource is greater than 0 (zero), the xdm command outputs a large amount of debugging information. It also disables daemon mode, which would discard the information and allow nonroot users to run the xdm command that would typically not be useful.
DisplayManager.daemonMode	The xdm command attempts to make itself into a daemon process unassociated with any terminal. This is accomplished by forking and leaving the parent process to exit, and then closing file descriptors and releasing the controlling terminal. In some environments this is not desired (in particular, when debugging). Setting this resource to False disables this feature.
DisplayManager.pidFile	The file name specified is created to contain an ASCII representation of the process ID of the main xdm process. The xdm command also uses file locking on this file to attempt to eliminate multiple daemons running on the same machine, which would have unpredictable results.
DisplayManager.lockPidFile	Controls whether the xdm command uses file locking to keep multiple display managers from running simultaneously.
DisplayManager.authDir	Names a directory in which the xdm command stores authorization files while initializing the session. The default value is /usr/lib/X11/xdm .
DisplayManager.autoRescan	A Boolean value that controls whether the xdm command rescans the configuration, servers, access control, and authentication keys files after a session ends and the files have changed. By default the value is True. You can force the xdm daemon to reread these files by sending a SIGHUP signal to the main process.
DisplayManager.removeDomainname	When computing the display name for XDMCP clients, the name resolver typically creates a fully qualified host name for the terminal. As this is sometimes confusing, the xdm command removes the domain name portion of the host name if it is the same as the domain name of the local host when this variable is set. The default value is True.
DisplayManager.keyFile	XDM-AUTHENTICATION-1 style XDMCP authentication requires that a private key be shared between the xdm daemon and the terminal. This resource specifies the file containing those values. Each entry in the file consists of a display name and the shared key. By default, the xdm command does not include support for XDM-AUTHENTICATION-1 because it requires the data encryption method (DES), which is not generally distributable because of United States export restrictions.

Item	Description
DisplayManager.accessFile	To prevent unauthorized XDMCP service and to allow forwarding of XDMCP IndirectQuery requests, this file contains a database of host names that are allowed direct access to this machine or have a list of hosts to which queries should be forwarded. The format of this file is described in the XDMCP Access Control section.
DisplayManager.exportList	A whitespace-separated list of additional environment variables to pass on to the Xsetup , Xstartup , Xsession , and Xreset programs.
DisplayManager.randomFile	A file to checksum to generate the seed of authorization keys. This should be a file that changes frequently. The default is /dev/mem .
DisplayManager.choiceTimeout	Number of seconds to wait for the display to respond after a user has selected a host from the chooser. If the display sends an XDMCP IndirectQuery within this time, the request is forwarded to the chosen host. Otherwise, it is assumed to be from a new session and the chooser is offered again. The default is 15.
DisplayManager.DISPLAY.resources	Specifies the name of the file to be loaded by the xrdb command as the resource database onto the root window of screen 0 of the display. The Login widget, Xsetup , and chooser programs use the resources set in this file. This resource data base is loaded just before the authentication procedure is started, so it can control the appearance of the login window. See the section Authentication Client , which describes the various resources that are appropriate to place in this file. There is no default value for this resource, but /usr/lib/X11/xdm/Xresources is the conventional name.
DisplayManager.DISPLAY.chooser	Specifies the program run to offer a host menu for indirect queries redirected to the special host name CHOOSER . /usr/lib/X11/xdm/chooser is the default. See the sections XDMCP Access Control and Chooser .
DisplayManager.DISPLAY.xrdb	Specifies the program used to load the resources. By default, the xdm command uses /usr/bin/X11/xrdb .
DisplayManager.DISPLAY.cpp	Specifies the name of the C preprocessor that is used by the xrdb command.
DisplayManager.DISPLAY.setup	Specifies a program that is run (as root) before offering the login window. This resource may be used to change the appearance of the screen around the login window or to put up other windows (for example, you may want to run xconsole here). By default, no program is run. The conventional name for a file used here is Xsetup . See the section Setup Program .
DisplayManager.DISPLAY.startup	Specifies a program that is run (as root) after the authentication process succeeds. By default, no program is run. The conventional name for a file used here is Xstartup . See the section Startup Program .

Item	Description
DisplayManager.DISPLAY.session	Specifies the session to be run (when not running as root). By default, /usr/bin/X11/xterm is run. The conventional name is the Xsession script. See the section Session Program .
DisplayManager.DISPLAY.reset	Specifies a program that is run (as root) after the session ends. By default, no program is run. The conventional name is the Xreset script. See the section Reset Program .
DisplayManager.DISPLAY.openDelay	Controls the behavior of the xdm command when attempting to open intransigent servers by specifying the length of the pause (in seconds) between successive attempts.
DisplayManager.DISPLAY.openRepeat	Controls the behavior of the xdm command when attempting to open intransigent servers by specifying the number of attempts to make.
DisplayManager.DISPLAY.openTimeout	Controls the behavior of the xdm command when attempting to open intransigent servers by specifying the number of seconds to wait while actually attempting the open (that is, the maximum time spent in the connect(2) system call).

Item	Description
DisplayManager.DISPLAY.startAttempts	Controls the behavior of the xdm command when attempting to open intransigent servers by specifying the number of times that the entire process is completed before giving up on the server. After the number of attempts specified by the Display Manager openRepeat resource have been made, or if the number of seconds specified by the Display Manager openTimeout resource elapse in any particular attempt, the xdm command ends and restarts the server, attempting to connect again. This process is repeated <i>startAttempts</i> times, at which point the display is declared inactive and disabled. Although this behavior may seem arbitrary, it has been empirically developed and works well on most systems. The default is a value of 5 for <i>openDelay</i> , a value of 5 for <i>openRepeat</i> , a value of 30 for <i>openTimeout</i> , and a value of 4 for <i>startAttempts</i> .
DisplayManager.DISPLAY.pingInterval	To discover when remote displays disappear, the xdm command occasionally pings them, using an X connection and XSync calls. This resource specifies the time (in minutes) between ping attempts. By default, it is set to 5 minutes. If you frequently use X terminals, which can become isolated from the managing host, you may want to increase this value. Note: AIXwindows sessions may continue to exist after the terminal has been accidentally disabled. The xdm command does not ping local displays. A workstation session can be ended if the server hangs for NFS service and does not respond to the ping.

Item	Description
DisplayManager.DISPLAY.pingTimeout	To discover when remote displays disappear, the xDM command occasionally pings them, using an X connection and XSync calls. This resource specifies the maximum amount of time (in minutes) to wait for the terminal to respond to the request. If the terminal does not respond, the session is declared inactive and ended. By default, it is set to 5 minutes. If you frequently use X terminals, which can become isolated from the managing host, you may want to increase this value. Note: AIXwindows sessions may continue to exist after the terminal has been accidentally disabled. The xDM command does not ping local displays. A workstation session could be ended if the server hangs for NFS service and does not respond to the ping.
DisplayManager.DISPLAY.terminateServer	Specifies whether the X server should be canceled when a session ends (instead of resetting it). This option can be used when the server tends to grow without bound over time, to limit the amount of time the server is run. The default value is False.
DisplayManager.DISPLAY.userPath	The xDM command sets the PATH environment variable for the session to this value. It should be a list of directories separated by colons; see the sh command in <i>Commands Reference</i> for a full description. :/bin:/usr/bin:/usr/bin/X11:/usr/ucb is a common setting. The default value can be specified at build time in the AIXwindows system configuration file with the DefaultUserPath resource.
DisplayManager.DISPLAY.systemPath	The xDM command sets the PATH environment variable for the startup and reset scripts to the value of this resource. The default for this resource is specified at build time by the DefaultSystemPath resource entry in the system configuration file; /etc:/bin:/usr/bin:/usr/bin/X11:/usr/ucb is a common choice. Note the absence of . (period) (the current directory) from this entry. This is a good practice to follow for root; it avoids many common "Trojan Horse" system penetration schemes.
DisplayManager.DISPLAY.systemShell	The xDM command sets the SHELL environment variable for the startup and reset scripts to the value of this resource. It is /bin/sh by default.
DisplayManager.DISPLAY.failSafeClient	If the default session fails to run, the xDM command returns to this program. This program is run with no arguments, using the same environment variables as the session would have had (see the section Session Program). By default, /usr/bin/X11/xterm is used.

Item	Description
DisplayManager.DISPLAY.grabServer DisplayManager.DISPLAY.grabTimeout	To improve security, the xdm command grabs the server and keyboard while reading the login name and password. The grabServer resource specifies if the server should be held for the duration of the name/ password reading. When set to False, the server is ungrabbed after the keyboard grab succeeds, otherwise the server is grabbed until just before the session begins. The default value is False. The grabTimeout resource specifies the maximum time that the xdm command waits for the grab to succeed. The grab may fail if some other client has the server grabbed, or possibly if the network latencies are high. This resource has a default value of 3 seconds; be cautious when raising it, as a user may be confused by a look-alike window on the display. If the grab fails, the xdm command becomes inactive and restarts the server (if possible) and the session.
DisplayManager.DISPLAY.authorize DisplayManager.DISPLAY.authName	The authorize is a Boolean resource that controls whether the xdm command generates and uses authorization for the local server connections. If authorization is used, the xdm command uses the authorization mechanisms indicated as a whitespace-separated list as the value of the authName resource. XDMCP connections dynamically specify which authorization mechanisms are supported, so the authName resource is ignored in this case. When the authorize resource is set for a display and authorization is not available, the user is informed by a different message displayed in the Login widget. By default, the authorize resource is True; authName is MIT-MAGIC-COOKIE-1.
DisplayManager.DISPLAY.authFile	Indicates the file is used to communicate the authorization data from the xdm command to the server, using the -auth server command-line option. It should be kept in a directory with restricted write permissions as it could easily be removed, disabling the authorization mechanism in the server.
DisplayManager.DISPLAY.authComplain	If set to a value of False, this disables the use of the unsecureGreeting in the login window. See the section Authentication Client . The default is a value of True.
DisplayManager.DISPLAY.resetSignal	The number of the signal that the xdm command sends to reset the server. See the section Controlling the Server . The default is 1(SIGHUP).
DisplayManager.DISPLAY.termSignal	The number of the signal that the xdm command sends to end the server. See the section Controlling the Server . The default is 15(SIGTERM).
DisplayManager.DISPLAY.resetForAuth	Causes the xdm command to send SIGHUP to the server after setting up the authorization file, causing an additional server reset to occur, during which time the new authorization information is read. The default is a value of False, which works for all AIXwindows servers.

Item	Description
DisplayManager.DISPLAY.userAuthDir	When the xdm command is unable to write to the usual user authorization file (\$HOME/.Xauthority), it creates a unique file name in this directory and sets the XAUTHORITY environment variable to the name of the created file. It uses /tmp by default.

XDMCP Access Control

The database file specified by the **DisplayManager.accessFile** resource provides information that the **xdm** command uses to control access from displays requesting **XDMCP** service. This file contains three types of entries:

- Entries that control the response to **Direct** and **Broadcast** queries.
- Entries that control the response to **Indirect** queries.
- Macro definitions.

Direct query entries contain either a host name or a pattern, which is distinguished from a host name by the inclusion of one or more pattern-matching characters. An ***** (asterisk) matches any sequence of 0 (zero) or more characters, and a **?** (question mark) matches any single character. These are compared against the host name of the display device. If the entry is a host name, all comparisons are done using network addresses, so that any name that converts to the correct network address may be used. For patterns, only actual host names are used in the comparison, so ensure that you do not attempt to match aliases. Preceding either a host name or a pattern with an **!** (exclamation point) causes hosts that match that entry to be excluded.

An Indirect entry also contains a host name or pattern, but follows it with a list of host names or macros to which **indirect** queries should be sent.

A macro definition contains a macro name and a list of host names and other macros that the macro expands to. To distinguish macros from host names, macro names start with a **%** (percent) character. Macros may be nested.

Indirect entries may also specify to have the **xdm** command run the **chooser** command to offer a menu of hosts to which to connect. For more information, see [Chooser](#).

When checking access for a particular display host, each entry is scanned in turn and the first matching entry determines the response. For example, a **Direct** query entry is ignored when scanning for an **Indirect** entry. A **Broadcast** query entry is ignored when scanning for a **Direct** entry.

Blank lines are ignored. The **#** character is treated as a comment delimiter causing the rest of that line to be ignored, and a **** (backslash) at the end of the line causes the new line to be ignored, allowing indirect host lists to span multiple lines.

The following is an example **Xaccess** file:

```
#
# Xaccess - XDMCP access control file
#

#
# Direct/Broadcast query entries
#

!xtra.lcs.mit.edu      # disallow direct/broadcast service for xtra
bambi.ogi.edu         # allow access from this particular display
*.lcs.mit.edu         # allow access from any display in LCS

#
# Indirect query entries
#
```

```
%HOSTS      expo.lcs.mit.edu xenon.lcs.mit.edu \\  
             excess.lcs.mit.edu kanga.lcs.mit.edu
```

```
extract.lcs.mit.edu  xenon.lcs.mit.edu    #force extract to contact xenon  
!xtra.lcs.mit.edu    dummy              #disallow indirect access  
*.lcs.mit.edu        %HOSTS              #all others get to choose
```

Chooser

For X terminals that do not offer a host menu for use with **Broadcast** or **Indirect** queries, the **chooser** program can do this for them. In the **Xaccess** file, specify **CHOOSE**R as the first entry in the Indirect host list. The **chooser** program sends a **Query** request to each of the remaining host names in the list and offers a menu of all the hosts that respond.

The list may consist of the word **BROADCAST**, in which case **chooser** sends a **Broadcast** query instead, again offering a menu of all hosts that respond.

The following is an example **Xaccess** file using **chooser**:

```
extract.lcs.mit.edu  CHOOSE R  %HOSTS      #offer a menu of these hosts  
xtra.lcs.mit.edu    CHOOSE R  BROADCAST  #offer a menu of all hosts
```

The program to use for **chooser** is specified by the **DisplayManager.DISPLAY.chooser** resource. Resources for this program can be put into the file named by the **DisplayManager.DISPLAY.resources** resource.

The **chooser** has been implemented using a Motif **SelectionBoxWidget**. Refer to the **XmSelectionBoxWidget Class** documentation for a description of resources and widget or gadget names.

Server Specification

The resource **DisplayManager.servers** gives a server specification or, if the values starts with a / (slash), the name of a file containing server specifications, one per line.

Each specification indicates a display that should constantly be managed and that is not using **XDMCP**. Each consists of at least three parts:

- Display name
- Display class
- Display type
- For local servers, a command line to start the server.

A typical entry for local display number 0 would be:

```
:0 IBM-GT local /usr/bin/X11/X :0
```

The display types are:

Item	Description
local	local display: \fIXdm\fp must run the server
foreign	remote display: \fIXdm\fp opens an X connection to a running server

The display name must be something that can be passed in the **-display** option to an X program. This string is used to generate the display-specific resource names, so be careful to match the names (for example, use `:0 local /usr/bin/X11/X :0` instead of `"localhost:0 local /usr/bin/X11/X :0"` if your other resources are specified as `"DisplayManager._0.session"`). The display class portion is also used in the display-specific resources as the class of the resource. This is useful if you have a large collection of similar displays (like a corral of X terminals) and would like to set resources for groups of them. When using **XDMCP**, the display is required to specify the display class, so the manual for your particular X terminal should

document the display class string for your device. If it does not, you can run the **xdm** command in debug mode and look at the resource strings that it generates for that device, which will include the class string.

Setup Program

The **Xsetup** file is run after the server is reset, but before the login window is offered. The file is typically a shell script. It is run as root, so be careful about security. This is the place to change the root background or bring up other windows that should be displayed on the screen along with the Login widget. Because **xdm** grabs the keyboard, other windows will not be able to receive keyboard input. They will be able to interact with the mouse, however; beware of potential security holes here. If **DisplayManager.DISPLAY.grabServer** is set, **Xsetup** will not be able to connect to the display at all. Resources for this program can be put into the file named by **DisplayManager.DISPLAY.resources**.

In addition to any specified by **DisplayManager.exportList**, the following environment variables are passed:

Item	Description
DISPLAY	Specifies the associated display name.
PATH	Specifies the value of DisplayManager.DISPLAY.systemPath .
SHELL	Specifies the value of DisplayManager.DISPLAY.systemShell .
XAUTHORITY	Specifies that it may be set to an authority file.

Authentication Client

The MIT authentication widget has been replaced by an authentication client composed of standard Motif widgets. The following is a list of the widget names (and their widget class):

```
outframe(xmFrameWidget)
inframe(xmFrameWidget)
main(XmFormWidget)
  tframe(xmFrameWidget)
    greeting(xmLabelGadget)
    logoline(xmFormWidget)
    dpyname(xmLabelWidget)
    userline(xmRowColumnWidget)
      userlabel(xmLabelWidget)
      username(xmTextWidget)
      passlabel(xmLabelWidget)
      password(xmTextWidget)
    failsafeline(xmFormWidget)
      failsafe(xmToggleButtonWidget)
    cancelline(xmFormWidget)
      cancel(xmPushButtonWidget)
    message(xmLabelWidget)
```

The authentication client reads a name/password pair from the keyboard. Put resources for this client into the file named by **DisplayManager.DISPLAY.resources**. All of these have reasonable default values, so it is unnecessary to specify any of them. See **/usr/lib/X11/xdm/Xresources** for more information on default values for authentication client resources as well as the appropriate widget class documentation. The following resources are also supported by the authentication client:

Item	Description
Xlogin*foreground	Specifies the color used for the foreground.
Xlogin*background	Specifies the color used for the background.
Xlogin*greeting	Specifies a string that identifies this window. The default is AIXwindows environment.
Xlogin*greetFont	Specifies the font used to display the greeting.
Xlogin*frameColor	Specifies the background color used to display the greeting.
Xlogin*titleFont	Specifies the font used to display the title.
Xlogin*namePrompt	Specifies the string displayed to prompt for a user name. The Xrdb program strips trailing white space from resource values. Add spaces escaped with backslashes at the end of the prompt. The default is "login:".

Item	Description
Xlogin*promptFont	Specifies the font used to display both prompts.
Xlogin*failFont	Specifies the font used for the failsafe button.
Xlogin*cancelFont	Specifies the font used for the cancel button.
Xlogin*messageFontlist	Specifies the font used to display the failure message.
Xlogin*failColor	Specifies the color used to display the failure message.
Xlogin*failTimeout	Specifies the number of seconds that the failure message is displayed. The default is thirty seconds.
Xlogin*sessionArgument	Specifies the argument to be passed to the session program.
Xlogin*XmText.translations	This specifies the translations use for the authentication client. Refer to the X Toolkit documentation for a complete discussion on translations. The default translation table is:

```
Ctrl<Key>b: backward-character()\n\
Ctrl<Key>a: beginning-of-line()\n\
Ctrl<Key>e: end-of-line()\n\
Ctrl<Key>f: forward-character()\n\
Ctrl<Key>d: kill-next-character()\n\
Ctrl<Key>k: kill-to-end-of-line()\n\
Ctrl<Key>u: kill-to-start-of-line()\n\
```

You may setup XDM to use the standard XDM translations by replacing the XmText translations as defined in Xresources:

Note: Use <Key>osfHelp instead of <Key>F1 due to the Motif default virtual bindings.)

```
Xlogin*XmText.translations: #override\n\
<Key>osfHelp:      set-session-argument(failsafe) finish-field()\n\
Ctrl<Key>Return:   set-session-argument(failsafe) finish-field()\n\
Ctrl<Key>H:        delete-previous-character() \n\
Ctrl<Key>D:        delete-character() \n\
Ctrl<Key>B:        move-backward-character() \n\
Ctrl<Key>F:        move-forward-character() \n\
Ctrl<Key>A:        move-to-beginning() \n\
Ctrl<Key>E:        move-to-end() \n\
Ctrl<Key>K:        erase-to-end-of-line() \n\
Ctrl<Key>U:        erase-line() \n\
Ctrl<Key>X:        erase-line() \n\
<Key>Return:      finish-field() \n\
<Key>BackSpace:   delete-previous-character() \n\
<Key>Delete:      delete-previous-character() \n\
```

In addition to the typical XmText actions, the following actions are also supported by the client to be compatible with the standard XDM translations:

delete-previous-character

Erases the character before the cursor.

delete-character

Erases the character after the cursor.

move-backward-character

Moves the cursor backward.

move-forward-character

Moves the cursor forward.

move-to-beginning

Moves the cursor to the beginning of the editable text.

move-to-end

Moves the cursor to the end of the editable text.

erase-to-end-of-line

Erases all text after the cursor.

erase-line

Erases the entire text.

finish-field

If the cursor is in the name field, proceeds to the password field; if the cursor is in the password field, checks the current name/password pair. If the name/password pair is valid, **xm** starts the session. Otherwise the failure message is displayed and the user is prompted again.

insert-char

Inserts the character typed.

set-session-argument

Specifies a single word argument that is passed to the session at startup. See the sections Session Program and Typical Usage.

Startup Program

The **Xstartup** file is typically a shell script. Because it is run as the root user, be careful about security when it runs. It usually contains commands that add entries to **/etc/utmp**, mount users' home directories from file servers, display the message of the day, or cancel the session if logins are not allowed.

In addition to the environment variables specified by **DisplayManager.exportList**, the following variables are passed:

Item	Description
DISPLAY	Specifies the associated display name.
HOME	Specifies the initial working directory of the user.
USER	Specifies the user name.
PATH	Specifies the value of DisplayManager.DISPLAY.systemPath .
SHELL	Specifies the value of DisplayManager.DISPLAY.systemShell .
XAUTHORITY	May be set to an authority file.

No arguments are passed to the script. The **xdm** command waits until this script exits before starting the user session. If the exit value of this script is nonzero, the **xdm** command discontinues the session and starts another authentication cycle.

Session Program

The **Xsession** program establishes the style of the user's session. It is run with the permissions of the authorized user.

In addition to any specified by **DisplayManager.exportList**, the following environment variables are passed:

Item	Description
DISPLAY	Specifies the associated display name.
HOME	Specifies the initial working directory of the user.
USER	Specifies the user name.
PATH	Specifies the value of DisplayManager.DISPLAY.userPath .
SHELL	Specifies the user's default shell (from getpwnam).
XAUTHORITY	May be set to a nonstandard authority file.

At most installations, the **Xsession** program should look in the user's home directory (**\$HOME**) for a file **.xsession**, which contains the commands that the user would like to use as a session. The **Xsession** program should also implement a system default session if no user-specified session exists. See the section Typical Usage.

An argument may be passed to this program from the authentication widget using the 'set-session-argument' action. This can be used to select different styles of session. Usually, this feature is used to allow the user to escape from the ordinary session when it fails. This allows users to repair their own **.xsession** if it fails, without requiring administrative intervention. The section Typical Usage demonstrates this feature.

Reset Program

The **Xreset** script is run after the user session has ended. Run as root, it should contain commands that undo the effects of commands in **Xstartup** by removing entries from **/etc/utmp** or unmounting directories from file servers. The environment variables that are passed to **Xstartup** are also passed to **Xreset**. This program is symmetrical with the **Xstartup** program.

Controlling the Server

The **xdm** command controls local servers using POSIX signals. The **SIGHUP** signal is expected to reset the server, closing all client connections and performing other cleanup duties. The **SIGTERM** signal is expected to cancel the server. If these signals do not perform the expected actions, the resources **DisplayManager.DISPLAY.resetSignal** and **DisplayManager.DISPLAY.termSignal** can specify alternate signals.

To control remote terminals that are not using **XDMCP**, the **xdm** command searches the window hierarchy on the display and uses the protocol request **KillClient** in an attempt to clean up the terminal for the next session. This may not actually cause all of the clients to become inactive, because only those that have created windows will be noticed. **XDMCP** provides a more sure mechanism; when the **xdm** command closes its initial connection, the session is over and the terminal is required to close all other connections.

Controlling XDM

The **xdm** command responds to two signals: **SIGHUP** and **SIGTERM**. When sent a **SIGHUP**, **xdm** rereads the configuration file, the access control file, and the servers file. For the servers file, it notices if entries have been added or removed. If a new entry has been added, the **xdm** command starts a session on the associated display. Entries that have been removed are disabled immediately, meaning that any session in progress is ended without notice and no new session is started.

When sent a **SIGTERM**, the **xdm** command stops all sessions in progress and exits. This can be used when shutting down the system.

The **xdm** command attempts to mark its various subprocesses for use by the **ps** command by editing the command-line argument list in place. Because the **xdm** command cannot allocate additional space for this task, it is useful to start the **xdm** command with a reasonably long command line (using the full path name should be enough). Each process that is servicing a display is marked **-display**.

Other Possibilities

You can use the **xdm** command to run a single session at a time, using the **xinit** command options or other suitable daemons by specifying the server on the command line:

```
xdm -server ":0 local /usr/bin/X11/X :0 -force"
```

It might also run a file server and a collection of X terminals. The configuration for this is identical to the previous sample, except the **Xservers** file would look like the following:

```
extol:0 VISUAL-19 foreign
exalt:0 NCD-19 foreign
explode:0 NCR-TOWERVIEW3000 foreign
```

This directs the **xdm** command to manage sessions on all three of these terminals. See the section [Controlling XDM](#) for a description of using signals to enable and disable these terminals.

Note: The **xdm** command does not coexist well with other window systems. To use multiple window systems on the same hardware, use the **xinit** command.

Examples

1. The sample **xstartup** script that follows prevents login while the file **/etc/nologin** exists. As there is no provision for displaying any messages here (there is no core X client that displays files), the setup in this example is not recommended because the login would fail without explanation. Thus, this is not a complete example, but a demonstration of the available functionality.

```
#!/bin/sh
#
# Xstartup
#
# This program is run as root after the user is verified
#
```

```
if [ \-f /etc/nologin ]; then
    exit 1
fi
exit 0
```

2. This **Xsession** script recognizes the special **failsafe** mode, specified in the translations in the preceding **Xresources** file, to provide an escape from the ordinary session:

```
#!/bin/sh
exec > $HOME/.xsession-errors 2>&1
case $# in
1)
    case $1 in failsafe)
        exec aixterm -geometry 80x24-0-0
        ;;
    esac
esac
startup=$HOME/.xsession
resources=$HOME/.Xresources
if [ -f /usr/bin/X11/startx ]; then
    exec /usr/bin/X11/startx -t -wait
elif [ -f $startup ]; then
    exec $startup
else
    if [ -f $resources ]; then
        xrdp -load $resources
    fi
    mwm &
    exec aixterm -geometry 80x24+10+10 -ls
fi
```

3. To have **xdm** come up from system startup, as root type the following:

```
/usr/lib/X11/xdm/xdmconf
```

4. To disable **xdm** on reboot, as root type the following:

```
/usr/lib/X11/xdm/xdmconf -d
```

5. When using **xdm** to manage your display, an authentication procedure ensures that only clients that are allowed can connect to your display. Clients that are built using X11 R4 and X11 R5 libraries understand this protocol. Clients that are built with X11 R3 or earlier libraries do not support this authentication protocol and are not allowed to connect to the Xserver unless **xhost** permission is granted. You can connect local clients by typing the following:

```
xhost =localhost
```

or

```
xhost =machine
```

where *machine* is the hostname of the local client.

Files

Item	Description
/usr/lib/X11/xdm/xdm-config	The default configuration file.
/usr/lib/X11/xdm/Xaccess	The default access file, listing authorized displays.
/usr/lib/X11/xdm/Xservers	The default server file, listing non-XDMCP servers to manage.
\$(HOME)/.Xauthority	User authorization file where xdm stores keys for clients to read.
/usr/lib/X11/xdm/chooser	The default chooser.
/usr/bin/X11/xrdb	The default resource database loader.

Item	Description
<code>/usr/bin/X11/X</code>	The default server.
<code>/usr/bin/X11/xterm</code>	The default session program and failsafe client.
<code>/usr/lib/X11/xdm/A<host>\-<suffix></code>	The default place for authorization files.

xfindproxy Command

Purpose

Locates proxy services.

Syntax

xfindproxy **-manager** *ManagerAddress* **-name** *ServiceName* **-server** *ServerAddress* [**-auth**] [**-host** *HostAddress*] [**-options** *Options*]

Description

xfindproxy is a program used to locate available proxy services. It utilizes the Proxy Management Protocol to communicate with a proxy manager. The proxy manager keeps track of all available proxy services, starts new proxies when necessary, and makes sure that proxies are shared whenever possible.

If **xfindproxy** is successful in obtaining a proxy address, it will print it to stdout. The format of the proxy address is specific to the proxy service being used. For example, for a proxy service of LBX, the proxy address would be the X display address of the proxy (e.g, `blah.x.org:63`).

If **xfindproxy** is unsuccessful in obtaining a proxy address, it will print an error to **stderr**.

Flags

Item	Description
-manager	This argument is required, and it specifies the network address of the proxy manager. The format of the address is a standard ICE network id (for example, <code>tcp/blah.x.org:6500</code>).
-name	This argument is required, and it specifies the name of the desired proxy service (for example, LBX). The name is case insensitive.
-server	This argument is also required, and it specifies the address of the target server. The format of the address is specific to the proxy service specified with the -name argument. For example, for a proxy service of LBX, the address would be an X display address (e.g, <code>blah.x.org:0</code>).
-auth	This argument is optional. If specified, xfindproxy will read 2 lines from standard input. The first line is an authorization/authentication name. The second line is the authorization/authentication data in hex format (the same format used by xauth). xfindproxy will pass this auth data to the proxy, and in most cases, will be used by the proxy to authorize/authenticate itself to the target server.

Item	Description
-host	This argument is optional. If xfindproxy starts a new proxy service, it will pass the host specified. The proxy may choose to restrict all connections to this host. In the event that xfindproxy locates an already existing proxy, the host will be passed, but the semantics of how the proxy uses this host are undefined.
-options	This argument is optional. If xfindproxy starts a new proxy service, it will pass any options specified. The semantics of the options are specific to each proxy server and are not defined here. In the event that xfindproxy locates an already existing proxy, the options will be passed, but the semantics of how the proxy uses these options are undefined.

xfs Command

Purpose

Supplies fonts to X Window System display servers.

Syntax

xfs [**-config** *ConfigurationFile*] [**-ls** *ListenSocket*] [**-port** *Number*]

Description

xfs is the AIXwindows font server. It supplies fonts to AIXwindows display servers.

The **xfs** server responds to the following signals:

Item	Description
SIGTERM	Causes the font server to exit cleanly.
SIGUSR1	Causes the server to re-read its configuration file.
SIGUSR2	Causes the server to flush any cached data it may have.
SIGHUP	Causes the server to reset, closing all active connections and re-reading the configuration file.

The server is usually run by a system administrator, and started by way of boot files such as **/etc/rc.tcpip**. Users may also wish to start private font servers for specific sets of fonts.

The configuration language is a list of keyword and value pairs. Each keyword is followed by an = (equal sign) and the desired value.

The following list shows recognized keywords and the types and descriptions of valid values:

Item	Description
#	A comment character when located in the first column.

Item	Description
catalogue (List of string)	Ordered list of font path element names. The current implementation only supports a single catalogue ("all"), containing all of the specified fonts.
alternate-servers (List of string)	List of alternate servers for this font server.
client-limit (Cardinal)	Number of clients that this font server will support before refusing service. This is useful for tuning the load on each individual font server.
clone-self (Boolean)	Whether this font server should attempt to clone itself when it reaches the client-limit.
default-point-size (Cardinal)	The default point size (in decipoints) for fonts that do not specify.
default-resolutions (List of resolutions)	Resolutions the server supports by default. This information may be used as a hint for pre-rendering and substituted for scaled fonts which do not specify a resolution. A resolution is a comma-separated pair of x and y resolutions in pixels per inch. Multiple resolutions are separated by commas.
error-file (String)	Filename of the error file. All warnings and errors are logged here.
port (Cardinal)	TCP port on which the server will listen for connections. The default is 7100.
use-syslog (Boolean)	Whether the syslog function (on supported systems) is to be used for errors.
deferglyphs (String)	Set the mode for delayed fetching and caching of glyphs. Value is none, meaning deferred glyphs is disabled. all, meaning deferred glyphs is enabled for all fonts, and 16, meaning deferred glyphs is enabled only for 16-bit fonts.

One of the following forms can be used to name a font server that accepts TCP connections:

```
tcp/hostname:port
tcp/hostname:port/cataloguelist
```

The hostname specifies the name (or decimal numeric address) of the machine on which the font server is running. The port is the decimal TCP port on which the font server is listening for connections. The cataloguelist specifies a list of catalogue names, with '+' as a separator. The following are some examples:

```
tcp/expo.lcs.mit.edu:7100, tcp/18.30.0.212:7101/all
```

One of the following forms can be used to name a font server that accepts DECnet connections:

```
decnet/nodename::font$objname
decnet/nodename::font$objname/cataloguelist
```

The nodename specifies the name (or decimal numeric address) of the machine on which the font server is running. The objname is a normal, case-insensitive DECnet object name. The cataloguelist specifies a list of catalogue names, with '+' as a separator.

Flags

Item	Description
-config <i>ConfigurationFile</i>	Specifies the configuration file the font server will use.
-ls <i>ListenSocket</i>	Specifies a file descriptor that is already set up to be used as the listen socket. This option is only intended to be used by the font server itself when automatically spawning another copy of itself to handle additional connections.
-port <i>Number</i>	Specifies the TCP port number on which the server will listen for connections.

Examples

```
#
# sample font server configuration file
#

# allow a max of 10 clients to connect to this font server
client-limit = 10

# when a font server reaches its limit, start up a new one
clone-self = on

# alternate font servers for clients to use
alternate-servers = hansen:7101,hansen:7102

# where to look for fonts
# the first is a set of Speedo outlines, the second is a set of
# misc bitmaps and the last is a set of 100dpi bitmaps
#
catalogue = /usr/lib/fonts/type1,
            /usr/lib/X11/ncd/fonts/misc,
            /usr/lib/X11/ncd/fonts/100dpi/

# in 12 points, decipoints
default-point-size = 120

# 100 x 100 and 75 x 75
default-resolutions = 100,100,75,75
```

Files

Item	Description
/usr/lib/X11/fs/config	The default configuration file.

xget Command

Purpose

Receives secret mail in a secure communication channel.

Syntax

xget

Description

The **xget** command is used to receive secret mail in a secure communication channel. The messages can be read only by the intended recipient. The **xget** command asks for your password and enables you to read your secret mail.

The **xget** command is used with the **enroll** command and the **xsend** command to send and receive secret mail. The **enroll** command sets up the password used to receive secret mail. The **xsend** command sends mail that can be read only by the intended recipient.

When you issue the **xget** command, you are prompted for your encryption key. Enter the password you previously set up using the **enroll** command.

The prompt for the **xget** command is a ? (question mark). The following subcommands control message disposition:

Item	Description
q (quit)	Writes any mail not yet deleted to the user's mailbox and exits. Pressing End Of File (Ctrl-D) has the same effect.
n (delete) or d (delete) or Enter	Deletes the current message and displays the next message.
!Command	Runs the specified workstation command.
s[Filename]	Saves the message in the named <i>File</i> parameter instead of in the default mail file, mbx .
w[Filename]	Saves the message, without its header, in the specified <i>File</i> parameter instead of in the default mail file mbx .
? (help)	Displays a subcommand summary.

Examples

1. To receive secret mail, enter:

```
xget
```

You are prompted for the password, established with the **enroll** command. After entering your password, the **xget** command prompt (?) and a listing of any secret mail is displayed.

2. To display your secret mail, at the **xget** prompt (?), press the Enter key.

After the most recent message is displayed, a ? (question mark) indicates the **xget** command is waiting for one of the **xget** subcommands. Enter **help** or a ? (question mark) to list the subcommands available.

3. To save a message or a file to the default mail file, enter:

```
xget
```

Press the Enter key after the ? (question mark) prompt until the desired file is displayed. When the appropriate file is displayed, enter:

```
s
```

In this example, the file is saved in the default mail file, **mbx**.

4. To save a message or a file to a specific file, enter:

```
xget
```

Press the Enter key after the ? (question mark) prompt until the desired file is displayed. When the appropriate file is displayed, enter:

```
s mycopy
```

In this example, the file is saved in a file named mycopy, instead of the default mail file.

5. To delete a message, enter:

```
xget
```

Press the Enter key after the ? (question mark) prompt until the desired file is displayed. When the appropriate file is displayed, enter:

```
d
```

In this example, the current file is deleted.

Files

Item	Description
/var/spool/secretmail/<i>User</i>.key	Contains the encrypted key for <i>User</i> .
/var/spool/secretmail/<i>User</i>.[0-9]	Contains the encrypted mail messages for <i>User</i> .
/usr/bin/xget	Contains executable files.

xhost Command

Purpose

Controls who accesses Enhanced X-Windows on the current host machine.

Syntax

```
xhost [ + | - ] [ Name ]
```

Description

The **xhost** command adds or deletes host names on the list of machines from which the X Server accepts connections.

This command must be run from the machine with the display connection. You can remove a name from the access list by using the **-Host** parameter. Do not remove the current name from the access list. If you do, log off the system before making any corrections.

Entering the **xhost** command with no variables shows the current host names with access your X Server and a message indicating whether or not access is enabled.

For security, options that affect access control may only be run from the *controlling host*. For workstations, this is the same machine as the server. For X terminals, it is the login host.

To enable a remote name by default, the name can be defined in the **/etc/X?.hosts** file, where ? is the display number to which you enable access.

For example, the display **jeanne:0** can be accessed by systems defined in the **/etc/X0.hosts** file on a system that uses the default host name of **jeanne**. In both the display name and the file name, 0 indicates the display number that the defined remote systems are allowed to access through Enhanced X-Windows.

Flags

Item	Description
+Name	Defines the host name (the plus sign is optional) to be added to the X Server access list.
- Name	Defines the host name to be removed from the X Server access list. Existing connections are not broken, but new connection attempts will be denied. Note that you can remove the current machine; however, further connections (including attempts to add it back) are not permitted. The only way to allow local connections again is to reset the server (thereby breaking all connections).
+	Specifies that access is unlimited. Access control is turned off.
-	Turns access control on.
	The complete <i>Name</i> has a the following <i>family:name</i> syntax:
inet	Internet host
local	Contains only one name, the empty string
	Note: The family is case sensitive. The format of the name varies with the family.

xinit Command

Purpose

Initializes the X Window System.

Syntax

xinit [[*Client*] *Options*] [- - [*Server*] [*Display*] *Options*]

Description

The **xinit** command starts the AIXwindows server and a first client program on systems that cannot start X directly from **/etc/init** or in environments that use multiple window systems. When this first client exits, the **xinit** command stops the X server and then ends.

If no specific client program is given on the command line, the **xinit** command looks for a file to run to start up client programs. The **xinit** command looks for the **\$XINITRC** environment variable. If the file is not there, it then looks for the **\$HOME/.xinitrc** file. If it still does not find the file, it follows these steps:

1. The **xinit** command looks next to **/usr/lib/X11/\$LANG/xinitrc**.
2. Next, it looks to **/usr/lpp/X11/defaults/\$LANG/xinitrc**.
3. And finally, it looks to **/usr/lpp/X11/defaults/xinitrc**.

If no such file exists, **xinit** uses the following as a default:

```
aixterm \-geometry +1+1 \-n login \-display :0
```

If no specific server program is given on the command line, the **xinit** command follows these steps:

1. The **xinit** command looks for a file to run as a shell script to start up the server. The **xinit** command looks for files first in the **\$XSERVERRC** environment variable.
2. If the file is not there, it looks for the **\$HOME/.xserverrc** file.
3. If it still does not find the **\$HOME/.xserverrc** file, it looks next to **/usr/lpp/X11/defaults/xserverrc** file.

4. And finally, if it does not find any of the previous files, the **xinit** command runs the **X** command to start the X server and uses the following as a default:

```
X :0
```

Note that this assumes that there is a program named X in the current search path. However, servers are usually named *Xdisplaytype* where *displaytype* is the type of graphics display which is driven by this server. The site administrator should, therefore, make a link to the appropriate type of server on the machine, or create a shell script that runs the **xinit** command with the appropriate server.

Note: If you attempt to start AIXwindows without an available pointer device, such as a mouse or a tablet, AIXwindows will not open. Some devices can be plugged in but not defined and thus not available to the system, as well as the reverse.

An important point is that programs which are run by **.xinitrc** should be run in the background if they do not exit right away, so that they do not prevent other programs from starting up. However, the last long-lived program started (usually a window manager or terminal emulator) should be left in the foreground so that the script does not exit (which indicates that the user is done and that xinit should exit).

An alternate client and/or server may be specified on the command line. The desired client program and its arguments should be given as the first command line arguments to **xinit**. To specify a particular server command line, add a **--** (double dash) to the **xinit** command line (after any client and arguments) followed by the desired server command.

Both the client program name and the server program name must begin with a / (slash) or a . (period). Otherwise, they are treated as arguments to be added to their respective startup lines. This makes it possible to add arguments (for example, foreground and background colors) without having to retype the whole command line.

If a clear server name is not given and the first argument following the **--** (double dash) is a : (colon) followed by a number, **xinit** uses that number as the display number instead of zero. All remaining arguments are added to the server command line.

The following environment variables are used with the **xinit** command:

Item	Description
DISPLAY	This variable gets set to the name of the display to which clients should connect.
XINITRC	This variable specifies an init file containing shell commands to start up the initial windows. By default, .xinitrc in the home directory is used.
<i>Options</i>	List any option you wish that is available to the client you specified.
<i>Client</i>	Specify the client with which you are working. For example, xterm or aixterm. The client you specify must begin with a . (dot) or a / (slash).
<i>Server</i>	Use any valid xserver. The server you specify must begin with a . (dot) or a / (slash).

Examples

1. To start up a server named X and run the user's **xinitrc** program, if it exists, or else start an **aixterm** command enter:

```
xinit
```

2. To start a specific type of server on an alternate display, enter:

```
xinit -- /usr/bin/X11/X qdss:1
```

3. To start up a server named X, and add the given arguments to the default **xinitrc** or **aixterm** command, enter:

```
xinit -geometry =80x65+10+10 -fn 8x13 -j -fg white -bg navy
```

4. To use the command **/Xsun -l -c** to start the server and add the arguments **-e widgets** to the default **xinitrc** or **aixterm** command, enter:

```
xinit -e widgets -- ./Xsun -l -c
```

5. To start a server named X on display 1 with the arguments **-a 2 -t 5**, then start a remote shell on the machine **fasthost** in which it runs the command **cpupig**, telling it to display back on the local workstation, enter:

```
xinit /usr/ucb/rsh fasthost cpupig -display ws:1 -- :1 -a 2 -t 5
```

6. The following sample of the **.xinitrc** script starts a clock, several terminals, and leaves the window manager running as the last application. Assuming that the window manager has been configured properly, the user then chooses the **Exit** menu item to end the AIXwindows session.

```
xrdb -load $HOME/.Xresources
xsetroot -solid gray &
xclock -g 50x50-0+0 -bw 0 &
xload -g 50x50-50+0 -bw 0 &
xterm -g 80x24+0+0 &
xterm -g 80x24+0-0 &
mwm
```

7. Sites that want to create a common startup environment could simply create a default **.xinitrc** script that references a site-wide startup file:

```
#!/bin/sh . /usr/local/lib/site.xinitrc
```

8. Another approach is to write a script that starts the **xinit** command with a specific shell script. Such scripts are usually named **x11**, **xstart**, or **startx** and are a convenient way to provide a simple interface for novice users:

```
#!/bin/sh xinit /usr/local/lib/site.xinitrc -- /usr/bin/X11/X bc
```

Files

Item	Description
.xinitrc	Contains the default client script files.
aixterm	Contains the command the client runs if .xinitrc does not exist.
.xserverrc	Contains the default server script.
X	Contains the command the server runs if .xserverrc does not exist.

xkbcomp Command

Purpose

Compiles XKB keyboard description.

Syntax

```
xkbcomp [ -a ] [ -C ] [ -dflts ] [ -I Directory ] [ -l ] [ -m Name ] [ -merge ] [ -o OutputFile ] [ -opt Parts ] [ -R Directory ] [ -synch ] [ -w Level ] [ -xkb ] [ -xkm ] Source [ Destination ]
```

Description

The **xkbcomp** command is a keymap compiler that converts a description of an XKB keymap into one of several output formats. The most common use for **xkbcomp** is to create a compiled keymap file (**.xkm** extension) which can be read directly by XKB-capable X servers or utilities. The keymap compiler can also produce C header files or XKB source files. The C header files produced by **xkbcomp** can be included by X servers or utilities that need a built-in default keymap. The XKB source files produced by **xkbcomp** are fully resolved and can be used to verify that the files which typically make up an XKB keymap are merged correctly or to create a single file which contains a complete description of the keymap.

The *Source* may specify an X display, or an **.xkb** or **.xkm** file; unless explicitly specified, the format of *destination* depends on the format of the source. Compiling a **.xkb** (keymap source) file generates a **.xkm** (compiled keymap file) by default. If the source is a **.xkm** file or an X display, **xkbcomp** generates a keymap source file by default.

If the *Destination* is an X display, the keymap for the display is updated with the compiled keymap.

The name of the *destination* is usually computed from the name of the source, with the extension replaced as appropriate. When compiling a single map from a file which contains several maps, **xkbcomp** constructs the destination file name by appending an appropriate extension to the name of the map to be used.

Flags

Item	Description
-a	Shows all keyboard information, reporting implicit or derived information as a comment. Only affects .xkb format output.
-C	Produces a C header file as output (.h extension).
-dflts	Computes the defaults for any missing components, such as key names.
-I <i>Directory</i>	Specifies the top-level directories to be searched for files included by the keymap description.
-l	List maps that specify the <i>map</i> pattern in any files listed on the command line.
-m <i>Name</i>	Specifies a map to be compiled from an file with multiple entries.
-merge	Merges the compiled information with the map from the server.
-o <i>Name</i>	Specifies a name for the generated output file. The default is the name of the source file with an appropriate extension for the output format.
-opt <i>Parts</i>	Specifies a list of optional parts. Compilation errors in any optional parts are not fatal. <i>Parts</i> may consist of any combination of the letters c , g , k , s , t which specify the compatibility map, geometry , keycodes , symbols , and types , respectively.
-R <i>Directory</i>	Specifies the root directory for relative path names.
-synch	Forces synchronization for X requests.
-w <i>Level</i>	Controls the reporting of warnings during compilation. A warning level of 0 disables all warnings; a warning level of 10 enables them all.

Item	Description
-xkb	Generates a source description of the keyboard as output (.xkb extension).
-xkm	Generates a compiled keymap file as output (.xkm extension).

xkbevd Daemon

Purpose

XKB event daemon.

Syntax

xkbevd [**-help**] [**-cfg** *File*] [**-sc** *Command*] [**-sd** *Directory*] [**-display** *Display*] [**-bg**] [**-synch**] [**-v**]

Description

The **xkbevd** event daemon listens for specified XKB events and executes requested commands if they occur. The configuration file consists of a list of event specification/action pairs and/or variable definitions.

An event specification consists of a short XKB event name followed by a string or identifier which serves as a qualifier in parentheses; empty parenthesis indicate no qualification and serve to specify the default command which is applied to events which do not match any of the other specifications. The interpretation of the qualifier depends on the type of the event:

- Bell events match using the name of the bell.
- Message events match on the contents of the message string.
- Slow key events accept any of **press**, **release**, **accept**, or **reject**.

No other events are recognized.

An action consists of an optional keyword followed by an optional string argument. **xkbevd** recognizes the following actions:

- **none**
- **ignore**
- **echo**
- **printEvent**
- **sound**
- **shell**

If the action is not specified, the string is taken as the name of a sound file to be played unless it begins with an exclamation point, in which case it is taken as a shell command.

Variable definitions in the argument string are expanded with fields from the event in question before the argument string is passed to the action processor. The general syntax for a variable is either:

```
$c
```

or

```
$(str)
```

where *c* is a single character and *str* is a string of arbitrary length. All parameters have both single-character and long names. The list of recognized parameters varies from event to event.

The **ignore**, **echo**, **printEvent**, **sound**, and **shell** actions do what you would expect commands named **ignore**, **echo**, **printEvent**, **sound**, and **shell** to do, except that the **sound** command has only been implemented and tested for SGI machines.

The only currently recognized variables are *soundDirectory* and *soundCommand*.

Flags

Item	Description
-bg	Tells xkbevd to fork itself and run in the background.
-cfg <i>File</i>	Specifies the configuration file to read. If no configuration file is specified, xkbevd looks for ~/.xkb/xkbevd.cf and \$(LIBDIR)/xkb/xkbevd.cf in that order.
-display <i>Display</i>	Specifies the display to use. If not present, xkbevd uses \$DISPLAY .
-help	Prints a usage message.
-sc <i>Command</i>	Specifies the command used to play sounds.
-sd <i>Directory</i>	Specifies a top-level directory for sound files.
-synch	Forces synchronization of all X requests. Slow.
-v	Prints more information, including debugging messages. Multiple specifications of -v causes more output.

xkbprint Command

Purpose

Prints an XKB keyboard description.

Syntax

```
xkbprint [ -? | -help ] [ -color ] [ -dflts ] [ -diffs ] [ -eps ] [ -fit ] [ -full ] [ -grid Resolution ] [ -if FontName ] [ -label Type ] [ -lc Locale ] [ -level1 ] [ -level2 ] [ -lg Group ] [ -ll Level ] [ -mono ] [ -n Number ] [ -nkg Number ] [ -npk Number ] [ -o File ] [ -R Directory ] [ -pict Which ] Source [ OutputFile ]
```

Description

The **xkbprint** command generates a printable or encapsulated PostScript description of the XKB keyboard description specified by *Source*. The *Source* can be any compiled keymap, **.xkm** file, that includes a geometry description or an X display specification. If an *OutputFile* is specified, **xkbprint** writes to it. Otherwise, **xkbprint** creates the output file, replacing the extension of the source file with **.ps** or **.eps** depending on the requested format. If the source is a non-local X display, for example **:0**, **xkbprint** appends the appropriate prefix to the display specification, replacing the colon with a **-** (dash). For a local display, **xkbprint** uses **server-n** where *n* is the number of the display.

Flags

Item	Description
-? -help	Prints a usage message.

Item	Description
-color	Prints using the colors specified in the geometry file; by default, xkbprint prints a black-and-white image of the keyboard.
-dflts	Attempts to compute default names for any missing components, such as keys.
-diffs	Shows symbols only where they are explicitly bound.
-eps	Generates an encapsulated PostScript file.
-fit	Fits the keyboard image on the page, this is the default.
-full	Prints the keyboard at full size.
-grid <i>Resolution</i>	Prints a grid with <i>Resolution</i> mm resolution over the keyboard.
-if <i>FontName</i>	Specifies an internal PostScript type 1 font to dump to the specified output file or to <i>fontName.pfa</i> , if no output file is specified. No keyboard description is printed if an internal font is dumped.
-label <i>Type</i>	Specifies the labels to be printed on keys. Valid types are: • none • name • code • symbols
-lc <i>Locale</i>	Specifies a locale in which KeySyms should be resolved.
-level1	Generates a level 1 PostScript.
-level2	Generates a level 2 PostScript.
-lg <i>Group</i>	Prints symbols in keyboard groups starting from <i>Group</i> .
-ll <i>Level</i>	Prints symbols starting from shift level <i>Level</i> .
-mono	Generates a black-and-white image of keyboard, this is the default.
-n <i>Number</i>	Prints <i>Number</i> of copies.
-nkg <i>Number</i>	Prints the symbols in <i>Number</i> keyboard groups.
-npk <i>Number</i>	Specifies the <i>Number</i> of keyboard images to print on each page. For EPS files, this specifies the total number of keyboard images to print.
-o <i>File</i>	Writes the output to <i>File</i> .
-R <i>Directory</i>	Use <i>Directory</i> as the root directory; all path names are interpreted relative to <i>Directory</i> .

Item	Description
-pict <i>Which</i>	Controls the use of pictographs instead of keysym names where available. Valid values for <i>Which</i> are: • all • none • common (default).
-synch	Forces synchronization for X requests.
-w <i>Level</i>	Sets warning level. • 0 for no warning • 10 for all warnings

xlock Command

Purpose

Locks the local X display until a password is entered.

Syntax

```
xlock [ -batchcount Number ] [ -bg Color ] [ -delay Users ] [ -display Display ] [ -fg Color ]
[ -font FontName ] [ -info TextString ] [ -invalid TextString ] [ -mode ModeName ] [ +mono | -mono ]
[ -username TextString ] [ -nice Level ] [ +nolock | -nolock ] [ -password TextString ] [ +remote |
-remote ] [ +allowaccess | -allowaccess ] [ +allowroot | -allowroot ] [ +echokeys | -echokeys ]
[ +enablesaver | -enablesaver ] [ -help ] [ -saturation Value ] [ -timeout Seconds ] [ +usefirst |
-usefirst ] [ +v | -v ] [ -validate TextString ]
```

Description

The **xlock** command locks the X server until the user enters a password at the keyboard. While the **xlock** command is running, all new server connections are refused. The screen saver is disabled, the mouse cursor is turned off, the screen is blanked, and a changing pattern is displayed. If a key or a mouse button is pressed, a prompt asks for the password of the user who started the **xlock** command.

If the correct password is typed, the screen is unlocked and the X server is restored. When typing the password, Ctrl-U and Ctrl-H are active as kill and erase, respectively. To return to the locked screen, click in the small icon version of the changing pattern.

To function properly, **xlock** needs to run with root permission since the operating system restricts access to the password and access control files. To give **xlock** root permission, perform the following steps:

1. Log in as root.
2. Go to the directory that contains the **xlock** program file.
3. Run these two commands:
 - a. **chown root xlock**
 - b. **chmod u+s xlock**

Authentication

The **xlock** command is a Pluggable Authentication Module (PAM) enabled X server command that locks the X server until the user enters a password. It supports both local UNIX authentication and PAM authentication for unlocking the X server.

You can set the system-wide configuration to use PAM for authentication by providing root user access and by modifying the value of the *auth_type* attribute to *PAM_AUTH* in the **usw** stanza of the */etc/security/login.cfg* file.

The authentication mechanisms that are used when PAM is enabled are dependent on the configuration of the login service in the */etc/pam.conf* file. The **xlock** command requires the */etc/pam.conf* file entry for the **auth**, **account**, **password**, and **session** module types. The following configuration is recommended for the */etc/pam.conf* file entry in the **xlock** command:

xlock	auth	required	pam_aix
xlock	account	required	pam_aix
xlock	password	required	pam_aix
xlock	session	required	pam_aix

Flags

Item	Description
-batchcount <i>Number</i>	Sets the number of things to do per batch. <i>Number</i> refers to different things depending on the mode: qix Refers to the number of lines rendered in the same color. hop Refers to the number of pixels rendered in the same color. image Refers to the number of sunlogos on screen at once. swarm Refers to the number of bees life and blank Does not apply.
-bg <i>Color</i>	Sets the color of the background on the password screen.
-delay <i>Number</i>	Sets the speed at which a mode operates to the number of microseconds to delay between batches of hopalong pixels, qix lines, life generations, image bits, and swarm motions. In the blank mode, it is important to set this to a small number because the keyboard and mouse are only checked after each delay. A delay of zero would needlessly consume the processing unit while checking for mouse and keyboard input in a tight loop since the blank mode has no work to do.
-display <i>Display</i>	Sets the X11 display to lock. The xlock command locks all available screens on the server and restricts you to locking only a local server, such as unix:0 , localhost:0 , or :0 (unless you set the -remote flag).
-fg <i>Color</i>	Sets the color of the text on the password screen.
-font <i>FontName</i>	Sets the font to be used on the prompt screen.
-help	Prints a brief description of available options.
-info <i>TextString</i>	Defines an informational message. The default is Enter password to unlock; select icon to lock.
-invalid <i>TextString</i>	Specifies an password message. The default is Invalid login.

Item	Description
-mode <i>ModeName</i>	Specifies one the following six display modes: blank Displays a black screen. hop Displays the real plane fractals from the September, 1986 issue of <i>Scientific American</i> . image Displays several randomly appearing sun logos. life Displays Conway's game of life. qix Displays spinning lines. swarm Displays a swarm of bees following a wasp.
-nice <i>NiceLevel</i>	Sets system nicelevel of the xlock process.
-password <i>TextString</i>	Specifies the password prompt string. The default is Password:.
-saturation <i>Value</i>	Sets saturation of the color ramp. A value of 0 (zero) is grayscale and a value of 1 is very rich color. A value of 0.4 is a medium pastel.
-timeout <i>Seconds</i>	Sets the number of seconds before the password screen times out.
-username <i>TextString</i>	Specifies the message shown in front of the user name. The default is Name:.
-validate <i>TextString</i>	Specifies the message that is shown while validating the password. The default is Validating login....
-/+allowaccess	Allows the disabling of the access control list, but still causes the local server to prompt for a password. If xlock is killed using the -KILL command, the access control list is not lost. This flag is also needed when running the xlock command remotely since access to the control list is restricted.
-/+allowroot	Allows the root password to unlock the server as well as the user who started the xlock command.
-/+echokeys	Causes the xlock command to echo to screen a '?' (question mark) character for each key typed into the password prompt. The default is no echo.
+/-enablesaver	Enables the default screensaver. It is possible to set delay parameters long enough to cause phosphor burn on some displays. This flag can be used as an added precaution.
+/-mono	Causes the xlock command to display monochrome (black and white) pixels rather than the default colored ones on color displays.
+/-nolock	Causes the xlock command to only draw the patterns and not to lock the display. A keypress or a mouse click terminates the screen saver.
+/-remote	Allows remote locking of X11 servers. This flag should be used with care. It is intended mainly to lock X11 terminals that cannot run the xlock command locally. If you lock a workstation other than your own, that person will need your password to unlock it. The -remote option does not disable your ability to toggle to another shell.

Item	Description
+/-usefirst	Allows using the keystroke which obtained the password screen as the first input character in the password. The default ignores the first keystroke.
+/-v	Minus prefix enables the verbose mode to tell which options the xlock command is going to use. The plus prefix is the default.

xlsfonts Command

Purpose

Displays the font list for X-Windows.

Syntax

xlsfonts [**-display** *Host:Display*] [**-l** [**l** [**l**]]] [**-m**] [**-C**] [**-1**] [**-w** *Width*] [**-n** *Columns*] [**-u**] [**-o**] [**-fn** *Pattern*]

Description

The **xlsfonts** command lists the fonts that match a specified *Pattern* parameter. Use the wildcard character "*" (asterisk) to match any sequence of characters (including none), and the "?" (question mark) to match any single character. If no pattern is given, "*" is assumed.

Note: The "*" and "?" characters must be placed within quotation marks to prevent them from being expanded by the shell.

You can use flags to specify servers, number and width of columns to print, size of font listings, whether the output should be sorted, and whether to use **OpenFont** instead of **ListFonts**.

Flags

Note: Using the **-l** (lowercase L) flag of the **xlsfonts** command can tie up your server for a long time. This is typical of single-threaded non-preemptable servers, and not a program error.

Item	Description
-1	Indicates that listings should use a single column. This flag is the same as the -n 1 flag.
-C	Indicates that listings should use multiple columns. This flag is the same as the -n 0 flag.
-display <i>Host:Display</i>	Identifies the X Server to contact by specifying the host name and display number.
-fn <i>Pattern</i>	Specifies the fontname <i>Pattern</i> that xlsfonts will list.
-l [l [l]]	(lowercase L) Indicates that medium, long, and very long listings, respectively, should be generated for each font.
-m	Indicates that long listings should also print the minimum and maximum bounds of each font.
-n <i>Columns</i>	Specifies the number of columns to use to display the output. By default, the xlsfonts command tries to fit as many columns of font names into the number of characters specified by the -w <i>Width</i> flag.

Item	Description
-o	Instructs the xlsfonts command to perform OpenFont (and QueryFont , if appropriate) instead of ListFonts . The -o flag is useful if the ListFonts or ListFontsWithInfo fails to list a known font, as is the case with some scaled font systems.
-u	Indicates that the output should remain unsorted.
-w <i>Width</i>	Specifies the width in characters that should be used to determine how many columns to print. The default is 79.

Environment Variable

Item	Description
DISPLAY	Gets the default host and display to use.

Examples

1. To specify a medium-sized list of each font, use a lowercase L and enter:

```
xlsfonts -l
```

2. To specify a three-column list of each font, enter:

```
xlsfonts -n 3
```

3. To display all fonts with the string iso8859 within their names, enter:

```
xlsfonts -ll "*"iso8859"*
```

4. To list all fonts with rom1 plus one following character in their names, enter:

```
xlsfonts rom1"?"
```

This obtains a listing similar to:

```
rom10  rom11  rom14  rom16  rom17
```

xmbind Command

Purpose

Configures virtual key bindings.

Syntax

```
xmbind [ -display Host:Display:ScreenID ] [ FileName ]
```

Description

The **xmbind** command is an X Windows System client that configures the virtual key bindings for AIXwindows applications. This action is performed by the **mwm** command at its startup, so the **xmbind** client is only needed when **mwm** is not in use or when you want to change bindings without restarting **mwm**. If a file is specified, its contents are used as the virtual key bindings. If a file is not specified, the **.motifbind** file in the user's home directory is used. If this file is not found, the **xmbind** command loads the default virtual key bindings.

Flags

Item	Description
-display <i>Host:Display:ScreenID</i>	Specifies the display to use. The -display option has the following parameters: Host Specifies the host name of a valid system on the network. Depending on the situation, this could be the host name of the user or the host name of a remote system. Display Specifies the number (usually 0) of the display on the system on which the output is to be displayed. ScreenID Specifies the number of the screen where the output is to be displayed. This number is 0 for single-screen systems.

Parameters

Item	Description
<i>FileName</i>	Specifies the file containing bindings for virtual mouse and key events.

Exit Status

This command returns the following exit values:

Item	Description
0	Indicates successful completion.
>0	Indicates an error occurred.

xmkmf Command

Purpose

Creates a **Makefile** from an **Imakefile**.

Syntax

xmkmf [**-a**] [*TopDir* [*CurDir*]]

Description

The **xmkmf** command creates a **Makefile** from an **Imakefile** shipped with third-party software. When invoked with no arguments or variables in a directory containing an **Imakefile** file, the **imake** command runs with arguments appropriate for your system (configured into **xmkmf** when X was built) and generates a **Makefile**.

Flag

Item	Description

-a	First builds the Makefile in the current directory, then automatically executes make Makefiles , make includes , and make depend . This is how to configure software that is outside of the MIT X build tree.

Variables

Specify *TopDir* and *CurDir* if you are working inside the MIT X build tree (highly unlikely unless you are an X developer).

Item	Description

<i>TopDir</i>	Specify as the relative path name from the current directory to the top of the build tree.

<i>CurDir</i>	Specify as a relative path name from the top of the build tree to the current directory.

The *CurDir* variable is required if the current directory has subdirectories; otherwise, the **Makefile** will not be able to build the subdirectories. If a *TopDir* variable is given in its place, **xmkmf** assumes nothing is installed on your system and searches for files in the build tree instead of using the installed versions.

xmwlm Command

Purpose

Provides recording of system performance or WLM metrics.

Syntax

```
xmwlm [ -d recording_dir ] [ -n recording_name ] [ -t trace_level ] [ -L ]
```

Description

The *xmwlm* agent provides recording capability for a limited set of local system performance metrics. These include common CPU, memory, network, disk, and partition metrics typically displayed by the *topas* command. Daily recordings are stored in the */etc/perf/daily* directory. The *topasout* command is used to output these recordings in raw ASCII or spreadsheet format. The *xmwlm* agent can also be used to provide recording data from Workload Management (WLM). This is the default format used when *xmwlm* is run without any flags. Daily recordings are stored in the */etc/perf/wlm* directory. The *wlmon* command can be used to process WLM-related recordings. The *xmwlm* agent can be started from the command line, from a user script, or can be placed near the end of the */etc/inittab* file. All recordings cover 24-hour periods and are only retained for seven days by default. You can configure the **retain** value in the */usr/lpp/perfagent/daily.cf* file to change the default recording duration.

Flags

Item	Description

-d <i>recording_dir</i>	

Specifies the output directory for the recording files. The default location is */etc/perf/wlm* when *xmwlm* is run without any flags and */etc/perf/daily* when *xmwlm* is run with the *-L* flag.

-L	

Specifies the collection of *topas*-like metrics. The metric set is not user configurable.

Item	Description
<code>-n recording_name</code>	Specifies the name for the recording file. By default, xmwlm creates recording files named in an xmwlm.YYMMDD format. For example, if -n myrecording is specified, the recording files will be named myrecording.YYMMDD .
<code>-t trace_level</code>	Specifies a trace level. xmwlm prints various information to a log file in the appropriate <code>/etc/perf</code> subdirectory. The trace level can be set from 1 to 9. More trace data is generated at higher trace levels. This trace data is useful to determine xmwlm recording status and for debugging purposes. The log file name is <code>xmwlm.log1</code> or <code>xmwlm.log2</code> . xmwlm cycles between these two files after a file reaches the maximum allowable size.

Session Recovery by xmwlm

If the xmwlm agent is terminated and restarted, xmwlm examines the recording files in the appropriate `/etc/perf` subdirectory or in the directory specified by the `-d` flag. If a recording file exists with the current date, xmwlm appends data to this file and continues to write to the recording file. Otherwise, a new recording file is created.

Location

`/usr/bin/xmwlm`

Files

Item	Description
<code>/usr/bin/xmwlm</code>	Contains the xmwlm agent. The agent is part of the <code>peragent.tools</code> fileset.

xmodem Command

Purpose

Transfers files with the **xmodem** protocol, detecting data transmission errors during asynchronous transmission.

Syntax

xmodem { `-s` | `-r` } *FileName*

Description

The **xmodem** shell command is used with the Asynchronous Terminal Emulation (ATE) program to transfer a file, designated by the *FileName* parameter, using the **xmodem** protocol.

The **xmodem** protocol is an 8-bit transfer protocol to detect data transmission errors and retransmit the data. The workstation sending data waits until the remote system sends a signal indicating it is ready to receive data.

After the receiving system get data, it returns an acknowledgment to the sending system. In the ATE program the receiving system times out if data is not received within 90 seconds after the file transfer is initiated.

Sending and receiving with the **xmodem** command are complementary operations. One system must be set to send while the other is set to receive. Use the **xmodem** command on the remote system in combination with the **send** subcommand or the **receive** subcommand from the ATE Connected Main Menu on the local system.

To interrupt an **xmodem** file transfer, press the Ctrl-X key sequence.

Note:

1. The DOS operating system terminates each line in an ASCII file with a newline character and a carriage return (Ctrl-M) character. UNIX terminates each line in an ASCII file only with a newline character. The carriage return characters are preserved when a DOS file is transferred to AIX. The vi text editor can be used to remove spurious Ctrl-M characters using the subcommand

```
:%s/<Ctrl-V><Ctrl-M>//
```

where <Ctrl-V> and <Ctrl-M> each represent a single control character that is typed. However, since Ctrl-V is the default ATE MAINMENU_KEY, the ATE defaults must be altered in order to issue the vi subcommand while logged in via ATE.

2. The xmodem file transfer process adds Ctrl-Z characters to the last packet transferred to make the packet 128 bytes long. Most files transferred will, therefore, have Ctrl-Z characters appended to the end. The DOS operating system terminates an ASCII file with a Ctrl-Z character. Every file transferred from DOS to AIX will, therefore, end with at least one Ctrl-Z character. These extra Ctrl-Z characters can be removed with the vi text editor.

Flags

Item	Description

-r	Receives data from the local workstation.
-s	Sends data to the local workstation.

Examples

Sending a File with the xmodem Protocol

To send the file `myfile` with the **xmodem** protocol, use the **ate** command and the **connect** or **directory** subcommand to establish a connection to the remote system.

1. After logging in to the remote system and before pressing the MAINMENU_KEY (usually the Ctrl-Vkey sequence) to return to ATE on the local system, enter:

```
xmodem -r myfile
```

at the shell command line. The **xmodem** protocol starts receive mode on the remote system.

2. Press the MAINMENU_KEY to return to ATE on the local system.

The ATE Connected Main Menu displays.

3. Enter the **send** subcommand at the prompt on the ATE Connected Main Menu:

```
s myfile
```

The **send** subcommand instructs the local system to send `myfile` to the remote system. After transferring the file, the ATE Connected Main Menu displays.

Receiving a File with the xmodem Protocol

Receive the file `infile` from a remote system using **xmodem** protocol with the **ate** command and the **connect** or **directory** subcommand establishing a connection to the remote system.

1. After logging in to the remote system and before pressing the MAINMENU_KEY (usually the Ctrl-V key sequence) to return to ATE on the local system, enter:

```
xmodem -s infile
```

at the shell command line. The **xmodem** protocol starts, in send mode, on the remote system.

2. Press the MAINMENU_KEY to return to ATE on the local system.

The ATE Connected Main Menu displays.

3. Enter the **receive** subcommand at the prompt on the ATE Connected Main Menu:

```
r infile
```

The **receive** subcommand instructs the local system to receive `infile` from the remote system. After transferring the file, the ATE Connected Main Menu displays.

File

Item	Description
ate.def	Contains ATE default values.

xmodmap Command

Purpose

Modifies keymaps in the X Server.

Syntax

```
xmodmap [ -display Display ] [ -e Expression ] [ -grammar ] [ -help ] [ -n ] [ -pk ] [ -pke ] [ -pm ] [ -pp ]  
[ -quiet ] [ -verbose ] [ FileName ]
```

Description

The **xmodmap** command edits and displays the keyboard modifier map and keymap table that client applications use to convert event keycodes into key symbols. It is usually run from the session startup script to configure the keyboard according to the personal tastes of the user.

Every time a keycode expression is evaluated, the server generates a **MappingNotify** event on every client. All of the changes should be batched together and done at one time. Clients that receive keyboard input and ignore **MappingNotify** events will not notice any changes made to keyboard mappings.

The *FileName* parameter specifies a file containing the **xmodmap** command expressions to be run. This file is usually kept in the home directory of the user with a name like **.xmodmaprc**. If no file is specified, input is taken from **stdin**.

The **xmodmap** command program reads a list of expressions and parses them all before attempting to run any of them. This makes it possible to refer to key symbols that are being naturally redefined without having to worry as much about name conflicts.

Item	Description
add	The key symbol names are evaluated as the line is read. This permits you to remove keys from a modifier without worrying about whether they were reassigned.

Item	Description
add <i>ModifierName</i> = <i>KeySymbolName</i> ...	Adds the given key symbols to the indicated modifier map. The key symbol names are evaluated after all input expressions are read to make it easy to write expressions to swap keys.
clear <i>ModifierName</i>	Removes all entries in the modifier map for the given modifier, where the valid names are Shift , Lock , Control , Mod1 , Mod2 , Mod3 , Mod4 , and Mod5 (case does not matter in modifier names, although it does matter for all other names). For example, clear Lock removes all keys bound to the shift lock modifier.
keycode <i>Number</i> = <i>KeySymbolName</i> ...	Assigns the list of key symbols to the indicated keycode (which can be specified in decimal, hex, or octal and be determined by running the xev program in the /usr/lpp/X11/Xamples/demos directory). Usually only one key symbol is assigned to a given code.
keysym <i>KeySymbolName</i> = <i>KeySymbolName</i> ...	The <i>KeySymbolName</i> on the left hand side is translated into matching keycodes used to perform the corresponding set of keycode expressions. The list of keysym names can be found in the keysym database /usr/lib/X11/XKeysymDB or the header file X11/keysymdef.h (without the XK_ prefix). Note that if the same keysym is bound to multiple keys, the expression is run for each matching keycode.
pointer = default	Sets the pointer map back to its default settings (such as, button 1 generates a code of 1, button 2 generates a 2, and so forth).
pointer = <i>Button1 Button2 Button3</i> ...	Sets the pointer map to contain the indicated button codes. The list always starts with the first physical button.
remove <i>ModifierName</i> = <i>KeySymbolName</i> ...	Removes all keys containing the given keysyms from the indicated modifier map. Unlike add , the keysym names are evaluated as the line is read in. This allows for the removal of keys from a modifier without having to worry about whether or not they have been reassigned.

Lines that begin with an ! (exclamation point) are taken as comments.

If you want to change the binding of a modifier key, you must also remove it from the appropriate modifier map.

Flags

Item	Description
-display <i>Display</i>	Specifies the host and display to use.
-e <i>Expression</i>	Specifies an expression to be run. You can specify any number of expressions from the command line.
-grammar	Prints a help message describing the expression grammar used in files and with the -e Expressions flag prints to standard error.
-help	Prints a brief description of the command line arguments to standard error. This is done whenever an unhandled argument is given to the xmodmap command.

Item	Description
-n	Indicates that the xmodmap command should not change the mappings, but should display what it would do when given this flag.
-pk	Indicates that the current keymap table should print on the standard output.
-pke	Indicates that the current keymap table should be printed on the standard output in the form of expressions that can be fed back to xmodmap . This flag is specific to X11R5.
-pm	Indicates that the current modifier map should print on the standard output.
-pp	Indicates that the current pointer map should print on the standard output.
-quiet	Turns off the verbose logging. This is the default.
-verbose	Indicates that the xmodmap command should print logging information as it parses its input.

Examples

1. The following command reverses the button codes that get generated so that the primary button is pressed using the index finger of the left hand on a 3 button pointer:

```
xmodmap -e "pointer = 1 2 3 4 5"
```

2. The following command attaches meta to the multi-language key (sometimes labeled Compose Character). It also takes advantage of the fact that applications that need a Meta key simply need to get the keycode and do not require the key symbol to be in the first column of the keymap table. This means that applications that are looking for a Multi_key (including the default modifier map) will not notice any change.

```
keysym Multi_key = Multi_key Meta_L
```

3. To automatically generate less than and greater than characters when the comma and period keys are shifted, reset the bindings for the comma and period with the following scripts:

```
!
! make shift-, be < and shift-. be >
!
keysym comma = comma less
keysym period = period greater
```

4. To swap the location of the Control and Shift Lock keys, use the following script:

```
!
! Swap Caps_Lock and Control_L
!
remove Lock = Caps_Lock
remove Control = Control_L
keysym Control_L = Caps_Lock
keysym Caps_Lock = Control_L
add Lock = Caps_Lock
add Control = Control_L
```

xmpeek Command

Purpose

The **xmpeek** command allows you to query any host about the status of its **xmtopas** daemon.

Syntax

```
xmpeek [ -a | -l ] [ hostname ]
```

Description

The **xmpeek** command is used to list down the data consumers that currently have instruments (stat sets) defined with the **xmtopas** daemon, and list down all known data consumers by the **xmtopas** daemon . The **xmpeek** command is also used to print down all the available SPMI statistics for any given host.

Flags

Item	Description
-a	If this flag is specified, one line is listed for each data consumer known by the daemon. If this flag is not used, only data consumers that currently have instruments (stat sets) defined with the daemon are listed. This flag is optional.
-l	Lists all the available statistics of the remote host. This flag is optional.
hostname	If the hostname is specified, the daemon on the named host is asked. If no host name is specified, the daemon on the local host is asked.

Examples

The following is an example of the output from the **xmpeek** program:

```
Statistics for xmtopas daemon on *** birte ***
Instruments currently defined:    1
Instruments currently active:    1
Remote monitors currently known: 2
--Instruments-- Values  Packets
Defined Active  Active  Sent      Internet Protocol
-----
1           1       16     3,344    Address      Port Hostname
-----
1           1       16     3,344    129.49.115.208 3885 xtra
```

Output from **xmpeek** can take two forms.

The first form is a line that informs you that the **xmtopas** daemon is not feeding any data-consumer programs. This form is used if no statsets are defined with the daemon and no command flags are supplied.

The second form includes at least as much as is shown in the preceding example, except that the single detail line for the data consumer on host **xtra** is shown only if either the **-a** flag is used or if the data consumer has at least one instrument (statset) defined with the daemon. Note that **xmpeek** itself appears as a data consumer because it uses the Remote Statistics Interface (RSI) API to contact the daemon. Therefore, the output always shows at least one known monitor.

In the fixed output, first the name of the host where the daemon is running is shown. Then follows three lines giving the totals for current status of the daemon. In the above example, you can see that only one instrument is defined and that it's active. You can also see that two data consumers are known by the daemon, but that only one of them has an instrument defined with the daemon in **birte**. Obviously, this output was produced without the **-a** flag.

An example of more activity is shown in the following sample output from **xmpeek**. The output is produced with the command:

```
xmpeek -a birte
```

Notice that some detail lines show zero instruments defined. Such lines indicate that an **are_you_there** message was received from the data consumer but that no states were ever defined or that any previously defined states were erased.

```
Statistics for smeared daemon on *** birte ***
Instruments currently defined:    16
Instruments currently active:    14
Remote monitors currently known: 6
--Instruments-- Values Packets  Internet Protocol
Defined Active  Active  Sent      Address      Port  Hostname
8             8       35    10,232    129.49.115.203 4184  birte
6             4       28     8,322    129.49.246.14 3211  umbra
0             0        0        0    129.49.115.208 3861  xtra
1             1       16     3,332    129.49.246.14 3219  umbra
0             0        0        0    129.49.115.203 4209  birte
1             1       16     422      129.49.115.208 3874  xtra
-----
16           14       95    22,308
```

Notice that the same host name may appear more than once. This is because every running copy of **xmperf** and every other active data-consumer program is counted and treated as a separate data consumer, each identified by the port number used for UDP packets as shown in the **xmpeek** output.

The second detail shows that one particular monitor on host **umbra** has six instruments defined but only four active. This would happen if a remote **xmperf** console has been opened but is now closed. When you close an **xmperf** console, it stays in the Monitor menu of the **xmperf** main window and the definition of the instruments of that console remains in the tables of the data-supplier daemon but the instruments are not active.

xmscheck Command

Purpose

The **xmscheck** command is available to pre-parse a recording configuration file and to determine how the running **xmtopas** daemon is configured for recording.

Syntax

```
xmscheck [ file_name ]
```

Description

When the **xmtopas** command is started with the command line argument **-v**, its recording configuration file parser writes the result of the parsing to the log file. The output includes a copy of all lines in the recording configuration file, any error messages, and a map of the time scale with indication of when recording starts and stops.

Although this is useful to document what is read from the recording configuration file, it is not a useful tool for debugging of a new or modified recording configuration file. Therefore, the program **xmscheck** command is available to preparse a recording configuration file before you move it to the **/etc/perf** directory, where the **xmtopas** command looks for the recording configuration file.

When **xmscheck** command is started without any command line argument, it parses the file **/etc/perf/xmservd.cf**. This way, you can determine how the running daemon is configured for recording. If a file name is specified on the command line, that file is parsed.

Output from the **xmscheck** command goes to stdout. The parsing is done by the exact same module that does the parsing in the **xmtopas** command. That module is linked in as part of both programs. The parsing checks that all statistics specified are valid and prints the time scale for starting and stopping recording in the form of a “time table.”

Item	Description

- ## Examples

[illegible]

xmtopas Command

The **xmtopas** daemon acts as a recording facility and is controlled through the `xmtopas.cf` configuration file. This daemon simultaneously provides near real-time network-based data monitoring and local recording on a given node.

```
xmtopas [-v] [-b UDP_buffer_size] [-i min_remote_interval] [-l remove_consumer_timeout]
[-m supplier_timeout] [-p trace_level] [-s max_logfile_size] [-t keep_alive_limit] [-x
xmtopas execution priority]
```

► You can change the **xmtpas** daemon by using the System Management Interface Tool (SMIT) or use System Resource Controller (SRC) to edit the `/etc/inetd.conf` file. To start the **xmtpas** daemon, uncomment the **xmtpas** daemon entry in the `/etc/inetd.conf` file. You can send inputs to the **inetd** daemon by specifying the inputs in the `/etc/inetd.conf` file. After you update the `/etc/inetd.conf`

file, run the **refresh -s inetd** command or the **kill -1 InetdPID** command for the **inetd** daemon to receive the configuration file changes. [L](#)

Note:

1. If you modify the **xmtopas** entry in the **inetd.conf** file on the Virtual IO Server (VIOS), the new entry is added after migration. You must replace the **xmtopas** entry to a default type, such as **xmtopas -p3**, before migration.
2. For any dynamic configuration changes to the logical partition, the **xmtopas** must be restarted to reflect the changes.

Flags

Item	Description
-v	Causes parsing information for the xmtopas recording configuration file to be written to the xmtopas log file.
-b <i>UDP_buffer_size</i>	Defines the size of the buffer used by the daemon to send and receive UDP packets. The buffer size must be specified in bytes and can be from 4,096 to 16,384 bytes. The buffer size determines the maximum number of data values that can be sent in one data_feed packet. The default buffer size is 4096 bytes, which allows for up to 124 data values in one packet.
-i <i>min_remote_interval</i>	Defines the minimum interval in milliseconds with which data feeds can be sent. Default is 500 milliseconds. A value between 100 and 5,000 milliseconds can be specified. Any value specified is rounded to a multiple of 100 milliseconds. Whichever minimum remote interval is specified causes all requests for data feeds to be rounded to a multiple of this value.
-l <i>remove_consumer_timeout</i>	Sets the time_to_live after feeding of statistics data has ceased as described in section Life and Death of xmtopas . Must be followed by a number of minutes. A value of 0 (zero) minutes causes the daemon to stay alive forever. The default time_to_live is 15 minutes.
-m <i>supplier_timeout</i>	When a dynamic data-supplier is active, this value sets the number of seconds of inactivity from the DDS before the SPMI assumes the DDS is dead. When the timeout value is exceeded, the SiShGoAway flag is set in the shared memory area and the SPMI disconnects from the area. If this flag is not given, the timeout period is set to 90 seconds. The size of the timeout period is kept in the SPMI common shared memory area. The value stored is the maximum value requested by any data consumer program, including the xmtopas command.
-p <i>trace_level</i>	Sets the trace level, which determines the types of events written to the /var/perf/xmtopas.log1 log file or the /var/perf/xmtopas.log2 log file. This flag must be followed by a digit from 0 to 9, with 9 being the most detailed trace level. Default trace level is 0 (zero), which disables tracing and logging of events but logs error messages.

Item	Description
-s <i>max_logfile_size</i>	Specifies the approximate maximum size of the log files. At least every time_to_live minutes, it is checked if the currently active log file is bigger than the <i>max_logfile_size</i> value. If so, the current log file is closed and logging continues to the alternate log file, which is first reset to zero length. The two log files are /var/perf/xmtopas.log1 and /var/perf/xmtopas.log2 . Default maximum file size is 100,000 bytes. You cannot make the <i>max_logfile_size</i> value smaller than 5,000 or larger than 10,000,000 bytes.
-t <i>keep_alive_limit</i>	Sets the <i>keep_alive_limit</i> value must be followed by a number of seconds from 60 to 900 (1 to 15 minutes). The default is 300 seconds (5 minutes).
-x <i>xmtopas_execution_priority</i>	Sets the execution priority of the xmtopas command. Use this option if the default execution priority of the xmtopas command is unsuitable in your environment. Generally, the daemon should be given as high execution priority as possible (a smaller number gives a higher execution priority). On systems other than IBM RS/6000 systems, the -x flag is used to set the nice priority of the xmtopas command. The nice priority is a value from -20 to 19. The default is -20.

Files used by the xmtopas command

You can specify the following entries in the **xmtopas.res** file:

```
docec:<arguments>
docluster:cluster=<cluster configuration file>
```

Example

```
docec: availmem=5 unavailmem=2
docluster: cluster=/etc/perf/xmtopasagg.cf
```

The following new fields are added to the **docec** entry in the **xmtopas.res** file to get the Hardware Management Console (HMC) details:

```
managedsys=[Managed system name under which this partition is configured]
hmc=[HMC name under which this partition is configured]
```

If the HMC platform cannot be configured for automatic queries, the global data fields that are not available to the local partition can be set by using the following options:

```
availmem = [Total amount of memory allocated to all partitions, in GB]
unavailmem = [Total amount of memory unallocated from the HMC, in GB]
availprocessor = [Total number of physical processors allocated for all partitions]
unavailprocessor = [Total number of physical processors unallocated from the HMC]
poolsize = [Defined Pool Size required if HMC Processor Utilization Authority restricts access]
partitions = [Number of partitions defined on the HMC]
reconfig = [Number of seconds between checking for HMC configuration changes.
Allowed values are 30, 60, 90, 120, 180, 240, 300 seconds.
The default is 60 seconds.]
```

Example

```
docec: hmc=hmc.mac.in.ibm.com managedsys=cec1
```

xpr Command

Purpose

Formats a window dump file for output to a printer.

Syntax

```
xpr [ -append FileName [ -noff ] ] [ -output FileName ] [ -landscape | -portrait ] [ -compact ]  
[ -cutoff Level ] [ -density Dpi ] [ -gray { 2 | 3 | 4 } ] [ -header String ] [ -height Inches ] [ -left Inches ]  
[ -noposition ] [ -plane PlaneNumber ] [ -psfig ] [ -report ] [ -rv ] [ -scale Scale ] [ -split Number ]  
[ -top Inches ] [ -trailer String ] [ -width Inches ] [ -device Device ] [ ImageFile ]
```

Description

The **xpr** command uses a window dump file produced by the **xwd** utility as input and formats the dump file for output on all printers supported by the hardware. If you do not specify a file argument, the **xpr** command uses standard input. By default, the **xpr** command prints the largest possible representation of the window on the output page.

The **xpr** command options allow you to add headers and trailers, specify margins, adjust the scale and orientation, and append multiple window dumps to a single output file. Output is to standard output unless the **-output** flag is specified.

Flags

Item	Description
-append <i>FileName</i>	Specifies a file name previously produced by the xpr command to which the window is to append. (This flag is not supported on PostScript printers.)
-compact	Uses simple run-length encoding for compact representation of windows with many white pixels. This flag compresses white space but not black space, so it is not useful for reverse-video windows. (This flag supports PostScript, LIPS II+, and LIPSIII output only.)
-cutoff <i>Level</i>	Changes the intensity level where colors are mapped to black or white for monochrome output on a LaserJet printer. The <i>Level</i> variable is expressed as a percentage of full brightness. Fractions are acceptable.

Item	Description
-device <i>Device</i>	Specifies the device on which the file prints. The xpr command supports the following printers: 3812 or pp IBM PP3812 4207 Proprinter 5201 IBM Quietwriter 1 model 2 5202 IBM Quietwriter 2 jprinter IBM Japanese Printer (Japanese data stream) ljet HP LaserJet and IBM Laser Printer ps PostScript printers (this is the default) lips2 Canon LaserShot LIPS II+ mode lips3 Canon LaserShot LIPS III mode
-density <i>Dpi</i>	Indicates the dots-per-inch (dpi) density that the HP printer uses. 300 dpi is the default. Allowable densities are 300, 150, 100, and 75 dpi.
-gray <i>Number</i>	Specifies gray-scale conversion to a color image, rather than mapping to a black-and-white image. The <i>Number</i> variable must be one of the following: 2 2 x 2 conversion 3 3 x 3 conversion 4 4 x 4 conversion This conversion doubles, triples, or quadruples, respectively, the effective width and height of the image. Note: This option is valid only for PostScript printers.
-header <i>String</i>	Specifies a header string to print above the window.
-height <i>Inches</i>	Specifies the maximum height of the page.
<i>ImageFile</i>	Contains the captured bitmap of the image. If you do not specify the <i>ImageFile</i> parameter, the xpr command uses standard input.
-landscape	Forces the window to print in landscape mode. (The display is laid out with the windows being wider than they are high.) By default, a window prints so that its longest side follows the long side of the paper.
-left <i>Inches</i>	Specifies the left margin in inches. Fractions are acceptable. By default, this flag prints the window on the center of the page.
-noff	When specified in conjunction with the -append flag, the window is displayed on the same page as the previous window. (This flag is not supported on PostScript printers.)

Item	Description
-noposition	Causes the header, trailer, and image positioning command generation to be bypassed for the LaserJet printer.
-output <i>FileName</i>	Specifies an output file name. If you do not specify this option, the xpr command uses standard output.
-plane <i>PlaneNumber</i>	Specifies which bit plane to use in an image. The default uses the entire image and maps values into black and white based on color intensities. This option is not supported for the LaserJet printer.
-portrait	Forces the window to print in portrait mode. (The display is laid out with the windows being higher than they are wide.) By default, a window prints so that its longest side follows the long side of the paper.
-psfig	Suppresses translation of the PostScript picture to the center of the page.
-report	Prints out statistics to standard error about the window <i>ImageFile</i> parameter.
-rv	Forces the window to print in reverse video.
-scale <i>Scale</i>	Affects the size of the window on the page. PostScript printers are able to translate each bit in a window pixel map into a grid of a specified size. For example, each bit might translate into a 3 x 3 grid. To specify a 3 x 3 grid, enter -scale 3 . By default, a window prints with the largest scale that fits on the page for the specified orientation. If you do not specify a device, the aspect ratio can vary.
-split <i>Number</i>	Splits a window into several pages. This might be necessary for very large windows that would otherwise cause the printer to overload and print the page in an obscure manner. (This flag is not supported on PostScript or HP Laserjet printers.)
-top <i>Inches</i>	Specifies the top margin for the window in inches. Fractions are acceptable. By default, this flag prints the window on the center of the page.
-trailer <i>String</i>	Specifies a trailer string to print below the window.
-width <i>Inches</i>	Specifies the maximum width of the page.

Note: The 4207, 5201, and 5202 printers' images must be recorded by the **xwd** utility in XYPixmap or XYBitmap format. XYPixmap images are converted into bitmaps using a thresholding algorithm. For the HP Laserjet printer, multiplane images must be recorded in ZPixmap format. Single plane images may be either XYPixmap, XYBitmap, or ZPixmap formats.

xpreview Command

Purpose

Displays troff files on an X display.

Syntax

```
xpreview [ -BackingStore BackingStoreType ] [ -page Number ] [ ToolKitFlag ... ] { File | - }
```

Description

The **xpreview** command is an AIXwindows 2.1- and Motif2.1-based application that displays output from the **troff** command on an AIXwindows display. The **troff** command output file must be prepared for the devX100 device.

The user interface contains the standard AIXwindows interface controls for calling the root menu, iconifying the window, and setting the window to full screen size. The interface also includes a main window with a scrollable display area for text. Use the pushbuttons for Next, Previous, Goto Page, Print Page, Print File, and Newfile to manipulate the viewing document.

Mouse button three actuates a popup menu for configuring print capabilities. The menu includes an option to set the command line and another to select a printer queue. The command line dialog box expects command line input through the **troff** command. For example,

```
pic -Tibm3816 troff-input-file |tbl|troff -mm -Tibm3816
```

is an acceptable command line. The printer queue option displays a list of configured printer queues. If this option is not selected, the **xpreview** command uses the system-defined default queue.

When you are previewing an input file, the Print Page and Print File buttons require command line input. Note that once a printer queue is selected, it remains selected for the duration of the viewing session, or until an alternate printer queue is selected.

Fonts supported for the devX100 device in European locales are:

- Times New Roman in normal, italic, and bold
- Courier in normal and bold
- Helvetica in normal and bold
- Symbol

The **xpreview** command supports the following font sizes: 8, 10, 14, 18, 24, 30, and 36.

The **xpreview** command does not display files resulting from the **troff** command constructed for a device other than those described in this document.

To preview a file on a certain device, the **xpreview** command requires the fonts found in the following directories:

- **/usr/lib/X11/fonts** directory for files formatted for font files other than Japanese
- **/usr/lib/X11/fonts/JP** for Japanese font files

Multibyte Support

The **xpreview** command supports multibyte locales. Also, to display Japanese characters, Japanese 16-dot fonts (part of the Japanese BSL package) and 24- and 32-dot fonts (part of the AIXwindows font package) must be installed. To display Korean characters, Korean fonts (part of the Korean BSL package) must be installed.

Japanese support currently includes the following font sets:

- In 16-dot: RomanKn12, Kanji12, and IBM_JPN12
- In 24-dot: RomanKn17, Kanji17, and IBM_JPN17
- In 32-dot: RomanKn23, Kanji23, and IBM_JPN23, or RomanKn23G, Kanji23G, and IBM_JPN23G

Korean support currently includes the following font sets:

- In 16-dot, EnglHg16 and Hangul16
- In 24-dot, EnglHg24 and Hangul24

Flags

The **xpreview** command accepts the standard **X** Toolkit command line flags, as well as the following flags:

Item	Description
-	Requires input to be read from standard input.
-help	Indicates that a brief summary of the allowed command line flags should be printed.
-BackingStore <i>BackingStoreType</i>	The -BackingStore flag causes the server to save the window contents so that when it is scrolled around the viewport, the window is painted from contents saved in server backing store. Redisplay of the drawing window can take up to a second or so. The <i>BackingStoreType</i> parameter can have one of the following values: Always, WhenMapped or NotUseful. Tip: Enter a space between the -BackingStore flag and its <i>BackingStoreType</i> parameter. Requirement: Use of this flag requires that the server be started with backing store enabled.
-page <i>Number</i>	Specifies the page number of the document to be first displayed.
<i>ToolKitFlag</i>	The following standard X Toolkit flags are commonly used with the xpreview command: -bg <i>Color</i> Specifies the color to use for the background of the window. The default is white. -bg <i>Color</i> Specifies the color to use for the background of the window. The default is white. -fg <i>Color</i> Specifies the color to use for displaying text. The default is black. -geometry <i>Geometry</i> Specifies the preferred size and position of the window. -display <i>Host:Display</i> Specifies the X server to contact. -xrm <i>ResourceString</i> Specifies a resource string to be used.
<i>File</i>	Specifies the file to be printed.

Examples

1. To build files output by the **troff** command into files that are suitable for use with the **xpreview** command, enter the following commands:

```
troff-TX100 troff-input | xpreview
pic -TX100 pic-troff-input | tbl | troff -man -TX100 | xpreview
```

2. To build files output by the **troff** command into files that are suitable for use with the Japanese language version of the **xpreview** command, enter the following commands:

```
LANG=ja_JP
troff -TX100 troff-input | xpreview -
```

```
pic -TX100 pic-troff-input | tbl | troff -man -TX100 \
| xpreview -
```

Files

Item	Description
/usr/lib/X11/app-defaults/XPreview	Contains user-configurable applications defaults file.
/usr/lib/X11/Ja_JP/app-defaults/XPreview	Contains user-configurable applications default file for the Japanese (IBM-943) locale.
/usr/lib/X11/ja_JP/app-defaults/XPreview	Contains user-configurable applications default file for the Japanese (IBM-eucJP) locale.
/usr/lib/X11/ko_KR/app-defaults/XPreview	Contains user-configurable applications default file for the Korean locale.
/usr/lib/X11/zh_TW/app-defaults/XPreview	Contains user-configurable applications default file for the Traditional Chinese locale.
/usr/lib/font/devX100	Contains troff fonts for devX100 devices.
/usr/lib/X11/fonts	Contains X fonts for 100 dpi devices.
/usr/lib/X11/fonts/JP	Contains X fonts for multi-byte characters.
/usr/lib/X11/fonts/JP	Contains X fonts for Japanese characters.

xprofiler Command

Purpose

Starts Xprofiler, a GUI-based AIX performance profiling tool.

Syntax

```
xprofiler [ program ] [ -b ] [ -s ] [ -z ] [ -a path ] [ -c file ] [ -L pathname ] [ [ -e function ] ... ] [ [ -E function ] ... ] [ [ -f function ] ... ] [ [ -F function ] ... ] [ -disp_max number_of_functions ] [ [ gmon.out ] ... ]
xprofiler -h | -help
```

Description

The `xprofiler` command invokes Xprofiler, a GUI-based AIX performance profiling tool. Xprofiler is used to analyze the performance of both serial and parallel applications. Xprofiler uses data collected by the `-pg` compiling option and presents a graphical representation of the functions in the application in addition to providing textual data in several report windows. These presentation formats are intended to identify the functions which are most processor-intensive.

Flags

Item	Description
-a	To specify an alternate search path or paths for library files and source code files. If more than one path is specified, the paths must be embraced by "," and each path should be separated by either ":" or space.
-b	Suppresses the printing of the field descriptions for the Flat Profile, Call Graph Profile, and Function Index reports when they are written to a file with the Save As option of the File menu.
-c	Loads a configuration file that contains information to be used to determine which functions will be displayed when Xprofiler is brought up.
-disp_max	Sets the number of function boxes that Xprofiler initially displays in the function call tree. The value supplied with this flag can be any integer between 0 and 5,000. Xprofiler displays the function boxes for the most processor-intensive functions through the number you specify. For instance, if you specify 50, Xprofiler displays the function boxes for the 50 functions in your program that consume the most processor. After this, you can change the number of function boxes that are displayed via the Filter menu options. This flag has no effect on the content of any of the Xprofiler reports.
-e	De-emphasizes the general appearance of the function box or boxes for the specified functions in the function call tree, and limits the number of entries for these function in the Call Graph Profile report. This also applies to the specified function's descendants, as long as they have not been called by non-specified functions. In the function call tree, the function boxes for the specified functions appear greyed-out. Its size and the content of the label remain the same. This also applies to descendant functions, as long as they have not been called by non-specified functions. In the Call Graph Profile report, an entry for the specified function only appears where it is a child of another function, or as a parent of a function that also has at least one non-specified function as its parent. The information for this entry remains unchanged. Entries for descendants of the specified function do not appear unless they have been called by at least one non-specified function in the program.
-E	Changes the general appearance and label information of the function box or boxes for the specified functions in the function call tree. Also limits the number of entries for these functions in the Call Graph Profile report, and changes the processor data associated with them. These results also apply to the specified function's descendants, as long as they have not been called by non-specified functions in the program. In the function call tree, the function box for the specified function appears greyed-out, and its size and shape also changes so that it appears as a square of the smallest allowable size. In addition, the processor time shown in the function box label, appears as 0 (zero). The same applies to function boxes for descendant functions, as long as they have not been called by non-specified functions. This option also causes the processor time spent by the specified function to be deducted from the left side processor total in the label of the function box for each of the specified function's ancestors. In the Call Graph Profile report, an entry for the specified function only appears where it is a child of another function, or as a parent of a function that also has at least one non-specified function as its parent. When this is the case, the time in the self and descendants columns for this entry is set to 0 (zero). In addition, the amount of time that was in the descendants column for the specified function is subtracted from the time listed under the descendants column for the profiled function. As a result, be aware that the value listed in the % time column for most profiled functions in this report will change.

Item	Description
-f	De-emphasizes the general appearance of all function boxes in the function call tree, except for that of the specified function(s) and its descendant(s). In addition, the number of entries in the Call Graph Profile report for the non-specified functions and non-descendant functions is limited. The -f flag overrides the -e flag. In the function call tree, all function boxes except for that of the specified function(s) and its descendant(s) appear greyed-out. The size of these boxes and the content of their labels remain the same. For the specified function(s), and its descendants, the appearance of the function boxes and labels remain the same. In the Call Graph Profile report, an entry for a non-specified or non-descendant function only appears where it is a parent or child of a specified function or one of its descendants. All information for this entry remains the same.
-F	Changes the general appearance and label information of all function boxes in the function call tree except for that of the specified function(s) and its descendants. In addition, the number of entries in the Call Graph Profile report for the non-specified and non-descendant functions is limited, and the processor data associated with them is changed. The -F flag overrides the -E flag. In the function call tree, the function box for the specified function appears greyed-out, and its size and shape also changes so that it appears as a square of the smallest allowable size. In addition, the processor time shown in the function box label, appears as 0 (zero). In the Call Graph Profile report, an entry for a non-specified or non-descendant function only appears where it is a parent or child of a specified function or one of its descendants. The time in the self and descendants columns for this entry is set to 0 (zero). When this is the case, the time in the self and descendants columns for this entry is set to 0 (zero). As a result, be aware that the value listed in the % time column for most profiled functions in this report will change.
-h -help	Writes the Xprofiler usage to STDERR and then exits. The information includes xprofiler command line syntax and a description of Xprofiler runtime options.
-L	Uses an alternate path name for locating shared libraries. If you plan to specify multiple paths, use the Set File Search Paths option of the File menu on the Xprofiler GUI.
-s	If multiple gmon.out files are specified when Xprofiler is started, produces the gmon.sum profile data file. The gmon.sum file represents the sum of the profile information in all the specified profile files. Note that if you specify a single gmon.out file, the gmon.sum file contains the same data as the gmon.out file.
-z	Includes functions that have both zero processor usage and no call counts in the Flat Profile, Call Graph Profile, and Function Index reports. A function will not have a call count if the file that contains its definition was not compiled with the -pg option, which is common with system library files.

Example

To use xprofiler, you must first compile your program (for example, foo.c) with -pg:

```
xlc -pg -o foo foo.c
```

1. When the program foo is executed, one gmon.out file will be generated for each processor involved in the execution. To invoke xprofiler, enter:

```
xprofiler foo [[gmon.out]...]
```

Files

Item	Description
/usr/lib/X11/app-defaults/Xprofiler	Location of the xprofiler command.

xrdb Command

Purpose

X Server resource database utilities.

Syntax

```
xrdb [ -display Display ] [ -help ] [ -quiet ] [ -retain ] [ -cpp FileName | -nocpp ] [ -D Name=Value ] [ -I Directory ] [ -U Name ] [ -all | -global | -screen | -screens ] [ -n ] [ -edit FileName | -backup String ] [ -merge FileName ] [ -load FileName ] [ -query | -remove | symbols ] [ -override ]
```

Description

The **xrdb** command gets or sets the contents of the RESOURCE_MANAGER property on the root window of screen 0 or the SCREEN_RESOURCES property on the root window of any or all screens, or everything combined. You normally run this program from your X startup file.

Most X clients use the RESOURCE_MANAGER and SCREEN_RESOURCES properties to get user preferences about color, fonts, and so on for applications. Having this information in the server (where it is available to all clients) instead of on disk solves the problem in previous versions of X that required you to maintain *defaults* files on every machine that you might use. It also allows for dynamic changing of defaults without editing files.

The RESOURCE_MANAGER property specifies resources that apply to all screens of the display. The SCREEN_RESOURCES property on each screen specifies additional (or overriding) resources to be used for that screen. (When there is only one screen, SCREEN_RESOURCES is normally not used; all resources are placed in the RESOURCE_MANAGER property.)

For compatibility, if there is no RESOURCE_MANAGER property defined (either because the **xrdb** command was not run or if the property was removed), the resource manager looks for a file called **.Xdefaults** in your home directory.

The file name (or the standard input if - or no file name is given) is optionally passed through the C preprocessor with the following symbols defined, based on the capabilities of the server being used:

Item	Description
SERVERHOST = <i>Hostname</i>	Specifies the hostname portion of the display to which you are connected.
SRVR _{<i>name</i>}	Turns the SERVERHOST hostname string into a legal identifier. For example <i>my-dpy.lcs.mit.edu</i> becomes <i>SRVR_my_dpy_lcs_mit_edu</i> .
HOST = <i>Hostname</i>	Specifies the hostname portion of the display to which you are connected.
DISPLAY_NUM = <i>num</i>	Specifies the number of the display on the server host.
CLIENTHOST = <i>Hostname</i>	Specifies the name of the host on which xrdb is running.
CLNT _{<i>name</i>}	Turns the CLIENTHOST hostname string into a legal identifier. For example <i>expo.lcs.mit.edu</i> becomes <i>CLNT_expo_lcs_mit_edu</i> .
WIDTH = <i>Number</i>	Specifies the width of the default screen in pixels.
HEIGHT = <i>Number</i>	Specifies the height of the default screen in pixels.
X_RESOLUTION = <i>Number</i>	Specifies the x resolution of the default screen in pixels per meter.

Item	Description
Y_RESOLUTION = <i>Number</i>	Specifies the y resolution of the default screen in pixels per meter.
PLANES = <i>Number</i>	Specifies the number of bit planes (the depth) of the root window of the default screen.
RELEASE = <i>Number</i>	Specifies the vendor release number for the server. The interpretation of this number varies depending on VENDOR .
REVISION = <i>Number</i>	Specifies the X protocol minor version supported by this server (currently 0).
VERSION = <i>Number</i>	Specifies the X protocol major version supported by this server (should always be 11).
VENDOR = <i>Vendor</i>	A string specifying the vendor of the server.
VNDR _name	Turns the VENDOR name string into a legal identifier. For example MIT X Consortium becomes VNDR_MIT_X_Consortium.
EXT _name	Turns each extension string into a legal identifier. A symbol is defined for each protocol extension supported by the server. For example X3D-PEX becomes EXT_X3D_PEX.
NUM_SCREENS = <i>num</i>	Specifies the total number of screens.
SCREEN_NUM = <i>num</i>	Specifies the number of current screen. from 0 (zero).
BITS_PER_RGB = <i>Number</i>	Specifies the number of significant bits in an RGB color specification. This is the log base 2 of the number of distinct shades of each primary that the hardware can generate. Note that it is not related to PLANES .
CLASS = <i>VisualClass</i>	Specifies the visual class of the root window of the default screen which is one of the following:
CLASS _visualclass= <i>visualid</i>	Specifies the visual class of the root window in a form can <i>#ifdef</i> on. The value is the numeric id of the visual. DirectColor, GrayScale, PseudoColor, StaticColor, StaticGray, TrueColor
CLASS _visualclass_depth= <i>num</i>	A symbol is defined for each visual supported for the screen. The symbol includes the class of the visual and its depth; the value is the numeric id of the visual. (If more than one visual has the same class and depth, the numeric id of the first one reported by the server is used.)S
COLOR	Defined only if CLASS is one of StaticColor, PseudoColor, TrueColor, or DirectColor .

Comment lines begin with an ! (exclamation mark) and are ignored.

Since **xrdb** can be read from standard input, use it to change the contents of properties directly from a terminal or from a shell script.

Flags

Item	Description
-all	Indicates that operation is performed on the screen-independent resource property (RESOURCE_MANAGER), as well as the screen-specific property (SCREEN_RESOURCES) on every screen of the display. For example, when used in conjunction with -query , the contents of all properties are output. For -load and -merge , the input file is processed once for each screen. The resources that occur in common in the output for every screen are collected and applied as the screen-independent resources. The remaining resources are applied for each individual per-screen property. This is the default mode of operation. This option is specific to X11R5.
-backup <i>String</i>	Specifies a suffix to append to the file name. Use it with -edit to generate a backup file. -edit is a prerequisite for -backup <i>String</i> .
-cpp <i>FileName</i>	Specifies the pathname of the C preprocessor program to use. Although the xrdb command was designed to use CPP, any program that acts as a filter and accepts the -D , -I , and -U flags can be used.
-D <i>Name=Value</i>	Passes through to the preprocessor and defines symbols to use with conditionals such as <code>#ifdef</code> .
-display <i>Display</i>	Specifies the X Server to use. It also specifies the screen to use for the -screen option, and it specifies the screen from which preprocessor symbols are derived for the -global option.
-edit <i>FileName</i>	Indicates that the contents of the specified properties should be edited into the given file, replacing any values listed there. This allows you to put changes you made to your defaults back into your resource file, preserving any comments or preprocessor lines.
-global	Indicates that the operation should only be performed on the screen-independent RESOURCE_MANAGER property. This option is specific to X11R5.
-help	Prints a brief description of the allowed flags.
-I <i>Directory</i>	(uppercase i) Passes through to the preprocessor and specifies a directory to search for files referenced with <code>#include</code> .
-load	Indicates that the input is loaded as the new value of the specified properties, replacing the old contents. This is the default action.
-merge	Indicates that the input merges with, instead of replaces, the current contents of the specified properties. This option performs a lexicographic sorted merge of the two inputs, which is probably not what you want, but remains for backward compatibility.
-n	Indicates that changes to the specified properties (when used with -load or -merge) or to the resource file (when used with -edit) should be shown on the standard output, but should not be performed. This option is specific to X11R5.
-nocpp	Indicates that the xrdb command should not run the input file through a preprocessor before loading it into properties.
-override	Indicates that the input should be added to, instead of replacing, the current contents of the specified properties. New entries override previous entries.
-query	Indicates that the current contents of the specified properties should print onto the standard output. Note that since preprocessor commands in the input resource file are part of the input file, not part of the property, they do not appear in the output from this flag.

Item	Description
-quiet	Indicates that a warning about duplicate entries should not display. This option is specific to X11R5.
-remove	Indicates that the specified properties should be removed from the server.
-retain	Indicates that the server should be instructed not to reset if the xrdb command is the first client. This should never be necessary under normal conditions, since the xdm and xinit commands always act as the first client. This option is specific to X11R5.
-screen	Indicates that the operation should only be performed on the SCREEN_RESOURCES property of the default screen of the display. This option is specific to X11R5.
-screens	Indicates that the operation should be performed on the SCREEN_RESOURCES property of each screen of the display. For -load and -merge , the input file is processed once for each screen. This option is specific to X11R5.
-symbols	Indicates that the symbols defined for the preprocessor should be printed onto the standard output.
-UName	Passes through to the preprocessor and removes any definitions of this symbol.

Examples

1. To load a file into the database:

```
xrdb -load myfile
```

2. To take the contents of the database just loaded and edit or put it into newfile:

```
xrdb -edit newfile
```

Files

The **xrdb** command generalizes the **~/.Xdefaults** files.

xsend Command

Purpose

Sends secret mail in a secure communication channel.

Syntax

xsend *User*

Description

The **xsend** command sends messages that can be read only by the intended recipient. This command is similar to the **mail** command, but the mail sent with this command is intended to be secret.

The **xsend** command is used with the **enroll** command and the **xget** command to send secret mail. The **enroll** command sets up the password used to receive secret mail. The **xget** command uses that password to receive the mail.

The **xsend** command reads standard input until an EOF (Ctrl-D) or a . (period) is entered. It then encrypts this text along with some header information and sends it. After sending the encrypted message, the **xsend** command mails a standard mail message to the recipient informing them they have received secret mail.

Note: Secret mail can only be sent to local users.

Examples

1. To send secret mail, enter:

```
xsend ron
```

When you have issued the **xsend** command with the recipient's name, the mail system is used to enter the text of the message. When you finish entering the message to user `ron`, press the Enter key, then Ctrl-D or a . (period) to exit the mail editor and send the message. The **xsend** command encrypts the message before it is sent.

2. To send a file to another user, enter:

```
xsend lance <proposal
```

In this example, the file `proposal` is sent to user `lance`.

Files

Item	Description
<code>/var/spool/secretmail/*.keys</code>	Contains the encrypted key for <i>User</i> .
<code>/var/spool/secretmail/*.[0-9]</code>	Contains the encrypted mail messages for <i>User</i> .
<code>/usr/bin/xsend</code>	Contains the command executable files.

xset Command

Purpose

Sets options for your X-Windows environment.

Syntax

```
xset [ -display Display ] [ b [ Volume [ Pitch [ Duration ] ] ] | -b | b on | b off ] [ bc | -bc ] c
[ Volume ] | -c | c on | c off ] [ - | + ] fp [ - | + | = ] Path [ ,Path, [ ... ] ] [ fp default ] [ fp rehash ]
[ - ] led [ Integer ] [ led on | led off ] [ m [ Accelerator ] [ Threshold ] ] [ m ouse ] default ]
[ p Pixel Color ] [ - ] r [ r on | r off ] [ s [ Length [ Period ] ] ] [ s blank | s noblank ] [ s expose |
s noexpose ] [ s on | s off ] [ s activate ] [ s reset ] [ s default ] [ q ]
```

Description

The **xset** command customizes your X-Windows environment.

Flags

Item	Description
-display <i>Host:Display</i>	Specifies the X server to use. For more information about servers, see the X command.

Item	Description
b or b on	Turns the bell on. This is the default setting. Note: Not all hardware is able to vary the bell characteristics, but for that which can, all of the b flag permutations and its variables are available.
b [<i>Volume</i> [<i>Pitch</i> [<i>Duration</i>]]]	Specifies the bell volume, pitch, and duration. This flag accepts up to three numeric values. Volume If only one numeric is given then it is assumed to be <i>Volume</i>. The bell volume is set to that numeric as a percentage of the bell's maximum possible volume dependent on current hardware capabilities. Pitch The second numeric in hertz values, is the tonal sound of the bell. Duration The third numeric in milliseconds, is the length of time that the bell rings.
-b or b off	Turns the bell off.
bc or -bc	Controls bug compatibility mode in the server, if possible. A preceding - (dash) disables this mode; otherwise, bug compatibility mode is enabled. The server must support the MIT-SUNDRY-NONSTANDARD protocol extension for the bc flag to work. New application development should be performed with bug compatibility mode disabled. The bc flag is provided for pre-X11 Release 4 (X11R4) clients. Some pre-X11R4 clients pass illegal values in various protocol requests. Such clients, when run with an X11R4 server, end abnormally or otherwise fail to operate correctly. This flag explicitly reintroduces certain bugs into the X server so that such clients still can be run.
c or c on	Turns on the click. System default.
c <i>Volume</i>	A numeric from 0 to 100 that specifies a percentage of the click's maximum possible volume dependent on current hardware capabilities.
-c or c off	Turns off the click.

Item	Description
fp=Path,...	Sets the font path to the directories given in the <i>Path</i> parameter. The directories are interpreted by the server, not by the client, and are server-dependent. The server ignores directories that do not contain font databases created by the mkfontdir command. All of the options and variables supported by the fp flag are available.
fp- or -fp	Deletes the font path specified by the <i>Path</i> parameter from the end of the current font path if the - (dash) precedes fp and from the front of the font path if the - (dash) follows fp .
fp+ or +fp	Adds the font path specified by the <i>Path</i> parameter to the bottom of font list if the - (dash) precedes fp and from the end of the font path if the - (dash) follows fp .
fp default	Resets the font path to the server's default.
fp rehash	Causes the server to reread the font databases in the current font path. Usually used only when adding new fonts to a font directory after running mkfontdir to recreate the font database.
led or led on	Turns all LEDs on.
-led Integer	Turns the LED specified by <i>Integer</i> off. Valid values are between 1 and 32.
led Integer	Turns the LED specified by <i>Integer</i> on. Valid values are between 1 and 32.
-led or led off	Turns all LEDs off. Note: Not all hardware assigns the same <i>Integer</i> variables to the same LED functions.
m	Allows you to control the precision of the mouse or other pointing device. If no variable or the default argument is specified, the system defaults are used. This flag accepts the following optional arguments and parameters: Acceleration Sets the multiplier for the mouse movement. The value can be specified as an integer or a fraction. Threshold Sets the minimum number of pixels needed to invoke a movement of the mouse. The value is specified in pixels. If only one parameter is given, it will be interpreted as the <i>Acceleration</i> parameter. default Uses the system defaults.

Item	Description
p	Controls pixel color values. The root background colors may be changed on some servers by altering the entries for BlackPixel and WhitePixel. Although these values are often 0 and 1, they need not be. Also, a server may choose to allocate those colors privately, in which case the xset command generates an error. The xset command also generates an error if the map entry is a read-only color. Valid parameters are: Pixel Specifies the color map entry number in decimal. Color Specifies a color.
r or r on	Enables autorepeat.
-r or r off	Disables autorepeat.
s or s default	Sets screen saver parameters to the default screen-saver characteristics.
s [<i>Length</i> [<i>Period</i>]]	Specifies the length of time the server must be inactive for the screen saver to activate. <i>Period</i> specifies the period in which the background pattern must be changed to avoid burn in. The values of <i>Length</i> and <i>Period</i> are specified in seconds. If only one numerical parameter is given, it is read as a <i>Length</i> parameter.
s on or s off	Turns the screen saver functions on and off, respectively.
s activate	Causes the screen saver to activate, even if it has been turned off.
s reset	Causes the screen saver to deactivate if it was activated.
s blank	Sets the preference to blank the video (if the hardware can do so) rather than display a background pattern.
s noblank	Sets the preference to display a pattern rather than blank the video.
s expose	Sets the preference to allow window exposures (the server can freely discard window contents).
s noexpose	Sets the preference to disable screen saver unless the server can regenerate the screens without causing exposure events.

Item	Description
q	Reports information on the current settings. These settings will be reset to default values when you log out. Note: Not all X implementations are guaranteed to honor all of these options.

Examples

1. To set the bell volume to medium, the tone to 50 hertz, and length of time the bell rings to 50 milliseconds:

```
xset b 50,50,50
```

2. To set the font path to the **/usr/lib/X11/fonts** directory:

```
xset fp= /usr/lib/x11/fonts
```

3. To cause the server to reread the font databases in the current font path:

```
xset fp rehash
```

4. To see information on the current settings:

```
xset q
```

which produces output similar to the following:

```
Keyboard Control:
  auto repeat:  on    key click percent:  0    LED mask:  00000000
  auto repeating keys:  0000000000000000
                        0000000000000000
                        0000000000000000
                        0000000000000000
  bell percent:  50    bell pitch:  400    bell duration:  100

Pointer Control:
  acceleration:  2 = 2 / 1    threshold:  4

Screen Saver:
  prefer blanking:  no    allow exposures:  no
  timeout:  0    cycle:  0

Colors:
  default colormap:  0x8006e    BlackPixel:  0    WhitePixel:  1

Font Path:
  /usr/lib/X11/fonts/,/usr/lib/X11/fonts/75dpi/,/usr/lib/X11/fonts/100dpi/,/usr/
lib/X11/fonts/oldx10/,/usr/lib/X11/fonts/oldx11/,/usr/lib/X11/fonts/bmug/,/usr/l
ib/X11/fonts/info-mac/,/usr/lib/X11/fonts/JP/,/usr/lib/X11/fonts/misc/
```

xsetroot Command

Purpose

Sets the root window parameters for the **X** command.

Syntax

```
xsetroot [ -bg Color ] [ -cursor CursorFile MaskFile ] [ -cursor_name CursorName ] [ -def ]
[ -display Display ] [ -fg Color ] [ -help ] [ -name String ] [ -rv ] [ -bitmap FileName | -gray | -grey |
-mod X Y | -solid Color ]
```

Description

The **xsetroot** command allows you to tailor the appearance of the background (root) window on a workstation display running X. Normally, you experiment with the **xsetroot** command until you find a personalized look that you like, then put the **xsetroot** command that produces it into your X startup file. If no options are specified or if the **-def** flag is specified, the window is reset to its default state. The **-def** flag can be specified with other flags and only the unspecified characteristics are reset to the default state.

Only one of the background color (tiling) changing flags (**-bitmap**, **-solid**, **-gray**, **-grey**, or **-mod**) can be specified at a time.

Flags

Item	Description
-bg <i>Color</i>	Uses the <i>Color</i> parameter as the background color.
-bitmap <i>FileName</i>	Uses the bitmap specified in the file to set the window pattern. You can make your own bitmap files (little pictures) using the bitmap program. The entire background is made of repeated tiles of the bitmap.
-cursor <i>CursorFile MaskFile</i>	Changes the pointer cursor to what you want when it is outside of any window. Cursor and mask files are bitmaps (little pictures) that can be made with the bitmap program. You probably want the mask file to be all black until you get used to the way masks work.
-cursor_name <i>CursorName</i>	Changes the pointer cursor to one of the standard cursors from the cursor font.
-def	Resets unspecified attributes to the default values. (Restores the background to the familiar gray mesh and the cursor to the hollow x shape.)
-display <i>Display</i>	Specifies the server connection. See the X command.
-fg <i>Color</i>	Uses the <i>Color</i> parameter as the foreground color. Foreground and background colors are meaningful only with the -cursor , -bitmap , or -mod flags.
-gray	Makes the entire background gray.
-grey	Makes the entire background grey.
-help	Prints a usage message and exits.
-mod <i>X Y</i>	Makes a plaid-like grid pattern on your screen. The <i>X</i> and <i>Y</i> parameters are integers ranging from 1 to 16. Zero and negative numbers are taken as 1.
-name <i>String</i>	Sets the name of the root window to the <i>String</i> parameter. There is no default value. Usually a name is assigned to a window so that the window manager can use a text representation when the window is iconified. This flag is not used because you cannot iconify the background.
-rv	Exchanges the foreground and background colors. Normally the foreground color is black and the background color is white.
-solid <i>Color</i>	Sets the background of the root window to the specified color. This flag is only used on color servers.

xss Command

Purpose

Improves the security of unattended workstations.

Syntax

```
xss [ -e CommandString ] [ -timeout Seconds ] [ -display DisplayPtr ] [ -v ] [ -fg Color ] [ -bg Color ] [ -geometry wxh+x+y ]
```

Description

The **xss** command works with the newly added Massachusetts Institute of Technology (MIT) Screen Saver Extensions in order to implement a user controllable screen saver/lock. This command is designed to improve the security of unattended workstations.

The **xss** command executes a user-specified command string when it receives a screen saver timeout message, or when the user activates the pushbutton. When no user-specified command is given, the **xss** command defaults to the **xlock** command.

Note: The **xss** command only uses the newly added MIT Screen Saver Extensions. The **xss** command does not work on an older X server, or when using an older X extension library.

Flags

Item	Description
-e <i>CommandString</i>	Sets the xss command to execute when either the screen saver times out, or the user activates the pushbutton. Note that if the <i>CommandString</i> parameter value is longer than one word, it must be surrounded by " " (double quotations).
-timeout <i>Seconds</i>	Sets the number of seconds of user inactivity before the screen saver times out, and causes the xss command to run the <i>CommandString</i> parameter.
-display <i>DisplayPtr</i>	Sets the connection to the X11 display.
-v	Turns on verbose mode.
-fg <i>Color</i>	Sets the foreground color of the pushbutton.
-bg <i>Color</i>	Sets the background color of the pushbutton.
-geometry <i>wxh+x+y</i>	Specifies the size and location of the client window.

Examples

When running remotely and using the **-display** flag for the **xss** command, remember that you may also have to use the **-display** flag option for the command that will be executed by the **xss** command. See the following running remote example:

1. Running remote:

```
xss -display myhost:0 -e "xlock -remote -display myhost:0"
```

2. Screen saver only:

```
xss -e "xlock -nolock"
```

3. Simple example:

```
xss -e xlock
```

xstr Command

Purpose

Extracts strings from C programs to implement shared strings.

Syntax

```
xstr [ -v ] [ -c ] [ - ] [ File ]
```

Description

The **xstr** command maintains a file **strings** into which strings in component parts of a large program are hashed. These strings are replaced with references to this array. This serves to implement shared constant strings, most useful if they are also read-only.

The command:

```
xstr -c File
```

extracts the strings from the C source in the *File* parameter, replacing string references by expressions of the form (**&xstr[number]**) for some number. An appropriate declaration of the xstr array is prepended to the file. The resulting C text is placed in the file **x.c**, to then be compiled. The strings from this file are appended into the **strings** file if they are not there already. Repeated strings and strings which are suffixes of existing strings do not cause changes to the file **strings**.

If a string is a suffix of another string in the file but the shorter string is seen first by the **xstr** command, both strings are placed in the file **strings**.

After all components of a large program have been compiled, a file **xs.c** declaring the common xstr array space can be created by a command of the form:

```
xstr
```

This **xs.c** file should then be compiled and loaded with the rest of the program. If possible, the array can be made read-only (shared), saving space and swap overhead.

The **xstr** command can also be used on a single file. The command:

```
xstr File
```

creates files **x.c** and **xs.c** as before, without using or affecting any **strings** file in the same directory.

It may be useful to run the **xstr** command after the C preprocessor if any macro definitions yield strings or if there is conditional code which contains strings which may not, in fact, be needed.

The **xstr** command reads from its standard input when the **-** (minus sign) flag is given and does not alter the **strings** file unless the **-c** flag is specified also.

An appropriate command sequence for running the **xstr** command after the C preprocessor is:

```
cc -E name.c | xstr -c -  
cc -C x.c  
mv x.o name.o
```

The **xstr** command does not touch the file **strings** unless new items are added, thus the **make** command can avoid remaking the **xs.o** file unless truly necessary.

Flags

Item	Description

-c	Extracts strings from the specified file, and places them in the strings file.
-v	Verbose mode. Tells when strings are found, or new in the strings file.
-	Reads from standard input.

Examples

1. To extract the strings from the C source in the *File.c* parameter, replacing string references by expressions of the form (**&xstr[number]**):

```
xstr -c File.c
```

An appropriate declaration of the xstr array is prepended to the file. The resulting C text is placed in the file **x.c**, to then be compiled.

2. To declare the common xstr array space in the **xs.c** file:

```
xstr
```

Files

Item	Description
strings	File which contains the extracted strings.
x.c	Massaged C source.
xs.c	C source for definition of array xstr.
/tmp/xs*	Temporary file when xstr command does not touch the strings file.
/usr/ccs/bin/mkstr	Contains an executable file.
/usr/ccs/bin/mkstr	Contains an executable file for Berkeley environment.

xterm Command

Purpose

Provides a terminal emulator for the X Window System.

Note: The **xterm** command is ported from the Massachusetts Institute of Technology (MIT) X Window System, Version 11, Release 6 with no functional enhancements. The **xterm** command does not have support for globalization. For the localized and internationalized terminal emulator, the user can use the **aixterm** or **dtterm** commands.

Syntax

```
xterm [ --Xtoolkitoption... ] [ -Option ... ]
```

Description

The **xterm** program is a terminal emulator for the X Window System. It provides DEC VT102 and Tektronix 4014 compatible terminals for programs that cannot use the window system directly. If the underlying operating system supports terminal resizing capabilities, the **xterm** program uses the facilities to notify programs running in the window whenever it is resized.

The VT102 and Tektronix 4014 terminals each have their own window so that you can edit text in one and look at graphics in the other at the same time. To maintain the correct aspect ratio (height/width), Tektronix graphics are restricted to the largest box with a 4014 aspect ratio that will fit in the window. This box is located in the upper left area of the window.

Although both windows might be displayed at the same time, one of them is considered the *active window* for receiving keyboard input and terminal output. This is the window that contains the text cursor. The active window can be chosen through escape sequences, the VT Options menu in the VT102 window, and the Tek Options menu in the 4014 window.

Emulations

The VT102 emulation is fairly complete, but does not support smooth scrolling, VT52 mode, the flashing character attribute, or the double-wide and double-size character sets. The **termcap** file entries that work with the **xterm** command include **xterm**, **vt102**, **vt100** and **``ansi**," and the **xterm** command automatically searches the **termcap** file in this order for these entries and then sets the **TERM** and the **TERMCAP** environment variables.

Many of the special **xterm** features might be modified under program control through a set of escape sequences different from the standard VT102 escape sequences.

The Tektronix 4014 emulation is also fairly good. It supports 12-bit graphics addressing, scaled to the window size. Four different font sizes and five different lines types are supported. There is no write-thru or defocused mode support.

The Tektronix text and graphics commands are recorded internally by the **xterm** command and may be written to a file by sending the COPY escape sequence (or through the Tektronix menu, as described in the following sections). The name of the file will be **COPYyy-MM-dd.hh:mm:ss**, where **yy**, **MM**, **dd**, **hh**, **mm**, and **ss** are the year, month, day, hour, minute, and second when the copy is performed (the file is created in the directory that the **xterm** command is started in, or the home directory for a login **xterm**).

Other Features

The **xterm** command automatically highlights the text cursor when the pointer enters the window (selected) and unhighlights it when the pointer leaves the window (unselected). If the window is the focus window, the text cursor is highlighted no matter where the pointer is located.

In VT102 mode, there are escape sequences to activate and deactivate an alternate screen buffer, which is the same size as the display area of the window. When activated, the current screen is saved and replaced with the alternate screen. Saving of lines scrolled off the top of the window is disabled until the usual screen is restored.

The **termcap** file entry for the **xterm** command allows the **vi** command editor to switch to the alternate screen for editing and to restore the screen on exit.

In either VT102 or Tektronix mode, there are escape sequences to change the name of the windows.

Options

The **xterm** terminal emulator accepts all of the standard X Toolkit command-line options as well as the following (if the option begins with a + instead of a -, the option is restored to its default value):

Item	Description
-help	Causes the xterm command to print out a message describing its options.
-132	Usually, the VT102 DECCOLM escape sequence that switches between 80- and 132-column mode is ignored. This option causes the DECCOLM escape sequence to be recognized, and the xterm window will resize appropriately.

Item	Description
-ah	Indicates that the xterm command should always highlight the text cursor. By default, the xterm command will display a hollow text cursor whenever the focus is lost or the pointer leaves the window.
+ah	Indicates that the xterm command should do text cursor highlighting based on focus.
-b <i>Number</i>	Specifies the size of the inner border (the distance between the outer edge of the characters and the window border) in pixels. The default is 2.
-cc <i>CharacterClassRange:Value[,...]</i>	Sets classes indicated by the given ranges for use in selecting by words.
-cn	Indicates that newlines should not be cut in line-mode selections.
+cn	Indicates that newlines should be cut in line-mode selections.
-cr <i>Color</i>	Specifies the color to use for the text cursor. The default is to use the same foreground color that is used for text.
-cu	Indicates that the xterm command should work around a bug in the more program that causes it to incorrectly display lines that are exactly the width of the window and are followed by a line beginning with a tab (the leading tabs are not displayed). This option is so named because it was originally thought to be a bug in the curses function cursor motion package.
+cu	Indicates that xterm should not work around the more function bug previously mentioned.
-e <i>Program [Arguments]</i>	Specifies the program (and its command-line arguments) to be run in the xterm window. It also sets the window title and icon name to be the base name of the program being run if neither the -T nor the -n option is given on the command line. Note: This must be the last option on the command line.
-fb <i>Font</i>	Specifies a font to be used when displaying bold text. This font must be the same height and width as the normal font. If only one of the normal or bold fonts is specified, it will be used as the normal font and the bold font will be produced by overstriking this font. The default is to do overstriking of the normal font.
-i	Turns on the useInsertMode resource.
+i	Turns off the useInsertMode resource.

Item	Description
-j	Indicates that the xterm command should do jump scrolling. Usually, text is scrolled one line at a time; this option allows the xterm command to move multiple lines at a time so that it does not fall as far behind. Its use is strongly recommended because it makes the xterm command much faster when scanning through large amounts of text. The VT100 escape sequences for enabling and disabling smooth scrolling as well as the VT Options menu can be used to turn this feature on or off.
+j	Indicates that the xterm command should not do jump scrolling.
-ls	Indicates that the shell that is started in the xterm window is a login shell (in other words, the first character of the <i>ArgumentVector</i> parameter is a dash, indicating to the shell that it should read the user's .login or .profile file).
+ls	Indicates that the shell that is started should not be a login shell (in other words, it will be a usual subshell).
-mb	Indicates that the xterm command should ring a margin bell when the user types near the right end of a line. This option can be turned on and off from the VT Options menu.
+mb	Indicates that the margin bell should not be rung.
-mc <i>Milliseconds</i>	Specifies the maximum time between multiclick selections.
-ms <i>Color</i>	Specifies the color to be used for the pointer cursor. The default is to use the foreground color.
-nb <i>Number</i>	Specifies the number of characters from the right end of a line at which the margin bell, if enabled, will ring. The default is 10.
-rw	Indicates that reverse wraparound should be allowed. This allows the cursor to back up from the leftmost column of one line to the rightmost column of the previous line. This is very useful for editing long shell command lines and is encouraged. This option can be turned on and off from the VT Options menu.
+rw	Indicates that reverse wraparound should not be allowed.
Item	Description
-aw	Indicates that auto wraparound should be allowed. This allows the cursor to automatically wrap to the beginning of the next line when it is at the rightmost position of a line and text is output.
+aw	Indicates that auto wraparound should not be allowed.

Item	Description
-s	Indicates that the xterm command may scroll asynchronously, meaning that the screen does not have to be kept completely up to date while scrolling. This allows the xterm command to run faster when network latencies are high and is typically useful when running across a large Internet or many gateways.
+s	Indicates that the xterm command should scroll synchronously.
-sb	Indicates that some number of lines that are scrolled off the top of the window should be saved and that a scrollbar should be displayed so that those lines can be viewed. This option can be turned on and off from the VT Options menu.
+sb	Indicates that a scrollbar should not be displayed.
-sf	Indicates that Sun Function Key escape codes should be generated for function keys.
+sf	Indicates that the standard escape codes should be generated for function keys.
-si	Indicates that output to a window should not automatically reposition the screen to the bottom of the scrolling region. This option can be turned on and off from the VT Options menu.
+si	Indicates that output to a window should cause it to scroll to the bottom.
-sk	Indicates that pressing a key while using the scrollbar to review previous lines of text should cause the window to be repositioned automatically in the usual position at the bottom of the scroll region.
+sk	Indicates that pressing a key while using the scrollbar should not cause the window to be repositioned.
-sl <i>Number</i>	Specifies the number of lines to save that have been scrolled off the top of the screen. The default is 64.
-t	Indicates that the xterm command should start in Tektronix mode, rather than in VT102 mode. Switching between the two windows is done using the Options menus.
+t	Indicates that the xterm command should start in VT102 mode.
-tm <i>String</i>	Specifies a series of terminal-setting keywords followed by the characters that should be bound to those functions, similar to the stty program. Allowable keywords include: intr , quit , erase , kill , eof , eol , swtch , start , stop , brk , susp , dsusp , rpnt , flush , weras , and lnext . Control characters might be specified as <i>^Character</i> (for example, <i>^c</i> or <i>^u</i>), and <i>^?</i> may be used to indicate Delete.

Item	Description
-tn <i>Name</i>	Specifies the name of the terminal type to be set in the TERM environment variable. This terminal type must exist in the termcap database and should have li# and co# entries.
-ut	Indicates that the xterm command should not write a record into the /etc/utmp system log file.
+ut	Indicates that the xterm command should write a record into the /etc/utmp system log file.
-vb	Indicates that a visual bell is preferred over an audible one. Instead of ringing the terminal bell whenever the Ctrl+G key sequence signal is received, the window will flash.
+vb	Indicates that a visual bell should not be used.
-wf	Indicates that the xterm command should wait for the window to be mapped the first time before starting the subprocess so that the initial terminal size settings and environment variables are correct. It is the application's responsibility to catch subsequent terminal size changes.
+wf	Indicates that the xterm command should not wait before starting the subprocess.
-C	Indicates that this window should receive console output. This is not supported on all systems. To obtain console output, you must be the owner of the console device, and you must have read and write permission for it. If you are running X windows under xdm on the console screen, you may need to have the session startup and reset programs explicitly change the ownership of the console device in order to get this option to work.
-Sccn	Specifies the last two letters of the name of a pseudoterminal to use in worker mode, plus the number of the inherited file descriptor. The option is parsed <code>` ` %c%c%d"</code> . This allows the xterm command to be used as an input and output channel for an existing program and is sometimes used in specialized applications.

The following command-line arguments are provided for compatibility with older versions. They may not be supported in the next release as the X Toolkit provides standard options that accomplish the same task.

Item	Description
%geom	Specifies the preferred size and position of the Tektronix window. It is shorthand for specifying the *tekGeometry resource.
#geom	Specifies the preferred position of the icon window. It is shorthand for specifying the *iconGeometry resource.
-T String	Specifies the title for the xterm program's windows. It is equivalent to -title .

Item	Description
-n <i>String</i>	Specifies the icon name for the xterm program's windows. It is shorthand for specifying the *iconName resource. Note that this is not the same as the Toolkit option -name (see the following). The default icon name is the application name.
-r	Indicates that reverse video should be simulated by swapping the foreground and background colors. It is equivalent to -rv .
-w <i>Number</i>	Specifies the width in pixels of the border surrounding the window. It is equivalent to -borderwidth or -bw .

The following standard X Toolkit command-line arguments are commonly used with the **xterm** command:

Item	Description
-bg <i>Color</i>	Specifies the color to use for the background of the window. The default is white.
-bd <i>Color</i>	Specifies the color to use for the border of the window. The default is black.
-bw <i>Number</i>	Specifies the width in pixels of the border surrounding the window.
-fg <i>Color</i>	Specifies the color to use for displaying text. The default is black.
-fn <i>Font</i>	Specifies the font to be used for displaying usual text. The default is fixed.
-name <i>Name</i>	Specifies the application name under which resources are to be obtained, rather than the default executable file name. The <i>Name</i> parameter should not contain . (dot) or * (asterisk) characters.
-title <i>String</i>	Specifies the window title string, which may be displayed by window managers if the user so chooses. The default title is the command line specified after the -e option, if any; otherwise, the application name.
-rv	Indicates that reverse video should be simulated by swapping the foreground and background colors.
-geometry <i>Geometry</i>	Specifies the preferred size and position of the VT102 window; see the X command.
-display <i>Display</i>	Specifies the X server to contact; see the X command.
-xrm <i>ResourceString</i>	Specifies a resource string to be used. This is especially useful for setting resources that do not have separate command-line options.
-iconic	Indicates that the xterm command should ask the window manager to start it as an icon rather than as the usual window.

Resources

The program understands all of the core X Toolkit resource names and classes as well as:

Item	Description
iconGeometry (class IconGeometry)	Specifies the preferred size and position of the application when iconified. It is not necessarily obeyed by all window managers.
termName (class TermName)	Specifies the terminal type name to be set in the TERM environment variable.

Item	Description
title (class Title)	Specifies a string that may be used by the window manager when displaying this application.
ttyModes (class TtyModes)	Specifies a string containing terminal-setting keywords and the characters to which they may be bound. Allowable keywords include: intr , quit , erase , kill , eof , eol , swtch , start , stop , brk , susp , dsusp , rprnt , flush , weras , and lnext . Control characters may be specified as ^Character (for example, ^c or ^u) and ^? may be used to indicate Delete. This is very useful for overriding the default terminal settings without having run an stty program every time an xterm window is started.
useInsertMode (class useInsertMode)	Forces the use of insert mode by adding appropriate entries to the TERMCAP environment variable. This is useful if the system termcap is broken. The default is false .
utmpInhibit (class UtmpInhibit)	Specifies whether xterm should try to record the user's terminal in /etc/utmp .
sunFunctionKeys (class SunFunctionKeys)	Specifies whether Sun Function Key escape codes should be generated for function keys instead of standard escape sequences.
waitForMap (class WaitForMap)	Specifies whether the xterm command should wait for the initial window map before starting the subprocess. The default is False .

The following resources are specified as part of the **vt100** widget (class **VT100**):

Item	Description
allowSendEvents (class AllowSendEvents)	Specifies whether synthetic key and button events (generated using the X protocol SendEvent request) should be interpreted or discarded. The default is False , meaning they are discarded. Note that allowing such events creates a large security hole.
alwaysHighlight (class AlwaysHighlight)	Specifies whether xterm should always display a highlighted text cursor. By default, a hollow text cursor is displayed whenever the pointer moves out of the window or the window loses the input focus.
appcursorDefault (class AppcursorDefault)	If True , the cursor keys are initially in application mode. The default is False .

Item	Description
appkeypadDefault (class AppkeypadDefault)	If True, the keypad keys are initially in application mode. The default is False.
autoWrap (class AutoWrap)	Specifies whether auto wraparound should be enabled. The default is True.
bellSuppresstime (class BellSuppresstime)	Specifies the number of milliseconds after a bell command is sent during which additional bells will be suppressed. The default is 200. If set to nonzero, additional bells will also be suppressed until the server reports that processing of the first bell has been completed; this feature is most useful with the visible bell.
boldFont (class BoldFont)	Specifies the name of the bold font to use instead of overstriking.
c132 (class C132)	Specifies whether the VT102 DECCOLM escape sequence should be honored. The default is False.
charClass (class CharClass)	Specifies comma-separated lists of character class bindings of the form <i>[low-]high:value</i> . These are used in determining which sets of characters should be treated the same when doing cut and paste. See “Character Classes” on page 1415 .
curses (class Curses)	Specifies whether the last column bug in the curses function should be worked around. The default is False.
cutNewline (class cutNewline)	If false , triple clicking to select a line does not include the Newline at the end of the line. If true , the Newline is selected. The default is true .
cutToBeginningofLines (class CutToBeginningOfLine)	If false , triple clicking to select a line selects only from the current word forward. If true , the entire line is selected. The default is true .
background (class Background)	Specifies the color to use for the background of the window. The default is white.
foreground (class Foreground)	Specifies the color to use for displaying text in the window. Setting the class name instead of the instance name is an easy way to have everything that would usually be displayed in the text color to change color. The default is black.
cursorColor (class Foreground)	Specifies the color to use for the text cursor. The default is black.
eightBitInput (class EightBitInput)	If True, meta characters input from the keyboard are presented as a single character with the eighth bit turned on. If False, meta characters are converted into a 2-character sequence with the character itself preceded by ESC . The default is True.
eightBitOutput (class EightBitOutput)	Specifies whether 8-bit characters sent from the host should be accepted as is or stripped when printed. The default is True.
font (class Font)	Specifies the name of the normal font. The default is fixed.

Item**font1** (class **Font1**)**font2** (class **Font2**)**font3** (class **Font3**)**font4** (class **Font4**)**font5** (class **Font5**)**font6** (class **Font6**)**geometry** (class **Geometry**)**hpLowerleftBugCompat** (class
hpLowerleftBugCompat)**internalBorder** (class **BorderWidth**)**jumpScroll** (class **JumpScroll**)**Item****loginShell** (class **LoginShell**)**marginBell** (class **MarginBell**)**multiClickTime** (class **MultiClickTime**)**multiScroll** (class **MultiScroll**)**nMarginBell** (class **Column**)**pointerColor** (class **Foreground**)**pointerColorBackground** (class **Background**)**pointerShape** (class **Cursor**)**Description**

Specifies the name of the first alternative font.

Specifies the name of the second alternative font.

Specifies the name of the third alternative font.

Specifies the name of the fourth alternative font.

Specifies the name of the fifth alternative font.

Specifies the name of the sixth alternative font.

Specifies the preferred size and position of the VT102 window.

Specifies whether to work around a bug in **xdb**, which ignores termcap and always sends ESC F to move to the lower left corner. **true** causes **xterm** in interpret ESC F as a request to move to the lower left corner of the screen. The default is **false**.

Specifies the number of pixels between the characters and the window border. The default is 2.

Specifies whether jump scrolling should be used. The default is True.

Description

Specifies whether the shell to be run in the window should be started as a login shell. The default is False.

Specifies whether the bell should be run when the user types near the right margin. The default is False.

Specifies the maximum time in milliseconds between multiclick select events. The default is 250 milliseconds.

Specifies whether scrolling should be done asynchronously. The default is False.

Specifies the number of characters from the right margin at which the margin bell should be rung, when enabled.

Specifies the foreground color of the pointer. The default is **XtDefaultForeground**.Specifies the background color of the pointer. The default is **XtDefaultBackground**.Specifies the name of the shape of the pointer. The default is **xterm**.

Item	Description
resizeGravity (class ResizeGravity)	Affects the behavior when the window is resized to be taller or shorter. NorthWest specifies that the top line of text on the screen stays fixed. If the window is made shorter, lines are dropped from the bottom; if the window is made taller, blank lines are added at the bottom. This is compatible with the behavior in MIT version X11R4. SouthWest (the default) specifies that the bottom line of text on the screen stays fixed. If the window is made taller, additional saved lines will be scrolled down onto the screen; if the window is made shorter, lines will be scrolled off the top of the screen, and the top saved lines will be dropped.
reverseVideo (class ReverseVideo)	Specifies whether reverse video should be simulated. The default is False.
reverseWrap (class ReverseWrap)	Specifies whether reverse wraparound should be enabled. The default is False.
saveLines (class SaveLines)	Specifies the number of lines to save beyond the top of the screen when a scrollbar is turned on. The default is 64.
scrollBar (class ScrollBar)	Specifies whether the scrollbar should be displayed. The default is False.
scrollTtyOutput (class ScrollCond)	Specifies whether output to the terminal should automatically cause the scrollbar to go to the bottom of the scrolling region. The default is True.
scrollKey (class ScrollCond)	Specifies whether pressing a key should automatically cause the scrollbar to go to the bottom of the scrolling region. The default is False.
scrollLines (class ScrollLines)	Specifies the number of lines that the scroll-back and scroll-forw actions should use as a default. The default value is 1.
signalInhibit (class SignalInhibit)	Specifies whether the entries in the Main Options menu for sending signals to xterm should be disallowed. The default is False.
tekGeometry (class Geometry)	Specifies the preferred size and position of the Tektronix window.
tekInhibit (class TekInhibit)	Specifies whether the escape sequence to enter Tektronix mode should be ignored. The default is False.
tekSmall (class TekSmall)	Specifies whether the Tektronix mode window should start in its smallest size if no explicit geometry is given. This is useful when running the xterm command on displays with small screens. The default is False.
tekStartup (class TekStartup)	Specifies whether xterm should start up in Tektronix mode. The default is False.

Item	Description
titeInhibit (class TiteInhibit)	Specifies whether xterm should remove ti and te termcap file entries (used to switch between alternate screens during startup of many screen-oriented programs) from the TERMCAP string. If set, the xterm command also ignores the escape sequence to switch to the alternate screen.
translations (class Translations)	Specifies the key and button bindings for menus, selections, programmed strings, and so forth. For more information, see Actions .
visualBell (class VisualBell)	Specifies whether a visible bell (flashing) should be used instead of an audible bell when the Ctrl+G key sequence signal is received. The default is False.

The following resources are specified as part of the **tek4014** widget (class **Tek4014**):

Item	Description
width (class Width)	Specifies the width of the Tektronix window in pixels.
height (class Height)	Specifies the height of the Tektronix window in pixels.
fontLarge (class Font)	Specifies the large font to use in the Tektronix window.
font2 (class Font)	Specifies font number 2 to use in the Tektronix window.
font3 (class Font)	Specifies font number 3 to use in the Tektronix window.
fontSmall (class Font)	Specifies the small font to use in the Tektronix window.
initialFont (class InitialFont)	Specifies which of the four Tektronix fonts to use initially. Values are the same as for the set-tek-text action. The default is large.
ginTerminator (class GinTerminator)	Specifies what characters should follow a GIN report or status report. The possibilities are <code>`none'</code> , which sends no terminating characters; <code>CRonly</code> , which sends CR; and <code>CR&EOT</code> , which sends both CR and EOT. The default is none.

The resources that may be specified for the various menus are described in the documentation for the **Athena SimpleMenu** widget. Following is a list of the names and classes of the entries in each of the menus.

The mainMenu has the following entries:

Item	Description
securekbd (class SmeBSB)	Invokes the secure() action.
allowsends (class SmeBSB)	Invokes the allow-send-events(toggle) action.
redraw (class SmeBSB)	Invokes the redraw() action.

Item	Description
line1 (class SmeLine)	This is a separator.
suspend (class SmeBSB)	Invokes the send-signal(tstp) action on systems that support job control.
continue (class SmeBSB)	Invokes the send-signal(cont) action on systems that support job control.
interrupt (class SmeBSB)	Invokes the send-signal(int) action.
hangup (class SmeBSB)	Invokes the send-signal(hup) action.
terminate (class SmeBSB)	Invokes the send-signal(term) action.
kill (class SmeBSB)	Invokes the send-signal(kill) action.
line2 (class SmeLine)	This is a separator.
quit (class SmeBSB)	Invokes the quit() action.

The vtMenu has the following entries:

Item	Description
scrollbar (class SmeBSB)	Invokes the set-scrollbar(toggle) action.
jumpscroll (class SmeBSB)	Invokes the set-jumpscroll(toggle) action.
reversevideo (class SmeBSB)	Invokes the set-reverse-video(toggle) action.
autowrap (class SmeBSB)	Invokes the set-autowrap(toggle) action.
reversewrap (class SmeBSB)	Invokes the set-reversewrap(toggle) action.
autolinefeed (class SmeBSB)	Invokes the set-autolinefeed(toggle) action.
appcursor (class SmeBSB)	Invokes the set-appcursor(toggle) action.
app keypad (class SmeBSB)	Invokes the set-app keypad(toggle) action.
scrollkey (class SmeBSB)	Invokes the set-scroll-on-key(toggle) action.
scrollttyoutput (class SmeBSB)	Invokes the set-scroll-on-tty-output(toggle) action.
allow132 (class SmeBSB)	Invokes the set-allow132(toggle) action.
cursesemul (class SmeBSB)	Invokes the set-cursesemul(toggle) action.
visualbell (class SmeBSB)	Invokes the set-visualbell(toggle) action.
marginbell (class SmeBSB)	Invokes the set-marginbell(toggle) action.
altscreen (class SmeBSB)	This entry is currently disabled.
line1 (class SmeLine)	This is a separator.
softreset (class SmeBSB)	Invokes the soft-reset() action.
hardreset (class SmeBSB)	Invokes the hard-reset() action.
clearsavedlines (class SmeBSB)	Invokes the clear-saved-lines() action.
line2 (class SmeLine)	This is a separator.
tekshow (class SmeBSB)	Invokes the set-visibility(tek,toggle) action.
tekmode (class SmeBSB)	Invokes the set-terminal-type(tek) action.
vthide (class SmeBSB)	Invokes the set-visibility(vt,off) action.

The fontMenu has the following entries:

Item	Description
fontdefault (class SmeBSB)	Invokes the set-vt-font(d) action.
font1 (class SmeBSB)	Invokes the set-vt-font(1) action.
font2 (class SmeBSB)	Invokes the set-vt-font(2) action.
font3 (class SmeBSB)	Invokes the set-vt-font(3) action.
font4 (class SmeBSB)	Invokes the set-vt-font(4) action.
font5 (class SmeBSB)	Invokes the set-vt-font(5) action.
font6 (class SmeBSB)	Invokes the set-vt-font(6) action.
fontescape (class SmeBSB)	Invokes the set-vt-font(e) action.
fontsel (class SmeBSB)	Invokes the set-vt-font(s) action.

The tekMenu has the following entries:

Item	Description
tektextlarge (class SmeBSB)	Invokes the set-tek-text(l) action.
tektext2 (class SmeBSB)	Invokes the set-tek-text(2) action.
tektext3 (class SmeBSB)	Invokes the set-tek-text(3) action.
tektextsmall (class SmeBSB)	Invokes the set-tek-text(s) action.
line1 (class SmeLine)	This is a separator.
tekpage (class SmeBSB)	Invokes the tek-page() action.
tekreset (class SmeBSB)	Invokes the tek-reset() action.
tekcopy (class SmeBSB)	Invokes the tek-copy() action.
line2 (class SmeLine)	This is a separator.
vtshow (class SmeBSB)	Invokes the set-visibility(vt,toggle) action.
vtmode (class SmeBSB)	Invokes the set-terminal-type(vt) action.
tekhide (class SmeBSB)	Invokes the set-visibility(tek,toggle) action.

The following resources are useful when specified for the **Athena Scrollbar** widget:

Item	Description
thickness (class Thickness)	Specifies the width in pixels of the scrollbar.
background (class Background)	Specifies the color to use for the background of the scrollbar.
foreground (class Foreground)	Specifies the color to use for the foreground of the scrollbar. The <i>thumb</i> of the scrollbar is a simple checkerboard pattern with alternating pixels for foreground and background colors.

Pointer Usage

After the VT102 window is created, the **xterm** command allows you to select text and copy it within the same or other windows.

The selection functions are invoked when the pointer buttons are used with no modifiers, and when they are used with the Shift key. The assignment of the functions to keys and buttons may be changed through the resource database.

Pointer button 1 (usually left) is used to save text into the cut buffer. Move the cursor to beginning of the text, and then hold the button down while moving the cursor to the end of the region and releasing the button. The selected text is highlighted and is saved in the global cut buffer and made the PRIMARY selection when the button is released.

Double-clicking selects by words, triple-clicking selects by lines, and quadruple-clicking goes back to characters. Multiple-click is determined by the amount of time from button up to button down, so you can change the selection unit in the middle of a selection. If the key or button bindings specify that an X selection is to be made, the **xterm** command will leave the selected text highlighted for as long as it is the selection owner.

Pointer button 2 (usually middle) "types" (pastes) the text from the PRIMARY selection, if any, otherwise from the cut buffer, inserting it as keyboard input.

Pointer button 3 (usually right) extends the current selection. If pressed while closer to the right edge of the selection than the left, it extends or contracts the right edge of the selection. If you contract the selection past the left edge of the selection, the **xterm** command assumes you really meant the left edge, restores the original selection, and then extends or contracts the left edge of the selection.

And the opposite also applies: if pressed while closer to the left edge of the selection than the right, it extends/contracts the left edge of the selection. If you contract the selection past the right edge of the selection, the **xterm** command assumes you really meant the right edge, restores the original selection, and then extends/contracts the right edge of the selection. Extension starts in the selection unit mode that the last selection or extension was performed in; you can multiple-click to cycle through them.

By cutting and pasting pieces of text without trailing new lines, you can take text from several places in different windows and form a command to the shell, for example, or take output from a program and insert it into your favorite editor. Because the cut buffer is globally shared among different applications, regard it as a "file" whose contents you know. The terminal emulator and other text programs should be treating it as if it were a text file; in other words, the text is delimited by new lines.

The scroll region displays the position and amount of text currently showing in the window (highlighted) relative to the amount of text actually saved. As more text is saved (up to the maximum), the size of the highlighted area decreases.

Clicking button 1 with the pointer in the scroll region moves the adjacent line to the top of the display window.

Clicking button 3 moves the top line of the display window down to the pointer position.

Clicking button 2 moves the display to a position in the saved text that corresponds to the pointer's position in the scrollbar.

Unlike the VT102 window, the Tektronix window does not allow the copying of text. It does allow Tektronix GIN mode, and in this mode the cursor will change from an arrow to a cross. Pressing any key will send that key and the current coordinates of the cross cursor. Pressing button one, two, or three will return the letters l, m, and r, respectively.

If the Shift key is pressed when a pointer button is pressed, the corresponding uppercase letter is sent. To distinguish a pointer button from a key, the high bit of the character is set (but this bit is usually stripped unless the terminal mode is RAW; see the **tty** command for details).

Menus

The **xterm** command has four menus, named `mainMenu`, `vtMenu`, `fontMenu`, and `tekMenu`. Each menu opens under the correct combinations of key and button presses. Most menus are divided into two sections, separated by a horizontal line. The top portion contains various modes that can be altered. A check mark is displayed next to a mode that is currently active. Selecting one of these modes toggles its state. The bottom portion of the menu lists command entries; selecting one of these performs the indicated function.

The xterm menu opens when the control key and pointer button one are pressed in a window. The mainMenu contains items that apply to both the VT102 and Tektronix windows. The **Secure Keyboard** mode is used when typing in passwords or other sensitive data in an unsecure environment.

Notable entries in the command section of the menu are **Continue**, **Suspend**, **Interrupt**, **Hangup**, **Terminate**, and **Kill**, which send the **SIGCONT**, **SIGTSTP**, **SIGINT**, **SIGHUP**, **SIGTERM**, and **SIGKILL** signals, respectively, to the process group of the process running under **xterm** (usually the shell). The **Continue** function is especially useful if the user has accidentally pressed Ctrl+Z, suspending the process.

The vtMenu sets various modes in the VT102 emulation, and is opened when the control key and pointer button two are pressed in the VT102 window. In the command section of this menu, the soft reset entry will reset scroll regions. This can be convenient when some program has left the scroll regions set incorrectly (often a problem when using VMS or TOPS-20).

The full reset entry will clear the screen, reset tabs to every eight columns, and reset the terminal modes (such as wrap and smooth scroll) to their initial states just after the **xterm** command has finished processing the command-line options.

The fontMenu sets the font used in the VT102 window. In addition to the default font and a number of alternatives that are set with resources, the menu offers the font last specified by the Set Font escape sequence (See " [Control Sequences](#)") and the current selection as a font name (if the PRIMARY selection is owned).

The tekMenu sets various modes in the Tektronix emulation, and is opened when the control key and pointer button two are pressed in the Tektronix window. The current font size is checked in the Modes section of the menu. The **PAGE** entry in the command section clears the Tektronix window.

Security

X windows environments differ in their security consciousness. MIT servers, run under **xdm**, are capable of using a *magic cookie* authorization scheme that can provide a reasonable level of security for many people. If your server is only using a host-based mechanism to control access to the server (see the **xhost** command), and if you enable access for a host and other users are also permitted to run clients on that same host, there is every possibility that someone can run an application that will use the basic services of the X protocol to snoop on your activities, potentially capturing a transcript of everything you type at the keyboard.

This is of particular concern when you want to type in a password or other sensitive data. The best solution to this problem is to use a better authorization mechanism than host-based control, but a simple mechanism exists for protecting keyboard input in the **xterm** command.

The xterm menu contains a **Secure Keyboard** entry that, when enabled, ensures that all keyboard input is directed *only* to the **xterm** command (using the **GrabKeyboard** protocol request). When an application prompts you for a password (or other sensitive data), you can enable **Secure Keyboard** using the menu, type in the data, and then disable **Secure Keyboard** using the menu again.

Only one X client at a time can secure the keyboard, so when you attempt to enable **Secure Keyboard** it may fail. In this case, the bell will sound. If the **Secure Keyboard** succeeds, the foreground and background colors will be exchanged (as if you selected the **Reverse Video** entry in the Modes menu); they will be exchanged again when you exit secure mode. If the colors do *not* switch, be *very* suspicious that you are being spoofed.

If the application you are running displays a prompt before asking for the password, it is safest to enter secure mode *before* the prompt gets displayed, and to make sure that the prompt gets displayed correctly (in the new colors), to minimize the probability of spoofing. You can also bring up the menu again and make sure that a check mark is displayed next to the entry.

Secure Keyboard mode will be disabled automatically if your xterm window becomes iconified (or otherwise unmapped), or if you start up a reparenting window manager (that places a title bar or other decoration around the window) while in **Secure Keyboard** mode. (This is a feature of the X protocol not easily overcome.) When this happens, the foreground and background colors will be switched back and the bell will sound in warning.

Character Classes

Clicking the middle mouse button twice in rapid succession will cause all characters of the same class (such as letters, white space, punctuation) to be selected. Because different people have different preferences for what should be selected (for example, should file names be selected as a whole or only the separate subnames), the default mapping can be overridden through the use of the **charClass** (class **CharClass**) resource.

This resource is a series of comma-separated *range:value* pairs. The *range* is either a single number or *low-high* in the range of 0 to 127, corresponding to the ASCII code for the character or characters to be set. The *value* is arbitrary, although the default table uses the character number of the first character occurring in the set.

The default table is:

```
static int charClass[128] = {
/* NUL  SOH  STX  ETX  EOT  ENQ  ACK  BEL */
 32,  1,  1,  1,  1,  1,  1,  1,
/* BS   HT   NL   VT   NP   CR   SO   SI */
 1,  32,  1,  1,  1,  1,  1,  1,
/* DLE  DC1  DC2  DC3  DC4  NAK  SYN  ETB */
 1,  1,  1,  1,  1,  1,  1,  1,
/* CAN  EM   SUB  ESC  FS   GS   RS   US */
 1,  1,  1,  1,  1,  1,  1,  1,
/* SP   !    "    #    $    %    &    ' */
 32,  33,  34,  35,  36,  37,  38,  39,
/* (    )    *    +    -    .    /    */
 40,  41,  42,  43,  44,  45,  46,  47,
/* 0    1    2    3    4    5    6    7 */
 48,  48,  48,  48,  48,  48,  48,  48,
/* 8    9    :    ;    <    =    >    ? */
 48,  48,  58,  59,  60,  61,  62,  63,
/* @    A    B    C    D    E    F    G */
 64,  48,  48,  48,  48,  48,  48,  48,
/* H    I    J    K    L    M    N    O */
 48,  48,  48,  48,  48,  48,  48,  48,
/* P    Q    R    S    T    U    V    W */
 48,  48,  48,  48,  48,  48,  48,  48,
/* X    Y    Z    [    \    ]    ^    _ */
 48,  48,  48,  91,  92,  93,  94,  48,
/* `    a    b    c    d    e    f    g */
 96,  48,  48,  48,  48,  48,  48,  48,
/* h    i    j    k    l    m    n    o */
 48,  48,  48,  48,  48,  48,  48,  48,
/* p    q    r    s    t    u    v    w */
 48,  48,  48,  48,  48,  48,  48,  48,
/* x    y    z    {    |    }    ~    DEL */
 48,  48,  48, 123, 124, 125, 126, 1};
```

For example, the string 33:48,37:48,45-47:48,64:48 indicates that the exclamation mark, percent sign, dash, period, slash, and & characters should be treated the same way as characters and numbers. This is useful for cutting and pasting electronic mailing addresses and file names.

Actions

It is possible to rebind keys (or sequences of keys) to arbitrary strings for input by changing the translations for the **vt100** or **tek4014** widgets. Changing the translations for events other than key and button events is not expected, and will cause unpredictable behavior. The following actions are provided for using within the vt100 or tek4014 translations resources:

Item	Description
bell ([Percent])	Rings the keyboard bell at the specified percentage above or below the base volume.
ignore ()	Ignores the event but checks for special pointer position escape sequences.
insert ()	Inserts the character or string associated with the key that was pressed.

Item	Description
insert-seven-bit()	Is a synonym for insert() .
insert-eight-bit()	Inserts an 8-bit (meta) version of the character or string associated with the key that was pressed. The exact action depends on the value of the eightBitInput resource.
insert-selection (<i>SourceName</i> [, ...])	Inserts the string found in the selection or cutbuffer indicated by the <i>SourceName</i> parameter. Sources are checked in the order given (case is significant) until one is found. Commonly used selections include PRIMARY, SECONDARY, and CLIPBOARD. Cut buffers are typically named CUT_BUFFER0 through CUT_BUFFER7.
keymap (<i>Name</i>)	Dynamically defines a new translation table whose resource name is <i>Name</i> with the suffix <i>Keymap</i> (case is significant). The name None restores the original translation table.
pop-up menu (<i>MenuName</i>)	Displays the specified popup menu. Valid names (case is significant) include mainMenu, vtMenu, fontMenu, and tekMenu.
secure()	Toggles the Secure Keyboard mode described in the section named " Security " , and is invoked from the securekbd entry in mainMenu.
select-start()	Begins text selection at the current pointer location. See the section entitled " Pointer Usage " for information on making selections.
select-extend()	Tracks the pointer and extends the selection. Only bind this to Motion events.
select-end (<i>DestName</i> [, ...])	Puts the currently selected text into all of the selections or cutbuffers specified by <i>DestName</i> .
select-cursor-start()	Is similar to select-start except that it begins the selection at the current text cursor position.
select-cursor-end (<i>DestName</i> [, ...])	Is similar to select-end except that it should be used with select-cursor-start .
set-vt-font (<i>d/1/2/3/4/5/6/e/s</i> [, <i>NormalFont</i> [, <i>BoldFont</i>]])	Sets the font or fonts currently being used in the VT102 window. The first argument is a single character that specifies the font to be used: <i>d</i> or <i>D</i> indicates the default font (the font initially used when the xterm command was started), <i>1</i> through <i>6</i> indicate the fonts specified by the <i>font1</i> through <i>font6</i> resources, <i>e</i> or <i>E</i> indicates the normal and bold fonts that have been set through escape codes (or specified as the second and third action arguments, respectively), and <i>s</i> or <i>S</i> indicates the font selection (as made by programs such as the xfontsel program) specified by the second action argument.

Item	Description
start-extend()	Is similar to select-start except that the selection is extended to the current pointer location.
start-cursor-extend()	Is similar to select-extend except that the selection is extended to the current text cursor position.
string(String)	Inserts the specified text string as if it had been typed. Quotation is necessary if the string contains white space or nonalphanumeric characters. If the string argument begins with the characters ``0x," it is interpreted as a hex character constant.
scroll-back(Count [,Units])	Scrolls the text window backward so that text that had previously scrolled off the top of the screen is now visible. The <i>Count</i> argument indicates the number of <i>Units</i> (which may be <i>page</i> , <i>halfpage</i> , <i>pixel</i> , or <i>line</i>) by which to scroll.
scroll-forw(Count [,Units])	Scrolls is similar to scroll-back except that it scrolls the other direction.
allow-send-events(On/Off/Toggle)	Sets or toggles the allowSendEvents resource and is also invoked by the allowsends entry in mainMenu.
redraw()	Redraws the window and is also invoked by the redraw entry in mainMenu.
send-signal(SigName)	Sends the signal named by <i>SigName</i> to the xterm subprocess (the shell or program specified with the -e command-line option) and is also invoked by the suspend , continue , interrupt , hangup , terminate , and kill entries in mainMenu. Allowable signal names are (case is not significant): tstp (if supported by the operating system), suspend (same as tstp), cont (if supported by the operating system), int, hup, term, quit, alarm, alarm (same as alm), and kill.
quit()	Sends a SIGHUP to the subprogram and exits. It is also invoked by the quit entry in mainMenu.
Item	Description
set-scrollbar(On/Off/Toggle)	Toggles the scrollbar resource and is also invoked by the scrollbar entry in vtMenu.
set-jumpscroll(On/Off/Toggle)	Toggles the jumpscroll resource and is also invoked by the jumpscroll entry in vtMenu.

Item	Description
set-reverse-video (<i>On/Off/Toggle</i>)	Toggles the reverseVideo resource and is also invoked by the reversevideo entry in vtMenu.
set-autowrap (<i>On/Off/Toggle</i>)	Toggles automatic wrapping of long lines and is also invoked by the autowrap entry in vtMenu.
set-reversewrap (<i>On/Off/Toggle</i>)	Toggles the reverseWrap resource and is also invoked by the reversewrap entry in vtMenu.
set-autolinefeed (<i>On/Off/Toggle</i>)	Toggles automatic insertion of linefeeds and is also invoked by the autolinefeed entry in vtMenu.
set-appcursor (<i>On/Off/Toggle</i>)	Toggles the handling Application Cursor Key mode and is also invoked by the appcursor entry in vtMenu.
set-appkeypad (<i>On/Off/Toggle</i>)	Toggles the handling of Application Keypad mode and is also invoked by the appkeypad entry in vtMenu.
set-scroll-on-key (<i>On/Off/Toggle</i>)	Toggles the scrollKey resource and is also invoked from the scrollkey entry in vtMenu.
set-scroll-on-tty-output (<i>On/Off/Toggle</i>)	Toggles the scrollTtyOutput resource and is also invoked from the scrollttyoutput entry in vtMenu.
set-allow132 (<i>On/Off/Toggle</i>)	Toggles the c132 resource and is also invoked from the allow132 entry in vtMenu.
set-cursesemul (<i>On/Off/Toggle</i>)	Toggles the curses resource and is also invoked from the cursesemul entry in vtMenu.
set-visual-bell (<i>On/Off/Toggle</i>)	Toggles the visualBell resource and is also invoked by the visualbell entry in vtMenu.
set-marginbell (<i>On/Off/Toggle</i>)	Toggles the marginBell resource and is also invoked from the marginbell entry in vtMenu.
set-altscreen (<i>On/Off/Toggle</i>)	Toggles between the alternate and current screens.
soft-reset ()	Resets the scrolling region and is also invoked from the softreset entry in vtMenu.
hard-reset ()	Resets the scrolling region, tabs, window size, and cursor keys and clears the screen. It is also invoked from the hardreset entry in vtMenu.
clear-saved-lines ()	Performs hard-reset (see previous entry) and also clears the history of lines saved off the top of the screen. It is also invoked from the clearsavedlines entry in vtMenu.
set-terminal-type (<i>Type</i>)	Directs output to either the vt or tek windows, according to the <i>Type</i> string. It is also invoked by the tekmode entry in vtMenu and the vtmode entry in tekMenu.
set-visibility (<i>vt/tek, On/Off/Toggle</i>)	Controls whether or not the vt or tek windows are visible. It is also invoked from the tekshow and vthide entries in vtMenu and the vtshow and tekhide entries in tekMenu.

Item	Description
set-tek-text (<i>large/2/3/small</i>)	Sets font used in the Tektronix window to the value of the resources tektextlarge , tektext2 , tektext3 , and tektextsmall according to the argument. It is also by the entries of the same names as the resources in tekMenu.
tek-page ()	Clears the Tektronix window and is also invoked by the tekpage entry in tekMenu.
tek-reset ()	Resets the Tektronix window and is also invoked by the tekreset entry in tekMenu.
tek-copy ()	Copies the escape codes used to generate the current window contents to a file in the current directory beginning with the name COPY . It is also invoked from the tekcopy entry in tekMenu.
visual-bell ()	Flashes the window quickly.

The Tektronix window also has the following action:

Item	Description
gin-press (<i>l/L/m/M/r/R</i>)	Sends the indicated graphics input code.

The default bindings in the VT102 window are:

```
Shift <KeyPress> Prior:    scroll-back(1,halpage) \n\
Shift <KeyPress> Next:    scroll-forw(1,halpage) \n\
Shift <KeyPress> Select:   select-cursor-start \
                           select-cursor-end(PRIMARY,
                           CUT_BUFFER0) \n\
Shift <KeyPress> Insert:   insert-selection(PRIMARY,
                           CUT_BUFFER0) \n\
~Meta<KeyPress>:         insert-seven-bit \n\
Meta<KeyPress>:         insert-eight-bit \n\
!Ctrl <Btn1Down>:        pop-up menu(mainMenu) \n\
!Lock Ctrl <Btn1Down>:    pop-up menu(mainMenu) \n\
~Meta <Btn1Down>:        select-start \n\
~Meta <Btn1Motion>:      select-extend \n\
!Ctrl <Btn2Down>:        pop-up menu(vtMenu) \n\
!Lock Ctrl <Btn2Down>:    pop-up menu(vtMenu) \n\
~Ctrl ~Meta <Btn2Down>:  ignore \n\
~Ctrl ~Meta <Btn2Up>:    insert-selection(PRIMARY,
                           CUT_BUFFER0) \n\
!Ctrl <Btn3Down>:        pop-up menu(fontMenu) \n\
!Lock Ctrl <Btn3Down>:    pop-up menu(fontMenu) \n\
~Ctrl ~Meta <Btn3Down>:  start-extend \n\
~Meta <Btn3Motion>:      select-extend \n\
<BtnUp>:                select-end(PRIMARY, CUT_BUFFER0) \n\
<BtnDown>:              bell(0)
```

The default bindings in the Tektronix window are:

```
~Meta<KeyPress>:         insert-seven-bit \n\
Meta<KeyPress>:         insert-eight-bit \n\
!Ctrl <Btn1Down>:        pop-up menu(mainMenu) \n\
!Lock Ctrl <Btn1Down>:    pop-up menu(mainMenu) \n\
!Ctrl <Btn2Down>:        pop-up menu(tekMenu) \n\
!Lock Ctrl <Btn2Down>:    pop-up menu(tekMenu) \n\
Shift ~Meta<Btn1Down>:   gin-press(L) \n\
~Meta<Btn1Down>:         gin-press(l) \n\
Shift ~Meta<Btn2Down>:   gin-press(M) \n\
~Meta<Btn2Down>:         gin-press(m) \n\
Shift ~Meta<Btn3Down>:   gin-press(R) \n\
~Meta<Btn3Down>:         gin-press(r)
```

The following is an example of how the **keymap** action is used to add special keys for entering commonly typed works:

```

*VT100.Translations:          #override <Key>F13: keymap(dbx)
*VT100.dbxKeymap.translations:
\
    <Key>F14:      keymap(None) \n\
    <Key>F17:      string("next") string(0x0d) \n\
    <Key>F18:      string("step") string(0x0d) \n\
    <Key>F19:      string("continue") string(0x0d) \n\
    <Key>F20:      string("print ")
                    insert-selection(PRIMARY,CUT_BUFFER0)

```

Environment

The **xterm** command sets the environment variables **TERM** and **TERMCAP** properly for the size window you have created. It also uses and sets the **DISPLAY** environment variable to specify which bitmap display terminal to use. The **WINDOWID** environment variable is set to the X window ID number of the xterm window.

Bugs

Large pastes do not work on some systems. This is not a bug in the **xterm** command; it is a bug in the pseudo terminal driver of those systems. The **xterm** command feeds large pastes to the pty only as fast as the pty will accept data, but some pty drivers do not return enough information to know if the write operation has succeeded.

Many of the options are not resettable after the **xterm** command starts.

Only fixed-width, character-cell fonts are supported.

Control Sequences

This section lists control sequences available for the **xterm** command.

Definitions

The following information shows how to interpret key sequences in this section.

Item	Description
c	The literal characters <i>c</i> .
C	A single (required) character.
P_s	A single (usually optional) numeric parameter, composed of one or more digits.
P_m	A multiple numeric parameter composed of any number of single numeric parameters, separated by a ; (semi-colon) character or characters.
P_t	A text parameter composed of printable characters.

VT100 Mode

Most of these control sequences are standard VT102 control sequences, but there are some sequences here from later DEC VT terminals, too. Major VT102 features not supported are smooth scrolling, double-size characters, flashing characters, and VT52 mode.

There are additional control sequences to provide xterm-dependent functions, like the scrollbar or window size. Where the function is specified by DEC or ISO 6429, the code assigned to it is given in parentheses. The escape codes to designate character sets are specified by ISO 2022; see that document for a discussion of character sets.

Control Sequence	Description
BEL	Bell (Ctrl+G)
BS	Backspace (Ctrl+H)

Control Sequence	Description
TAB	Horizontal Tab (HT) (Ctrl+I)
LF	Line Feed or New Line (NL) (Ctrl+J)
VT	Vertical Tab (Ctrl+K) same as LF
FF	Form Feed or New Page (NP) (Ctrl+L) same as LF
CR	Carriage return (Ctrl+M)
SO	Shift Out (Ctrl+N) → Switch to ALternate Character Set: Invokes the G1 character set.
SI	Shift In (Ctrl+O) → Switch to Standard Character Set: Invokes the G0 character set (the default).
ESC # 8	DEC Screen Test (DCECALN)
ESC (C	Designate G0 Character Set (ISO 2022) C = 0 DEC Special Character and Line Drawing Set C = A United Kingdom (UK) C = B United States (USASCII)
ESC) C	Designate G1 Character Set (ISO 2022) C = 0 DEC Special Character and Line Drawing Set C = A United Kingdom (UK) C = B United States (USASCII)
ESC * C	Designate G2 Character Set (ISO 2022) C = 0 DEC Special Character and Line Drawing Set C = A United Kingdom (UK) C = B United States (USASCII)
ESC + C	Designate G3 Character Set (ISO 2022) C = 0 DEC Special Character and Line Drawing Set C = A United Kingdom (UK) C = B United States (USASCII)
ESC 7	Save Cursor (DECSC)
ESC 8	Restore Cursor (DECRC)
ESC =	Application Keypad (DECPAM)
ESC >	Normal Keypad (DECNM)

Control Sequence	Description
ESC D	Index (IND)
ESC E	Next Line (NEL)
ESC H	Tab Set (HTS)
ESC M	Reverse Index (RI)
ESC N	Single Shift Select of G2 Character Set (SS2): Affects next character only.
ESC P	Single Shift Select of G3 Character Set (SS2): Affects next character only.
ESC O P_t ESC \	Device Control String (DCS). xterm implements no DCS functions; P_t is ignored. P_t need not be printable characters.
ESC Z	Return Terminal ID (DECID). Obsolete form of ESC [c (DA)
ESC [P_s @	Insert P_s (Blank) Character of Characters (default=1) (ICH)
ESC [P_s A	Cursor Up P_s Times (default=1) (CUU)
ESC [P_s B	Cursor Down P_s Times (default=1) (CUD)
ESC [P_s C	Cursor Forward P_s Times (default=1) (CUF)

Item	Description
ESC [P_s D	Cursor Backward P_s Times (default=1) (CUB)
ESC [P_s ; P_s H	Cursor Position [row;column] (default=1) (CUP)
ESC [P_s J	Erase in Display (ED) $P_s = 0$ Clear Below (Default) $P_s = 1$ Clear Above $P_s = 2$ Clear All
ESC [P_s K	Erase in Line (EL) $P_s = 0$ Clear to Right (Default) $P_s = 1$ Clear to Left $P_s = 2$ Clear All
ESC [P_s L	Insert P_s Lines (default=1) (IL)
ESC [P_s M	Delete P_s Lines (default=1) (DL)
ESC [P_s P	Delete P_s Characters (default=1) (DCH)

Item	Description
ESC [P_s ; P_s ; P_s ; P_s ; P_s T	Initiate hilite mouse tracking. Parameters are [<i>Func</i> ; <i>Startx</i> ; <i>Starty</i> ; <i>FirstRow</i> ; <i>LastRow</i>]. See Mouse Tracking .
ESC [P_s c	SendDevice Attributes (DA)Delete P_s Characters (default=1) (DCH) $P_s = 0$ or omitted Request attribute from terminal ESC [? 1 ; 2 c ("I am a VT100 with Advanced Video Option.")
ESC [P_s ; P_s f	Horizontal and Vertical Position [row;column] (default = [1,1]) (HVP)
ESC [P_s g	Tab Clear (TBC) $P_s = 0$ Clear Current Column (default) $P_s = 3$ Clear All
ESC [P_m h	Set Modes (SM) $P_s = 4$ Insert Mode (IRM) $P_s = 2 0$ Automatic Newline (LNM)
ESC [P_m l	Reset Modes (RM) $P_s = 4$ Replace Mode (IRM) $P_s = 2 0$ Normal Linefeed (LNM)
ESC [P_m m	Character Attributes (SGR) $P_s = 0$ Normal (default) $P_s = 1$ Bold $P_s = 4$ Underscore $P_s = 5$ Blink (displayed as Bold) $P_s = 7$ Inverse

Item	Description
ESC [P_s n	Device Status Report (DSR) $P_s = 5$ Status Report ESC [0 n ("OK") $P_s = 6$ Report Cursor Position (CPR)[row;column] as ESC [r ; c R $P_s = 20$ Automatic Newline (LNM)
ESC [P_s ; P_s r	Set Scroll Region [top;bottom] (default = fullsize of window) (DECSTBM)
ESC [P_s x	Request Terminal Parameters (DECREQTPARM)

Item	Description
ESC [? P_m h	DEC Private Mode (DECSET) $P_s = 1$ Application Cursor Keys (DECCKM) $P_s = 2$ Designate USASCII for character sets G0–G3. (In VT102, this selects VT52 mode (DECANM), which xterm does not support.) $P_s = 3$ 132 Column Mode (DECCOLM) $P_s = 4$ Smooth (Slow) Scroll (DECSCLM) $P_s = 5$ Reverse Video (DECSCNM) $P_s = 6$ Origin Mode (DECOM) $P_s = 7$ Wraparound Mode (DECAWM) $P_s = 8$ Auto-repeat Keys (DECARM) $P_s = 9$ Set Mouse X and Y on button press. See “Mouse Tracking” on page 1427. $P_s = 3\ 8$ Enter Tektronix Mode (DECTEK) $P_s = 4\ 0$ Allow 80 <—> 132 Mode $P_s = 4\ 1$ curses function fix $P_s = 4\ 4$ Turn On Margin Bell $P_s = 4\ 5$ Reverse Wraparound Mode $P_s = 4\ 7$ Use Alternate Screen Buffer (unless disabled by titelnhibit resource) $P_s = 1\ 0\ 0\ 0$ Set Mouse X and Y on button press and release. See “Mouse Tracking” on page 1427. $P_s = 1\ 0\ 0\ 1$ Use Hilite Mouse Tracking.

Item	Description
ESC [? P_m l	DEC Private Mode Reset (DECRST) $P_s = 1$ Normal Cursor Keys (DECCKM) $P_s = 3$ 80 Column Mode (DECCOLM) $P_s = 4$ Jump Fast Scroll (DECSCLM) $P_s = 5$ Normal Video (DECSCNM) $P_s = 6$ Normal Cursor Mode (DECOM) $P_s = 7$ No Wraparound Mode (DECAWM) $P_s = 8$ No Auto-repeat Keys (DECARM) $P_s = 9$ Do not Send Mouse X and Y on button press. $P_s = 4 0$ Disallow 80 $\leftrightarrow$ 132 Mode $P_s = 4 1$ No curses function fix $P_s = 4 4$ Turn Off Margin Bell $P_s = 4 5$ No Reverse Wraparound Mode $P_s = 4 7$ Use Normal Screen Buffer $P_s = 1 0 0 0$ Do not Send Mouse X and Y on button press and release. $P_s = 1 0 0 1$ Do not Use Hilite Mouse Tracking. xxx
ESC [? P_m r	Restore DEC Private Mode Values. The value of P_s previously saved is restored. P_s values are the same as DECSET.
ESC [? P_m s	Save DEC Private Mode Values. P_s values are the same as DECSET.
ESC]? P_s ; P_t BEL	Set Text Parameters $P_s = 0$ Change Icon Name and Window Title to P_t $P_s = 1$ Change Icon Name to P_t $P_s = 2$ Change Window Title to P_t $P_s = 5 0$ Set Font to P_t

Item	Description
ESC P_t ESC \	Private Message (PM). xterm implements no PM functions; P_t need not be printable characters.
ESC _ P_t ESC \	Application Program Command (APC). Private Message (PM). xterm implements no APC functions; P_t is ignored. P_t need not be printable characters.
ESC c	Full Reset (RIS)
ESC n	Select the G2 Character Set (LS2)
ESC o	Select the G3 Character Set (LS3)
ESC l	Invoke the G3 Character Set as GR (LS3R). Has no visible effect in xterm .
ESC }	Invoke the G2 Character Set as GR (LS2R). Has no visible effect in xterm .
ESC	Invoke the G1 Character Set as GR (LS1R). Has no visible effect in xterm .

XTERM Description Limitation

The xterm terminal description in the DEC.TI file on AIX Version 4 provides underline mode by using the SGR attribute. The SMUL and RMUL attributes are not currently defined in the XTERM terminal description on AIX Version 4. Use the more generic capability named SGR.

```
tput sgr x y
```

Where x is either a 1 or a 0 to turn standout mode on or off respectively, and y is either a 1 or a 0 to turn underline mode on or off respectively. See the article "[terminfo](#) file format" for more details on the SGR capability.

```
tput sgr 0 1    turn off standout; turn on underline
tput sgr 0 0    turn off standout; turn off underline
tput sgr 1 1    turn on standout; turn on underline
tput sgr 1 0    turn on standout; turn off underline
```

Mouse Tracking

The **VT** widget can be set to send the mouse position and other information on button presses. These modes are typically used by editors and other full-screen applications that want to make use of the mouse.

There are three mutually exclusive modes, each enabled (or disabled) by a different parameter in the DECSET (or DECRST) escape sequence. Parameters for all mouse tracking escape sequences generated by the **xterm** command encode numeric parameters in a single character as *value*+040. The screen coordinate system is 1-based.

For example ! is 1. The screen coordinate system is 1-based.

X10 compatibility mode sends an escape sequence on button press encoding the location and the mouse button pressed. It is enabled by specifying parameter 9 to DECSET. On button press, the **xterm** command sends the following "6 characters" . Cb is button-1. Cx and Cy are the x and y coordinates of the mouse when the button was pressed.

ESC [M $C_b C_x C_y$

Normal tracking mode sends an escape sequence on both button press and release. Modifier information is also sent. It is enabled by specifying parameter 1000 to DECSET. On button press or release, the **xterm** command sends the following "key sequence" :

ESC [M C_bC_xC_y

The low two bits of C_b encode button information: 0=MB1 pressed, 1=MB2 pressed, 2=MB3 pressed, 3=release. The upper bits encode what modifiers were down when the button was pressed and are added together. 4=Shift, 8=Meta, 16=Control. C_x and C_y are the x and y coordinates of the mouse event. The upper left corner is (1,1).

Mouse hilite tracking notifies a program of a button press, receives a range of lines from the program, highlights the region covered by the mouse within that range until button release, and then sends the program the release coordinates. It is enabled by specifying parameter 1001 to DECSET.



Attention: Use of this mode requires a cooperating program or it will hang the **xterm** command. On button press, the same information as for normal tracking is generated; the **xterm** command then waits for the program to send mouse tracking information. *All X events are ignored until the following proper escape sequence is received from the pty:*

ESC [P_s; P_s; P_s; P_s; T

The parameters are *Func*, *Startx*, *Starty*, *FirstRow*, and *LastRow*. The *Func* parameter is nonzero to initiate hilite tracking and 0 (zero) to abort. The *Startx* and *Starty* parameters give the starting x and y location for the highlighted region. The ending location tracks the mouse, but is never above row *FirstRow* and is always above row *LastRow*. (The top of the screen is row 1.) When the button is released, the **xterm** command reports the ending position one of two ways: if the start and end coordinates are valid text locations, the **xterm** command reports the "ending position" as follows:

ESC [t C_xC_y

If either coordinate is past the end of the line, the **xterm** command reports the "ending position" as follows:

ESC [T C_xC_yC_xC_yC_xC_y

The parameters are *Startx*, *Starty*, *Endx*, *Endy*, *Mousex*, and *Mousey*. The *Startx*, *Starty*, *Endx*, and *Endy* parameters give the starting and ending character positions of the region. The *Mousex* and *Mousey* parameters give the location of the mouse at button up, which might not be over a character.

Tektronix 4014 Mode

Most of these sequences are standard Tektronix 4014 control sequences. The major features missing are the write-thru and defocused modes. This document does not describe the commands used in the various Tektronix plotting modes but does describe the commands to switch modes.

xwd Command

Purpose

Dumps the image of an Enhanced X-Windows window.

Syntax

```
xwd [ -add Value ] [ -frame ] [ -display Display ] [ -help ] [ -nobdrs ] [ -xy ] [ -out File ] [ -root | -id id |  
-name Name ] [ -icmap ] [ -screen ]
```

Description

The **xwd** command is an Enhanced X-Windows window dumping utility. The **xwd** command allows you to store window images in a specially formatted dump file. This file can then be read by various other X utilities that perform functions such as redisplaying, printing, editing, formatting, archiving, and image processing. Select the target window by clicking the mouse in the desired window. The keyboard bell rings once at the beginning of the dump and twice when the dump is completed.

Flags

Item	Description
-add <i>Value</i>	Specifies a signed value to add to every pixel. This option is specific to X11R5.
-frame	This option indicates that the window manager frame should be included when manually selecting a window.
-display <i>Display</i>	Specifies the server connection.
-help	Prints the usage command syntax summary.
-nobdrs	Specifies that the window dump does not include the pixels that compose the X window border. This is useful if you want to include the window contents in a document as an illustration. The result of the -nobdrs flag depends on which window manager is running. Many window managers remove all borders from the client. For example, the XGetWindowAttributes function returns the value of 0 for the <code>border_width</code> field regardless of the border width when the client was started. Therefore, any border that is visible on the screen belongs to the window manager; the client has no knowledge of it. In this case, the -nobdrs flag has no effect.
-out <i>File</i>	Specifies the output file on the command line. The default is to output to standard out.
-root	Indicates that the root window should be selected for the window dump, without requiring the user to select a window with the pointer. This option is specific to X11R5.
-id <i>id</i>	Indicates that the window with the specified resource id should be selected for the window dump, without requiring the user to select a window with the pointer. This option is specific to X11R5.
-name <i>Name</i>	Indicates that the window with the specified WM_NAME property should be selected for the window dump, without requiring the user to select a window with the pointer. This option is specific to X11R5.
-icmap	Forces the first installed colormap of the screen to be used to obtain RGB values. By default, the colormap of the chosen window is used. This option is specific to X11R5.
-screen	Indicates that the GetImage request used to obtain the image should be done on the root window, rather than directly on the specified window. In this way, you can obtain pieces of the other windows that overlap the specified window and, more importantly, capture menus or other popups that are independent windows but appear over the specified window. This option is specific to X11R5.
-xy	Selects xy format dumping instead of the default z format. This option applies to color displays only.

File

Item	Description
XWDFile.h	X Window dump file format definition file.

xwd Command

Purpose

Retrieves and displays the dumped image of an Enhanced X-Windows window.

Syntax

```
xwd [ -in FileName ] [ -noclick ] [ -geometry Geometry ] [ -display Display ] [ -new ] [ -std MapType ]  
[ -raw ] [ -vis visual_type | visual_id ] [ -help ] [ -rv ] [ -plane Number ] [ -fg Color ] [ -bg Color ]
```

Description

The **xwd** command retrieves the dumped image of an Enhanced X-Windows window. It does so by displaying in a window an image saved in a specially formatted dump file previously produced by the **xwd** command. The dump file format is determined by the **XWDFile.h** file.

You can use flags to specify color display, window size and position, input field, and visual class or identification. You can also select a single bit plane of the image to display.

Flags

Item	Description
-bg <i>Color</i>	Specifies the color to display for the 0 (zero) bits in the image if a bitmap image (or a single plane of an image) is displayed.
-display <i>Display</i>	Specifies the server to connect to; see the X command.
-fg <i>Color</i>	Specifies the color to display for the 1 bits in the image if a bitmap image (or a single plane of an image) is displayed.
-geometry <i>Geometry</i>	Specifies the size and position of the window. Typically, you will only specify the position and let the size default to the actual size of the image.
-help	Prints a short description of the allowable options.
-in <i>FileName</i>	Specifies the input file on the command line. If the input file is not specified, the standard input is assumed.
-new	Creates a new color map for displaying the image. If the image characteristics match those of the display, this flag can display the image on the screen faster, but at the cost of using a new color map (which on most terminals causes other windows to go technicolor).
-noclick	Prevents the application from ending when a button in the window is clicked. You can end the application by typing a q or Q character, or the Ctrl-C key sequence.
-plane <i>Number</i>	Selects a single bit plane of the image to display. Planes are numbered, with 0 (zero) being the least significant bit. Use this flag to determine which plane to pass to the xpr command for printing.

Item	Description
-raw	Displays the dumped image in whatever color values currently exist on the screen. This flag is useful when undumping an image back onto the same screen that the image originally came from, while the original windows are still on the screen. This results in getting the image on the screen faster.
-rv	Swaps the foreground and background colors if a bitmap image (or a single plane of an image) displays. This flag is useful when displaying a bitmap image that has the color sense of pixel values 0 and 1 reversed from what they are on the display.
-std <i>MapType</i>	Uses the specified Standard Colormap to display the image. You can obtain the map type by converting the type to uppercase letters, prepending RGB_ and appending _MAP . Typical map types are best , default , and gray . See the <code>/usr/lpp/X11/Xamples/clients/xstdcmap</code> for information about creating Standard Colormaps.
-vis <i>visual_type</i> <i>visual_id</i>	Specifies a particular visual type or visual id. The default picks the best one or you can specify default, which is the same class as the colormap of the root window. You can specify a particular class: StaticGray, GrayScale, StaticColor, PseudoColor, DirectColor, TrueColor. Specify Match to use the same class as the source image. Specify an exact visual id (specific to the server) as a hexadecimal number (prefixed with 0x) or as a decimal number. This string is not case sensitive.

Environment Variables

Item	Description
DISPLAY	Gets the default display.

Example

To retrieve a specific file from the dump window, enter:

```
xwud -in FileName
```


y

The following AIX commands begin with the letter y.

yacc Command

Purpose

Generates an LALR(1) parsing program from input consisting of a context-free grammar specification.

Syntax

```
yacc [ -b Prefix ] [ -C ] [ -d ] [ -l ] [ -NnNumber ] [ -NmNumber ] [ -NrNumber ] [ -p Prefix ] [ -s ] [ -t ] [ -v ]  
[ -y Path ] Grammar
```

Description

The **yacc** command converts a context-free grammar specification into a set of tables for a simple automaton that executes an LALR(1) parsing algorithm. The grammar can be ambiguous; specified precedence rules are used to break ambiguities.

You must compile the output file, **y.tab.c**, with a C language compiler to produce a **yyparse** function. This function must be loaded with the **yylex** lexical analyzer, as well as with the **main** subroutine and the **yyerror** error-handling subroutine (you must provide these subroutines). The **lex** command is useful for creating lexical analyzers usable by the **yyparse** subroutine. Simple versions of **main** and **yyerror** subroutines are available through the **yacc** library, **liby.a**. Also, **yacc** can be used to generate C++ output.

You can compile the **yacc**-generated C file (**y.tab.c**) with the **-DYACC_MSG** option to include code necessary to use the Message Facility. When you use this option during compilation, error messages generated by the **yyparse** subroutine and the **YYBACKUP** macro are extracted from the **yacc_user.cat** catalog.

This allows you to receive error messages in languages other than English in non-English locales. If the catalog cannot be found or opened, the **yyparse** and **YYBACKUP** subroutines display the default English messages.

The **yacc** command is affected by the **LANG**, **LC_ALL**, **LC_CTYPE**, and **LC_MESSAGES** environment variables.

Flags

Table 30. Flags

Item	Description
-b Prefix	Use <i>Prefix</i> instead of y as the prefix for all output file names. The code file y.tab.c , the header file y.tab.h (created when -d is specified), and the description file y.output (created when -v is specified) are changed to <i>Prefix.tab.c</i> , <i>Prefix.tab.h</i> , and <i>Prefix.output</i> , respectively.
-C	Produces the y.tab.C file instead of the y.tab.c file for use with a C++ compiler. To use the I/O Stream Library for input and output, define the macro, _CPP_IOSTREAMS .
-d	Produces the file y.tab.h . This contains the #define statements that associate the yacc -assigned token codes with your token names. This allows source files other than y.tab.c to access the token codes by including this header file.
-l	Does not include any #line constructs in y.tab.c . Use this only after the grammar and associated actions are fully debugged.

Table 30. Flags (continued)

Item	Description
-NnNumber	Changes the size of the token and nonterminal names array to <i>Number</i> . The default value is 8000. Valid values are only those greater than 8000.
-NmNumber	Changes the size of the memory states array to <i>Number</i> . Default value is 40000. Valid values are only those greater than 40000.
-NrNumber	Changes the internal buffer sizes to handle large grammars. The default value is 2000. Valid values are only those greater than 2000.
-p Prefix	Use <i>Prefix</i> instead of yy as the prefix for all external names created by the yacc command. External names affected include: yychar , yyval , yydebug , yyparse() , yylex() , and yyerror() . (Previously, -p was used to specify an alternate parser; now, -yPath can be used to specify an alternate parser.)
-s	Breaks the yyparse function into several smaller functions. Since its size is somewhat proportional to that of the grammar, it is possible for the yyparse function to become too large to compile, optimize, or execute efficiently.
-t	Compiles run-time debugging code. By default, this code is not included when y.tab.c is compiled. However, the run-time debugging code is under the control of the preprocessor macro, YYDEBUG . If YYDEBUG has a nonzero value, the C compiler (cc) includes the debugging code, regardless of whether the -t flag is used. YYDEBUG should have a value of 0 if you don't want the debugging code included by the compiler. Without compiling this code, the yyparse subroutine will have a faster operating speed. The -t flag causes compilation of the debugging code, but it does not actually turn on the debug mode. To get debug output, the yydebug variable must be set either by adding the C language declaration, <code>int yydebug=1</code> to the declaration section of the yacc grammar file or by setting yydebug through dbx .
-v	Prepares the file y.output . It contains a readable description of the parsing tables and a report on conflicts generated by grammar ambiguities.
-y Path	Uses the parser prototype specified by <i>Path</i> instead of the default /usr/lib/yaccpar file. (Previously, -p was used to specify an alternate parser.)

Exit Status

This command returns the following exit values:

Table 31. Exit status

Ite m	Description
0	Successful completion.
>0	An error occurred.

Examples

1. The following command:

```
yacc grammar.y
```

draws **yacc** rules from the **grammar.y** file, and places the output in **y.tab.c**.

2. The following command:

```
yacc -d grammar.y
```

functions the same as example 1, but it also produces the **y.tab.h** file which would contain C-style **#define** statements for each of the tokens defined in the **grammar.y** file.

Files

Table 32. Files

Item	Description
y.output	Contains a readable description of the parsing tables and a report on conflicts generated by grammar ambiguities.
y.tab.c	Contains an output file.
y.tab.h	Contains definitions for token names.
yacc.tmp	Temporary file.
yacc.debug	Temporary file.
yacc.acts	Temporary file.
/usr/ccs/lib/yaccpar	Contains parser prototype for C programs.
/usr/ccs/lib/liby.a	Contains a run-time library.

yes Command

Purpose

Outputs an affirmative response repetitively.

Syntax

yes [*charstring*]

Description

The **yes** command outputs an affirmative response repetitively. Use the **yes** command as piped input to another command that requires an affirmative response before it completes the specified action. For example, the **yes** command is useful when deleting multiple files from a directory. The Ctl-C key sequence terminates the continuous affirmative responses.

Note: The current locale is determined by the **LC_MESSAGES** environment variable or the *charstring* parameter, if specified. The *charstring* parameter can be any single character or character stream. If you enter an *charstring* parameter after issuing the **yes** command, the *charstring* parameter displays to the screen until you type the Ctl-C key sequence.

Example

To display the word **first** to the screen, type:

```
yes first
```

This statement displays the word until you enter the Ctl-C key sequence.

File

Table 33. File

Item	Description
<code>/usr/bin/yes</code>	Contains the yes command.

ypbind Daemon

Purpose

Enables client processes to bind, or connect, to an NIS server.

Syntax

```
/usr/lib/netsvc/yp/ypbind [ -s -ypset -ypsetme ]
```

Description

The **ypbind** daemon binds, or connects, processes on a Network Information Services (NIS) client to services on an NIS server. This daemon, which runs on every NIS client, is started and stopped by the following System Resource Controller (SRC) commands:

```
startsrc -s ypbind
```

```
stopsrc -s ypbind
```

When a client requests information from a Network Information Services (NIS) map, the **ypbind** daemon broadcasts on the network for a server. When the server responds, it gives the daemon the Internet address and port number of a host. This is the host that provides the information the client is seeking. The **ypbind** daemon stores this address information in the `/var/yp/binding` directory using a file name of **domainname.version**. Then, the next time the client wants to access an NIS map, the client's **ypbind** daemon refers to the addresses in the **domainname.version** file.

The **ypbind** daemon can maintain bindings to several domains and their servers **-ypsetme** simultaneously. The default domain is the one specified by the **domainname** command at startup time.

Note:

1. If a domain becomes unbound (usually when the server crashes or is overloaded), the **ypbind** daemon broadcasts again to find another server.
2. To force a client to bind to a specific server, use the **ypset** command.
3. To find out which server a client is bound to, use the **ypwhich** command.
4. If the `/var/yp/binding/domainname/ypservers` file exists, **ypbind** will attempt to contact the servers listed in that file before broadcasting. The file should contain a list of server IP addresses, one per line.
5. By default, the NIS client will wait indefinitely for the NIS server, during which time, logins to the client system are not possible. It is possible, however, to limit the length of this wait. If the **YPBIND_MAXWAIT** environment variable is set (usually in `/etc/environment`) before the **ypbind** daemon is started, this value (in seconds) will limit the amount of time the NIS client will wait for the NIS server. If this limit is exceeded, the client behaves as if NIS were unavailable and continues using local files. This will allow local logins, such as root.
6. If a domain becomes unbound and it is listed in the `/var/yp/binding/domainname/ypservers` file, by default **ypbind** daemon attempts to contact the server that is currently down; however, if the **YPBIND_SKIP** environment variable is set to 1 (usually set in the `/etc/environment` file) before the **ypbind** daemon is started, the server that is currently down will not be contacted again.

Flags

Table 34. Flags

Item	Description
-s	Runs the ypbind daemon in a secure mode on privileged communications ports.
-ypset	Indicates the local host accepts ypset commands from local or remote hosts.
-ypsetme	Indicates that the local host accepts ypset commands only from the local host. This flag overrides the -ypset flag if both are specified.

Note:

1. If neither the **-ypset** or **-ypsetme** flags are specified, the local host rejects all **ypset** commands from all hosts. This is the most secure mode because the NIS server cannot change.
2. If neither the **-ypset** or **-ypsetme** flags are specified, the local host rejects all **ypset** commands from all hosts. This is the most secure mode because the NIS server cannot change. However, if no NIS servers exist on the networks directly connected to the client machine, then the **-ypsetme** flag must be used and the NIS server should be specified with the **ypset** command.

Files

Table 35. Files

Item	Description
/var/yp/binding directory	Contains Internet addresses and port numbers for NIS servers.
/var/yp/binding/domainname/ypservers	Contains a list of internet addresses, one per line, of servers to attempt to contact before broadcasting.
domainname.version	Binary file that contains the address and port number of the current NIS server.

ypcat Command

Purpose

Prints out a Network Information Services (NIS) map.

Syntax

To Display the Network Information Services Database

```
ypcat [ -k ] [ -t ] [ -d DomainName ] MapName
```

To Display the Nickname Translation Table

```
ypcat -x
```

Description

The **ypcat** command prints out the Network Information Services (NIS) map you specify with the *MapName* parameter. You can specify either a map name or a map nickname. Because the **ypcat** command uses the NIS service, you do not need to specify a server.

Flags

Table 36. Flags

Item	Description
-k	Displays the keys for those maps in which the values are null or for which the key is not part of the value. (None of the maps derived from files that have an ASCII version in the /etc directory fall into this class.)
-t	Indicates that the name specified by the <i>MapName</i> parameter is <i>not</i> a nickname. This flag causes the ypcat command to bypass the nickname translation table and search only for the map specified by the <i>MapName</i> parameter.
-d <i>DomainName</i>	Searches the specified domain for the specified map.
-x	Displays the nickname translation table. This table lists the map nicknames the command knows of and indicates the map name (as specified by the <i>MapName</i> parameter) associated with each nickname.

Examples

1. To look at the networkwide password map, **passwd.byname**, type:

```
ypcat passwd
```

In this example, passwd is the nickname for the **passwd.byname** map.

2. To locate a map, type:

```
ypcat -t passwd
```

In this example, the **ypcat** command bypasses any maps with the nickname of passwd and searches for a map with the full name of passwd.

3. To display a map in another domain, type:

```
ypcat -d polaris passwd
```

In this example, the **ypcat** command locates the map named passwd in the domain named polaris.

4. To display the map nickname translation table, type:

```
ypcat -x
```

In this example, the **ypcat** command displays a list of map nicknames and their associated map names.

ypinit Command

Purpose

Sets up NIS maps on a Network Information Services (NIS) server.

Syntax

To Set up NIS on an NIS Master Server

```
/usr/sbin/ypinit [ -o ] [ -n ] [ -q ] -m [ WorkerName ... ]
```

To Set up NIS on an NIS Worker Server

```
/usr/sbin/ypinit -s MasterName
```

Description

The **ypinit** command sets up NIS maps on a Network Information Services (NIS) master server or NIS worker server. Only users with root user authority can use the **ypinit** command.

By default, the **ypinit** command uses the ASCII system files as input files for the map being created.

Flags

Table 37. Flags

Item	Description
-m [WorkerName...]	Indicates that the local host is to be the NIS master. If the -q flag is used the -m flag can be followed by the names of the machines that will be the NIS worker servers.
-n	Indicates that the ypinit command is not to stop if it finds errors.
-o	Allows any existing maps for the current NIS domain to be overwritten.
-q	Indicates that the ypinit command is to get arguments from the command line instead of prompting for input.
-s MasterName	Copies NIS maps from the server workstation you specify in the <i>MasterName</i> parameter.

Examples

1. To set up an NIS master server that functions as the master for all NIS maps, type the following command on the command line:

```
ypinit -m
```

This command invokes the **make** procedure, which follows the instructions in the **/var/yp/Makefile** file.

2. To set up an NIS worker server, type:

```
ypinit -s zorro
```

In this example, the **ypinit** command copies the NIS maps onto your workstation from the NIS server named **zorro**, making your workstation an NIS worker server.

3. To set up an NIS master server without being prompted for input, type:

```
ypinit -o -n -q -m worker
```

Note: If the system has previously been configured as an NIS master server, ensure that the directory, **/var/yp/binding**, is removed before executing **ypinit**. If old information is stored in **/var/yp/binding**, it may cause errors to occur during configuration of the NIS master server.

Files

Table 38. Files

Item	Description
<u>/etc/bootparams</u>	Lists clients that diskless clients can use for booting.
<u>/etc/passwd</u>	Contains an entry for each user that has permission to log on to the machine.
<u>/etc/group</u>	Contains an entry for each user group allowed to log on to the machine.

Table 38. Files (continued)

Item	Description
<u>/etc/hosts</u>	Contains an entry for each host on the network.
<u>/var/yp/Makefile</u>	Contains rules for making NIS maps.
<u>/etc/networks</u>	Contains the name of each network in the DARPA Internet.
<u>/etc/netmasks</u>	Lists network masks used to implement IP standard subnetting.
<u>/etc/netid</u>	Contains identification information for machines, hosts, and groups.
<u>/etc/rpc</u>	Contains map information for RPC programs.
<u>/etc/services</u>	Contains an entry for each server available through the Internet.
<u>/etc/protocols</u>	Defines Internet protocols used on the local host.
<u>/etc/netgroup</u>	Contains information about each user group on the network.
<u>/etc/ethers</u>	Contains the Ethernet addresses of hosts on the Internet network.
<u>/etc/publickey</u>	Contains public or secret keys for NIS maps.

ypmatch Command

Purpose

Displays the values of given keys within a Network Information Services (NIS) map.

Syntax

To Display Key Values for an NIS Map

```
/usr/bin/ypmatch [ -d Domain ] [ -k ] [ -t ] Key... MapName
```

To Display the NIS Map Nickname Table

```
/usr/bin/ypmatch -x
```

Description

The **ypmatch** command displays the values associated with one or more keys within a Network Information Services (NIS) map. Use the *MapName* parameter to specify either the name or nickname of the map you want to search.

When you specify multiple keys in the *Key* parameter, the system searches the same map for all of the keys. Because pattern matching is not available, match the capitalization and length of each key exactly. If the system does not find a match for the key or keys you specify, a diagnostic message is displayed.

Flags

Table 39. Flags

Item	Description
<u>-d</u> <i>Domain</i>	Specifies a domain other than the default domain.
<u>-k</u>	Prints a key followed by a colon before printing the value of the key. This is useful only if the keys are not duplicated in the values or if you have specified so many keys that the output could be confusing.
<u>-t</u>	Inhibits translation of nickname to map name.

Table 39. Flags (continued)

Item	Description
-x	Displays the map nickname table. This lists the nicknames (as specified by the <i>MapName</i> parameter) the command knows of and indicates the map name associated with each nickname.

Examples

To display the value associated with a particular key, type:

```
ypmatch -d ibm -k host1 hosts
```

In this example, the **ypmatch** command displays the value of the host1 key from the hosts map in the ibm domain.

yppasswd Command

Purpose

Changes your network password in Network Information Services (NIS).

Syntax

```
yppasswd [ -f[Name ] | -s [ Name [ ShellProg ] ] ]
```

Description

The **yppasswd** command changes (or installs) a network password and associates it with the username you specify in the *Name* variable. To create or change a password, you must be the owner of the password you want to change. The NIS password can be different from the one on your own machine. Root users on an NIS server can change the password of another user without knowing the user's original password. To change the password of another user, the Root user enters their password in place of the user's original password. Root users on an NIS client do not have this privilege.

When you enter the **yppasswd** command on the command line, the system prompts you to enter the old password. Then the system prompts you to enter the new password. The password must be a minimum four characters long that includes uppercase and lowercase characters. Otherwise, the password must be six characters long or more. These rules are relaxed if you are insistent enough.

If you enter the old password incorrectly, you must enter the new password before the system gives you an error message. The system requires both passwords because the **update** command sends them to the server at the same time. The server detects the error and notifies you that the old password entered is incorrect.

To verify the new password, the system prompts you to enter it again. For the new password to take effect, the **yppasswdd** daemon must be running on your NIS server.

Note: The **yppasswd** command cannot establish rules for passwords as does the **passwd** command.

Flags

Table 40. Flags

Item	Description
-f [Name]	Changes geocos information of the user <i>Name</i> in the NIS maps. Geocos information is general information that is stored in the <i>/etc/passwd</i> file.

Table 40. Flags (continued)

Item	Description
-s [<i>Name</i> [<i>ShellProg</i>]]	Changes the login shell for user <i>Name</i> in the NIS maps.

Example

1. To change a user's NIS password, enter the following command:

```
yppasswd Joe
```

This example demonstrates how to change the NIS password for the user named Joe. The system first prompts you to enter the old password of Joe and then the new password.

2. To change the login shell to `/bin/ksh` for the username Joe, if the **yppasswdd** daemon is not started with the **-noshell** flag, enter the following command:

```
yppasswd -s Joe /bin/ksh
```

3. To change the `gecos` information in the `passwd` file for the username Joe, if the **yppasswdd** daemon is not started with the **-nogecos** flag, enter the following command:

```
yppasswd -f Joe
Old NIS password:
Joe's current gecoc:
John Doe Test User Id
Change (yes) or (no)? >y
To?>Joe User Test User Id
```

yppasswdd Daemon

Purpose

Receives and executes requests from the **yppasswd** command.

Syntax

```
rpc.yppasswdd FileName [ -nogecos ] [ -nopw ] [ -noshell ] [ -r | -m [ Argument... ] ]
```

Description

The **yppasswdd** daemon is a server that receives and executes requests for new passwords from the **yppasswd** command. These requests require the daemon to verify the user's old password and change it. The daemon changes the password in the file you specify in the *FileName* parameter, which has the same format as the `/etc/passwd` file.

To make it possible to update the Network Information Services (NIS) password map from remote machines, the **yppasswdd** daemon must be running on the master server that contains the NIS password map.

Note: The **yppasswdd** daemon is not run by default, nor can it be started up from the **inetd** daemon like other Remote Procedure Call (RPC) daemons.

The **yppasswdd** daemon can be started and stopped with the following System Resource Controller (SRC) commands:

```
startsrc -s yppasswdd
```

```
stopsrc -s yppasswdd
```

Flags

Item	Description
-m	Runs the make command using the makefile in the <code>/var/yp</code> directory. This adds the new or changed password to the NIS password map. Any arguments that follow the -m flag are passed to the make command.
-nogecos	Indicates the server will not accept changes for gecos information from the yppasswd command.
-nopw	Indicates that the server will not accept password changes from the yppasswdd command.
-noshell	Indicates the server will not accept changes for user shells from the yppasswd command.
-r	Directly updates the <code>/var/yp/domainname/passwd.byname</code> and <code>/var/yp/domainname/passwd.byuid</code> database files on the Master server as well as any Worker servers with new or changed passwords. This option is faster than the -m flag because the make command is not run. The -r flag is useful when the database files are large (several thousand entries or more).

Note: The System Resource Controller (SRC) starts the **yppasswdd** daemon with the **-m** flag specified by default. Use the **chssys** command to change the default to the **-r** flag.

Example

To propagate updated passwords immediately, invoke the **yppasswdd** daemon as follows:

```
startsrc -s yppasswdd
```

Files

Item	Description
<u>/etc/inetd.conf</u>	Defines how the inetd daemon handles Internet service requests.
<u>/var/yp/Makefile</u>	Contains rules for making NIS maps.
<u>/etc/rc.nfs</u>	Contains the startup script for the NFS and NIS daemons.
<u>/etc/security/passwd</u>	Stores password information.

yppoll Command

Purpose

Displays the order number (ID number) of the Network Information Services (NIS) map currently in use on the server.

Syntax

```
/usr/sbin/yppoll [ -h Host ] [ -d Domain ] MapName
```

Description

The **yppoll** command uses the **ypserv** daemon to display the order number of the map you specify in the *MapName* parameter. An order number is a map's ID number and is assigned by the system. This number

changes whenever a map is updated. Use the **yppoll** command whenever you want to make sure your servers are using the most current version of a particular map.

The **yppoll** command can run on systems that have either version 1 or version 2 of the Network Information Services (NIS) protocol installed. Be aware, however, that each version of the protocol has its own set of diagnostic messages.

Note: When specifying a *MapName*, be sure to enter the map's full name. The **yppoll** command does not recognize map nicknames.

Flags

Table 41. Flags

Item	Description
-h Host	Enables you to specify a server other than the default server. To find out which server the command defaults to, use the ypwhich command.
-d Domain	Enables you to specify a domain other than the default domain. To find out which domain the command defaults to, use the domainname command.

Examples

1. To look at a map located on a particular host, type:

```
/usr/sbin/yppoll -h thor netgroups.byuser
```

In this example, the **yppoll** command displays the order number for the `netgroups.byuser` map located on the host named `thor`.

2. To look at a map on a domain, type:

```
/usr/sbin/yppoll -d atlantis hosts.byname
```

In this example, the **yppoll** command displays the order number for the `hosts.byname` map located in the domain `atlantis`.

yppush Command

Purpose

Prompts the Network Information Services (NIS) worker servers to copy updated NIS maps.

Syntax

```
/usr/sbin/yppush [ -v ] [ -d Domain ] MapName
```

Description

The **yppush** command, which is issued from the `/usr/etc/yp` directory, prompts the Network Information Services (NIS) worker servers to make copies of updated NIS maps. The *MapName* variable specifies that map to be transferred to the worker servers of the master servers. To get a list of the servers it needs to prompt, the **yppush** command reads the **ypservers** map, specified by the *Domain* parameter or the current default domain. When prompted, each worker server uses the **ypxfr** command to copy and transfer the map back to its own database.

You can use the System management interface tool (SMIT) to run this command. To use SMIT, type:

```
smit yppush
```

Note: If your system uses version 1 of the NIS protocol, the **ypxfr** command is not the transfer agent.

Flags

Table 42. Flags

Item	Description
-d Domain	Specifies a domain other than the default domain. The maps for the specified domain must exist.
-v	Displays messages as each server is called and then displays one message for each server's response, if you are using the version 2 protocol. If this flag is omitted, the command displays error messages only. Note: Version 1 of the NIS protocol does not display messages. If your system uses version 1, use the yppoll command to verify that the transfer took place.

Examples

1. To copy a map from another domain to the worker servers, type:

```
/usr/sbin/yppush -d atlantis netgroup
```

In this example, the **yppush** command copies the **netgroup** map from the **atlantis** domain.

2. To display the in-progress status of the **yppush** command as it calls each worker server, type:

```
/usr/sbin/yppush -v -d atlantis netgroup
```

In this example, the **yppush** command displays in-progress messages as it copies the **netgroup** map from the **atlantis** domain onto each of the network's worker servers.

Files

Table 43. Files

Item	Description
/var/yp/DomainName/ypservers.{dir, pag}	Lists servers that the yppush command prompts to make copies of updated NIS maps.

ypserv Daemon

Purpose

Looks up information in local Network Information Services (NIS) maps.

Syntax

```
/usr/lib/netsvc/yp/ypserv
```

Description

The **ypserv** daemon looks up information in its local Network Information Services (NIS) maps. The operations performed by the **ypserv** daemon are defined for the implementor by the NIS Protocol Specification and for the programmer by the **/usr/include/rpcsvc/yp_prot.h** header file. Communication with the **ypserv** daemon is by means of Remote Procedure Calls (RPC).

The **ypserv** daemon runs only on server machines. The **ypserv** daemon is started and stopped by the following System Resource Controller (SRC) commands:

```
startsrc -s ypserv
```

```
stopsrc -s ypserv
```

The **ypserv** daemon performs the following operations on a specified map within an NIS domain:

Table 44. The ypserv daemon operations

Item	Description
Match	Takes a key and returns the associated value.
Get_first	Returns the first key-value pair from the map.
Get_next	Enumerates the next key-value pair in the map.
Get_all	Ships the entire NIS map to a requestor in response to a single RPC request.
Get_order_number	Supplies information about a map instead of map entries. The order number actually exists in the map as a key-value pair, but the server does not return it through the normal lookup functions. However, the pair will be visible if you examine the map with the makedbm command.
Get_master_name	Supplies information about a map instead of map entries. The master name actually exists in the map as a key-value pair, but the server does not return it through the normal lookup functions. However, the pair will be visible if you examine the map with the makedbm command.

Log information is written to the **/var/yp/ypserv.log** file if it exists when the **ypserv** daemon starts running.

If the **/var/yp/securenets** file exists, the **ypservr** command only responds to hosts within the ip range specified in this file.

Files

Item	Description
/etc/rc.nfs	Contains the startup script for the NFS and NIS daemons.
/var/yp/ypserv.log	Contains the log for the ypserv daemon.

ypset Command

Purpose

Directs a client machine to a specific server.

Syntax

```
/usr/sbin/ypset [ -V1 ] [ -d Domain ] [ -h Host ] Server
```

Description

The **ypset** command directs the **ypbind** daemon on the client to the **ypserv** daemon on the server. The **ypbind** daemon goes to the server you specify in the *Server* parameter to get Network Information

Services (NIS) services for the domain you specify in the *Domain* parameter. The **ypbind** daemon gets the NIS services from the **ypserv** daemon on the server.

After the binding is set, it is not tested until a client process (such as the **ypcat** command or the **ypwhich** command) tries to get a binding for the domain. If the attempt to bind fails (the specified server is down or is not running the **ypserv** daemon), the **ypbind** daemon makes another attempt to bind for the same domain.

Specify either a name or an Internet Protocol (IP) address in the *Server* parameter. If you specify a name, the **ypset** command attempts to resolve the name to an IP address through the use of the NIS service. This works only if your machine has a current valid binding for the domain in question. In most cases, you should specify the server as an IP address.

In cases where several hosts on the local network are supplying NIS services, the **ypbind** daemon can rebind to another host. If a server is down or is not running the **ypserv** daemon, the **ypbind** daemon rebinds the client to another server. In this way, the network information service balances the load among the available NIS servers.

Use the **ypset** command if the network:

- Does not support broadcasting.
- Supports broadcasting but does not have an NIS server.
- Accesses a map that exists only on a particular NIS server.

An alternative to using **ypset** is to use the `/var/yp/binding/domain_name/ypservers` file. This file, if present, should contain a list of NIS servers to attempt to bind to, one server per line. If the **ypbind** daemon cannot bind to any of the servers in the **ypservers** file, then it will attempt to use the server specified by **ypset**. If that fails, it will broadcast on the subnet for a NIS server.

Flags

Table 45. Flags

Item	Description
-d <i>Domain</i>	Specifies a domain other than the default domain.
-h <i>Host</i>	Sets the binding for the ypbind daemon on the specified host instead of on the local host. The host can be specified as a name or as an IP address.
-v1	Binds the specified server for the (old) version 1 NIS protocol.

Example

To set a server to bind on a host in a particular domain, enter:

```
ypset -d ibm -h venus mars
```

In this example, the **ypset** command causes the host named venus to bind to the server named mars.

ypupdated Daemon

Purpose

Updates information in Network Information Services (NIS) maps.

Syntax

```
/usr/lib/netsvc/yp/rpc.yupdated [ -i | -s ]
```

Description

The **ypupdated** daemon updates information in Network Information Services (NIS) maps. Before it can update information, however, the daemon consults the **updaters** file in the **/var/yp** directory to determine which NIS maps should be updated and how they should be changed.

By default, the **ypupdated** daemon requires the most secure method of authentication available to it, either DES (secure) or UNIX (insecure).

The **ypupdated** daemon is started and stopped by the following System Resource Controller (SRC) commands:

```
startsrc -s ypupdated
stopsrc -s ypupdated
```

Flags

Table 46. Flags

Item	Description
-s	Accepts only calls authenticated using the secure Remote Procedure Call (RPC) mechanism (AUTH_DES authentication). This disables programmatic updating of NIS maps unless the network supports these calls.
-i	Accepts RPC calls with the insecure AUTH_UNIX credentials. This allows programmatic updating of NIS maps in all networks.

Examples

To start the **ypupdated** daemon from the command line, type:

```
startsrc -s ypupdated
```

File

Table 47. File

Item	Description
<u>/var/yp/updaters</u>	A makefile for updating NIS maps.

ypwhich Command

Purpose

Identifies either the Network Information Services (NIS) server or the server that is the master for a given map.

Syntax

To Identify the NIS Server

```
ypwhich [ -d Domain ] [ -V1 | -V2 ] [ HostName ]
```

To Identify the Master NIS Server for a Map

```
/usr/bin/ypwhich [ -t ] [ -d Domain ] [ -m [ MapName ] ]
```


To Display the Map Nickname Table

```
/usr/bin/ypwhich -x
```

Description

The **ypwhich** command identifies which server supplies Network Information Services (NIS) services or which server is the master for a map, depending on how the **ypwhich** command is invoked. If invoked without arguments, this command displays the name of the NIS server for the local machine. If you specify a host name, the system queries that host to find out which master it is using.

Flags

Table 48. Flags

Item	Description
-d <i>Domain</i>	Uses the specified domain instead of the default domain.
-V1	Indicates which server is serving the old version 1 NIS protocol client processes.
-V2	Indicates which server is serving the current version 2 NIS protocol client processes. If neither version is specified, the ypwhich command attempts to locate the server that supplies the version 2 services. If there is no version 2 server currently bound, the ypwhich command then attempts to locate the server supplying version 1 services. Because servers and clients are both backward-compatible, the user need seldom be concerned about which version is currently in use.
-t	Inhibits nickname translation, which is useful if there is a map name identical to a nickname.
-m <i>MapName</i>	Finds the master NIS server for a map. No host can be specified with the -m flag. The <i>MapName</i> variable can be a map name or a nickname for a map. When the map name is omitted, the -m flag produces a list of available maps.
-x	Displays the map nickname table. This lists the nicknames (<i>MapName</i>) the command knows of and indicates the map name associated with each nickname.

Examples

1. To find the master server for a map, type:

```
ypwhich -m passwd
```

In this example, the **ypwhich** command displays the name of the server for the passwd map.

2. To find the map named passwd, rather than the map nicknamed passwd, type:

```
ypwhich -t -m passwd
```

In this example, the **ypwhich** command displays the name of the server for the map whose full name is passwd.

3. To find out which server serves clients that run the old version 1 of the NIS protocol, type:

```
ypwhich -V1
```

4. To display a table of map nicknames, type:

```
ypwhich -x
```

ypxfr Command

Purpose

Transfers a Network Information Services (NIS) map from an NIS server to a local host.

Syntax

```
/usr/sbin/ypxfr [ -f ] [ -c ] [ -d Domain ] [ -h Host ] [ -s Domain ] [ -C TID Program IPAddress Port ] [ -S MapName ]
```

Description

The **ypxfr** command transfers a Network Information Services (NIS) map from an NIS server to the local host as follows:

1. Creates a temporary map in the **/var/yp/Domain** directory (which must already exist) on the client.
2. Fetches the map entries from the server and fills in the map on the client, one at a time.
3. Gets and loads the map parameters (order number and server).
4. Deletes any old versions of the map.
5. Moves the temporary map to the real map name.

If the **/var/yp/securenets** file exists, the **ypxfr** command only responds to hosts that are listed in this file.

The *MapName* variable specifies the name of a map that will be transferred from an NIS server.

If run interactively, the **ypxfr** command sends output to the user's terminal. If invoked without a controlling terminal, the **ypxfr** command appends its output to the **/var/yp/ypxfr.log** file (if the file already exists). This file records each transfer attempt and its results. The **ypxfr** command is most often invoked from the root user's **crontab** file or by the **ypserv** daemon.

To maintain consistent information between servers, use the **ypxfr** command to update every map in the NIS database periodically. Be aware though that some maps change more frequently than others and therefore need to be updated more frequently. For instance, maps that change infrequently, such as every few months, should be updated at least once a month. Maps that change frequently, such as several times a day, should be checked hourly for updates. The **services.byname** map, for example, may not change for months at a time, while the **hosts.byname** map may change several times a day.

To perform periodic updates automatically, use a **crontab** entry. To update several maps at one time, group commands together in a shell script. Examples of a shell script can be found in the **/usr/etc/yp** directory in the following files: **ypxfr_1perday**, **ypxfr_2perday**, **ypxfr_1perhour**.

You can use the System management interface tool (SMIT) to run this command. To use SMIT, enter:

```
smit ypxfr
```

Flags

Table 49. Flags

Item	Description
-C <i>TID Program IPAddress Port</i>	Tells the ypxfr command where to find the yppush command. The ypserv daemon invokes the ypxfr command to call back a yppush command to the host. Use the parameters to indicate the following: TID Specifies the transaction ID of the yppush command. Program Specifies the program number associated with the yppush command. IPAddress Specifies the Internet Protocol address of the port where the yppush command resides. Port Specifies the port that the yppush command is listening on. Note: This option is only for use by the ypserv daemon.
-c	Prevents sending of a request to Clear Current Map to the local ypserv daemon. Use this flag if the ypserv daemon is not running locally at the time you are running the ypxfr command. Otherwise, the ypxfr command displays an error message and the transfer fails.
-d <i>Domain</i>	Specifies a domain other than the default domain. The maps for the specified domain must exist.
-f	Forces the transfer to occur even if the version at the master is not more recent than the local version.
-h <i>Host</i>	Gets the map from host specified, regardless of what the map says the master is. If a host is not specified, the ypxfr command asks the NIS service for the name of the master and tries to get the map from there. The <i>Host</i> variable can contain a name or an Internet address in the form <i>a.b.c.d</i>.
-S	Requires the ypserv server, from which it obtains the maps to be transferred, use <i>privileged</i> IP ports. Because only root user processes are typically allowed to use privileged ports, this feature adds an extra measure of security to the transfer. If the map being transferred is a secure map, the ypxfr command sets the permissions on the map to 0600.
-s <i>Domain</i>	Specifies a source domain from which to transfer a map that should be the same across domains (such as the services.byname map).

Examples

To get a map from a host in another domain, enter:

```
/usr/sbin/ypxfr -d ibm -h venus passwd.byname
```

In this example, the **ypxfr** command gets the `passwd.byname` map from the host name `venus` in the `ibm` domain.

Files

Table 50. Files

Item	Description
/var/yp/ypxfr.log	Contains the log file.
/usr/sbin/ypxfr_1perday	Contains the script to run one transfer each day, for use with the cron daemons.
/usr/sbin/ypxfr_2perday	Contains the script to run two transfers each day.
/usr/sbin/ypxfr_1perhour	Contains the script for hourly transfers of volatile maps.

Z

The following AIX commands begin with the letter z.

zcat Command

Purpose

Expands a compressed file to standard output.

Syntax

zcat [**-n**] [**-V**] [*File ...*]

Description

The **zcat** command allows the user to expand and view a compressed file without uncompressing that file. The **zcat** command does not rename the expanded file or remove the **.Z** extension. The **zcat** command writes the expanded output to standard output.

Flags

Table 51. Flags

Item	Description
-n	Omits the compressed file header from the compressed file. Note: Use the -n option if the file was compressed using the -n option.
-V	Writes the current version and compile options to standard error.

Parameters

Table 52. Parameters

Item	Description
<i>File ...</i>	Specifies the compressed files to expand.

Return Values

If the **zcat** command exits with a status of 1 if any of the following events occur:

- The input file was not produced by the **compress** command.
- An input file cannot be read or an output file cannot be written.

If no error occurs, the exit status is 0.

Exit Status

Table 53. Exit status

Item	Description
0	Successful completion.

Table 53. Exit status (continued)

Item	Description
>0	An error occurred.

Examples

To view the foo.Z file without uncompressing it, enter:

```
zcat foo.Z
```

The uncompressed contents of the foo.Z file are written to standard output. The file is not renamed.

zdump Command

Purpose

Displays the time zone information.

Syntax

```
zdump [--version] [--help] [-v] [-V] [-c [loyear,]hiyear] [-t [lotime,]hitime] [ -i ] zonename ...
```

Description

The **zdump** command prints the time zone information for each zone name that is specified in the command line.

Flags

Table 54. Flags

Item	Description
-c [loyear,]hiyear	Reports the verbose output for the specified years. Cutoff time is computed by using the proleptic Gregorian calendar with year 0 and Universal Time (UT) ignoring leap seconds. The smaller year attribute (<i>loyear</i>) is exclusive and the higher year attribute (<i>hiyear</i>) is inclusive. For example, a <i>loyear</i> value of 1970 excludes a transition that occurs at 1970-01-01 00:00:00 Coordinated Universal Time, but a <i>hiyear</i> value of 1970 includes the transition. The default cutoff time is -500,2500.
> - -help <	> Displays a short usage message and exits. <
> -i <	> Provides a list of all the transitions and displays a description of the time intervals. For each time zone on the command line, the -i flag displays an interval format description of the time zone. For more information, see the Interval format section. <

Table 54. Flags (continued)

Item	Description
-t [<i>lotime</i> ,] <i>hitime</i>	Reports verbose output for a particular time that is specified in the <i>lotime</i> and <i>hitime</i> attributes in the syntax of <i>yyyy-mm-dd hh:mm:ss</i> Coordinated Universal Time (UTC). For example, 1970-01-01 00:00:00. The cutoff time includes the leap seconds depending on the <i>zonename</i> parameter. Similar to the -c flag, the smaller time limit is exclusive and the higher time limit is inclusive.
-v	For each zone name that is specified in the command line, the zdump command prints the following items: • The lowest possible value of time. • The time at one day after the lowest possible value of time. • The times both one second before and exactly at each detected discontinuity of time. • The highest possible value of time. • The time at one day before the highest possible value of time. Each line is followed by <i>isdst=D</i> , where <i>D</i> is a positive, zero, or negative value that depends on whether the specified time is Daylight Saving Time, standard time, or an unknown time type. Each line is also followed by <i>gmtoff=N</i> , where the specified local time is <i>N</i> seconds east of Greenwich.
-V	Prints an output that is similar to the output of the -v flag except that this flag does not display the times that are relative to the extreme time values. This flag generates output that is simpler to compare with different time representations.
--version	Displays the version information of the zdump command.

Parameters

Table 55. Parameters

Item	Description
<i>zonename</i>	Represents the name of the zone of which the time zone information is displayed.

Interval format

>|

The interval format is a compact text representation of transitions for a specified time zone that is both human and machine readable. The interval format representation starts with a line *TZ=string*, where *string* is the time zone for which the transition data is displayed. The time zone information for each transition is displayed in a tabular format with the date, time, and interval columns. The first entry in the table displays a value only for the interval column as it refers to the time interval before first transition.

The values of the date column are in *yyyy-mm-dd* format, and the values of the time column are in *hh:mm:ss* format. The time is the local time immediately after the transition. The interval column consists of a UT offset in $\pm$ hhmmss format, a time zone abbreviation, and an *isdst* flag. A positive value of the UT offset indicates that the time zone is in the east of UT, and a negative value indicates that the time zone is in the west of UT. The UT offset value of -00 denotes a UT placeholder in areas where the actual offset is not specified. The value of UT offset is -00 when the UT offset is zero and the time zone abbreviation begins with - or is zzz. The time zone abbreviation that is equal to the UT offset is omitted.

For the standard time, the `isdst` flag is omitted. The value of the `isdst` flag is a positive decimal integer (usually 1) for the daylight saving and is a negative decimal integer for an unknown value.

The following examples shows the interval format representation of transitions for a specified time zone:

```
TZ="Pacific/Honolulu"
- - -103126 LMT
1896-01-13 12:01:26 -1030 HST
1933-04-30 03 -0930 HDT 1
1933-05-21 11 -1030 HST
1942-02-09 03 -0930 HWT 1
1945-08-14 13:30 -0930 HPT 1
1945-09-30 01 -1030 HST
1947-06-08 02:30 -10 HST
```

The `TZ="Pacific/Honolulu"` indicates that the time zone is Pacific/Honolulu. The first row entry in the interval format representation indicates that the local time is 10 hours, 31 mins, and 26 secs west of UT and the standard time abbreviation is LMT. The next row indicates that the date of transition is 13 January 1896, the time of transition is 12:01:26, the time interval is 10 hours, 30 mins west of UT, and the standard time abbreviation is HST. The `isdst` flag value of 1 in the third entry indicates Daylight Saving Time.

[|<](#)

Exit Status

0

The command was completed successfully.

>0

An error occurred.

Examples

1. To report time zone information for Singapore, enter the following command:

```
zdump -v Singapore
```

2. To display time zone information for Türkiye stopping near the start of the year 2035, enter the following command:

```
zdump -v -c 2035 Turkey
```

3. To report the time zone information for New York during the years 2015 - 2017, enter the following command:

```
zdump -v -c 2015,2017 America/New_York
```

The output is similar to the following sample:

```
America/New_York Fri Dec 13 20:45:52 1901 UT = Fri Dec 13 15:45:52 1901 EST isdst=0
gmtoffset=-18000
America/New_York Sat Dec 14 20:45:52 1901 UT = Sat Dec 14 15:45:52 1901 EST isdst=0
gmtoffset=-18000
America/New_York Sun Mar 8 06:59:59 2015 UT = Sun Mar 8 01:59:59 2015 EST isdst=0
gmtoffset=-18000
America/New_York Sun Mar 8 07:00:00 2015 UT = Sun Mar 8 03:00:00 2015 EDT isdst=1
gmtoffset=-14400
America/New_York Sun Nov 1 05:59:59 2015 UT = Sun Nov 1 01:59:59 2015 EDT isdst=1
gmtoffset=-14400
America/New_York Sun Nov 1 06:00:00 2015 UT = Sun Nov 1 01:00:00 2015 EST isdst=0
gmtoffset=-18000
America/New_York Sun Mar 13 06:59:59 2016 UT = Sun Mar 13 01:59:59 2016 EST isdst=0
gmtoffset=-18000
America/New_York Sun Mar 13 07:00:00 2016 UT = Sun Mar 13 03:00:00 2016 EDT isdst=1
gmtoffset=-14400
America/New_York Sun Nov 6 05:59:59 2016 UT = Sun Nov 6 01:59:59 2016 EDT isdst=1
gmtoffset=-14400
```



```
America/New_York Sun Nov 6 06:00:00 2016 UT = Sun Nov 6 01:00:00 2016 EST isdst=0
gmtoff=-18000
America/New_York Mon Jan 18 03:14:07 2038 UT = Sun Jan 17 22:14:07 2038 EST isdst=0
gmtoff=-18000
America/New_York Tue Jan 19 03:14:07 2038 UT = Mon Jan 18 22:14:07 2038 EST isdst=0
gmtoff=-18000
```

- To report the time zone information for New York during the years 2015 - 2017 without the lowest and highest time values, enter the following command:

```
zdump -V -c 2015,2017 America/New_York
```

The output is similar to the following sample:

```
America/New_York Sun Mar 8 06:59:59 2015 UT = Sun Mar 8 01:59:59 2015 EST isdst=0
gmtoff=-18000
America/New_York Sun Mar 8 07:00:00 2015 UT = Sun Mar 8 03:00:00 2015 EDT isdst=1
gmtoff=-14400
America/New_York Sun Nov 1 05:59:59 2015 UT = Sun Nov 1 01:59:59 2015 EDT isdst=1
gmtoff=-14400
America/New_York Sun Nov 1 06:00:00 2015 UT = Sun Nov 1 01:00:00 2015 EST isdst=0
gmtoff=-18000
America/New_York Sun Mar 13 06:59:59 2016 UT = Sun Mar 13 01:59:59 2016 EST isdst=0
gmtoff=-18000
America/New_York Sun Mar 13 07:00:00 2016 UT = Sun Mar 13 03:00:00 2016 EDT isdst=1
gmtoff=-14400
America/New_York Sun Nov 6 05:59:59 2016 UT = Sun Nov 6 01:59:59 2016 EDT isdst=1
gmtoff=-14400
America/New_York Sun Nov 6 06:00:00 2016 UT = Sun Nov 6 01:00:00 2016 EST isdst=0
gmtoff=-18000
```

Files

Table 56. Files

Item	Description
/usr/sbin/zdump	Contains the SystemV zdump command.
/usr/share/lib/zoneinfo	Contains the standard time zone directory.

zic Command

Purpose

The **zic** command provides a time zone compiler.

Syntax

```
zic [ -v ] [ -p Posixrules ] [ -d Directory ] [ -L Leapseconds ] [ -b bloat ] [ -r [@lo][@hi] ] [ -R [@hi] ] [ --version ] [ --help ] [ FileName ... ]
```

Description

The **zic** command processes text from the files that are named on the command line and creates the time conversion binary files that are specified in this input. The time zone information is processed from the standard input if file name is specified as - (hyphen).

Input lines in the specified files are made up of fields. Field separators can be any number of white-space characters. A pound sign (#) in the input file indicates a comment that extends to the end of the line on which the pound sign appears. White-space characters and pound signs can be enclosed in double quotation marks (") if they are to be used as part of a field. Blank lines are ignored.

A rule line has the form:

Rule	NAME	FROM	TO	TYPE	IN	ON	AT	SAVE	LETTER/S

Following are the fields that make up the rule line:

NAME

This field provides a random name for the set of rules for which this Rule is applicable.

FROM

This field provides the first year in which the rule applies. Specifying **minimum** (min) indicates the minimum year with a representable time value. Specifying **maximum** (max) indicates the maximum year with a representable time value.

TO

This field provides the final year in which the rule applies. This option is indicated as a valid time value or by specifying a **minimum** and **maximum**. Specifying **only** is used to repeat the value of the **FROM** field.

TYPE

This field provides the type of year in which the rule applies.

The **TYPE** field has the following values:

'.'

The rule applies in all years between **FROM** and **TO**, inclusively.

uspres

The rule applies in US Presidential election years.

nonpres

The rule applies in years other than US Presidential election years.

IN

This field represents the month in which the rule takes effect. Month names can be abbreviated.

ON

This field represents the day on which the rule takes effect. Following are the recognized forms:

- **lastFri** represents the last Friday in the month.
- **lastMon** represents the last Monday in the month.
- A number that represents the day of the month. For example, 5 represents the fifth of the month.
- **lastSun** represents the last Sunday in the month.
- **Sun>=8** represents the first Sunday on or after the eighth.
- **Sun<=25** represents the last Sunday on or before the 25th.

Names of days of the week can be abbreviated or spelled out in full. Spaces are not supported within the **ON** field.

AT

Represents the time of day at which the rule takes effect. Following are the recognized forms:

- A number that represents time in hours. For example, 2 indicates two hours.
- 2:00 indicates two AM in hours and minutes.
- 15:00 represents three PM in the afternoon by using the 24-hour format time format.
- 1:28:14 indicates one hour, twenty-eight minutes, and fourteen seconds, by using the hours, minutes, seconds format.

Any of these forms can be followed by the letter **w** if the given time is local wall-clock time or **s** if the given time is local standard time. In the absence of **w** or **s**, wall-clock time is assumed.

Regions with more than two types of local time are required to use the local standard time in the **AT** field of the earliest transition time's rule to ensure the accuracy of the earliest transition time that is stored in the resulting time-zone binary.

SAVE

This field represents the amount of time to be added to local standard time when the rule is in effect. This field has the same format as the **AT** field. The **w** and **s** suffixes are not valid with this field.

LETTER/S

This field provides the variable part of the time-zone abbreviations that are used when this rule is in effect. When this field contains - (hyphen), the variable is null. The **S** character is used to indicate **EST** and the **D** character is used to indicate **EDT**.

A zone line has the form:

Zone	NAME	GMTOFF	RULES/SAVE	FORMAT	[UNTIL]

Following are the fields that make up a zone line:

NAME

Indicates the name of the time zone. The name that is specified by the **NAME** field is the name that is used to create a file that contains time conversion information for the zone.

GMTOFF

Indicates the amount of time to add to GMT to get standard time in this zone. This field has the same format as the **AT** and **SAVE** fields of rule lines. Begin the field with a minus sign if time must be subtracted from GMT.

RULES/SAVE

Indicates the name of the rules that apply in the time zone or alternately, an amount of time to add to local standard time. If value of this field is - (hyphen), then standard time always applies in the time zone.

FORMAT

Indicates the format for time zone abbreviations in this time zone. The **%s** characters are used to show where the variable part of the time zone abbreviation goes.

UNTIL

Indicates the time at which the GMT offset or the rules change for a location. It is specified as year, month, day, and time of day. If this field is specified, the time zone information is generated from the specified GMT offset and rule change until the time specified.

The next line must be a continuation line. The continuation line places information that starts at the time that is specified in the **UNTIL** field of the previous line into the file that is used by the previous line. This line has the same format as a zone line, except that the **Zone** string and the name are omitted. Continuation lines can contain a **UNTIL** field, just as zone lines do, indicating that the next line is a further continuation.

A link line has the form:

Link	LINK-FROM	LINK-TO

The **LINK-FROM** field must appear as the **NAME** field in a zone line; the **LINK-TO** field is used as an alternate name for that zone.

Except for continuation lines, lines can appear in any order in the input.

The **zic** command has a limitation of compiling input that contains a date before 14 December 1901 because dates before this time cannot be represented by a 32-bit `time_t` data type.

Flags

Table 57. Flags

Item	Description
> -b <i>bloat</i> <	> Provides binary data as specified by the <i>bloat</i> variable that is compatible with the earlier versions of software. Following are the valid values for the <i>bloat</i> variable: fat The fat option generates additional data entries in the binary file. These entries address potential bugs or incompatibilities in an older version of software. For example, an older version of software that incorrectly handles the 64-bit generated data. slim The slim option generates a smaller binary file that helps to identify bugs or incompatibilities in an older version of software. This binary file does not work around the bugs or incompatibilities. The default value for the <i>bloat</i> variable is the slim option. Note: You can also change the size of the binary file by using the -r option. <
-d <i>Directory</i>	Creates files that contain time conversion information in the <i>Directory</i> directory, instead of the /usr/share/lib/zoneinfo/ standard directory.
> --help <	> Displays a short usage message and exits. <
-l <i>TimeZone</i>	Uses the <i>TimeZone</i> time zone as local time. The zic command acts as if the file contained a link similar to the following link: <pre>Link timezone localtime</pre>
-L <i>Leapseconds</i>	Reads the leap second information from the <i>Leapseconds</i> file. If this option is not used, leap second information does not appear in the output.
-p <i>Posixrules</i>	Use the <i>posixrules</i> rules when handling POSIX-format time zone environment variables. The zic command acts as if the file contains a link as in this example: <pre>Link timezone posixrules</pre>

Table 57. Flags (continued)

Item	Description
<code>> -r [@lo]/[@hi] <</code>	<code>> </code> The -r flag limits the applicability of the output binary files to timestamps in the range that is specified by the <i>lo</i> and <i>hi</i> variables. The values of the <i>lo</i> and <i>hi</i> variables are signed decimal counts of seconds since the EPOCH (1970-01-01 00:00:00 Coordinated Universal Time). The range includes the value of the <i>lo</i> variable but excludes the value of the <i>hi</i> variable. If the <i>lo</i> and the <i>hi</i> variables are not specified, the range defaults to extreme values. The output binary files use UT offset 0 and the abbreviation -00 if the <i>lo</i> and the <i>hi</i> variables are not specified. For example, the <code>zic -r @0</code> command returns binary data for nonnegative timestamps (the time after EPOCH). The <code>zic -r @0/@2147483648</code> command returns binary data for nonnegative timestamps that fit into 31-bit signed integers. On platforms with GNU date, the <code>zic -r @\$ (date +%s)</code> command excludes binary data that is intended for timestamps before the specified date. Using the -r flag reduces the size of the output binary file. However, the size of the output binary file might increase due to the need to represent the timestamp range boundaries. If the value of the <i>hi</i> variable causes a time zone information format (TZif) file to contain explicit entries for pre-<i>hi</i> transitions instead of representing them with an extended POSIX TZ format string, the size of the output binary file might increase. Note: You can also reduce the size of the binary file by using the -b slim option. <code> <</code>
<code>> -R @hi <</code>	<code>> </code> The -R flag generates redundant trailing explicit transitions for timestamps that occur less than the specified value of the <i>hi</i> variable in seconds since the EPOCH. These transitions can be more concisely represented via the extended POSIX TZ string. The -R flag does not affect the represented timestamps. You can accommodate nonstandard TZif readers that ignore the extended POSIX TZ string by using the -R flag. However, this option increases the size of the output binary file.<code> <</code>
-v	Provides a message if a year that appears in a data file is outside the range of years representable by system time values (0:00:00 AM GMT, 1 January 1970, to 3:14:07 AM GMT, 19 January 2038).
<code>> --version <</code>	<code>> </code> Displays the version information of the file and exits.<code> <</code>

Parameters

Table 58. Parameters

Item	Description
<i>FileName</i>	A file that contains input lines that specify to create files that contain the time conversion information. If <i>FileName</i> is - (hyphen), then standard input is read.

Examples

1. A rule line can have the following format:

```
Rule  USA  1970    max    -    Sep  Sun<=14 3:00    0    S
```

2. A zone line can have the following format:

```
Zone    Turkey          3:00    Turkey          EET%s
```

3. A link line can have the following format:

```
Link    MET    CET
```

4. To compile a `timezone.infile` file that contains input time zone information and place the binary values into the standard time zone `/usr/share/lib/zoneinfo/` directory, type:

```
zic timezone.infile
```

5. To compile a `timezone.infile` file that contains input time zone information and place the binary values into a directory that is specified with **-d** option, type:

```
zic -d tzdir timezone.infile
```

6. To report warnings during compilation of the time zone input file when the range of years are incorrect, type:

```
zic -v timezone.infile
```

7. To compile a `format.zi` file that contains input time zone information and to place the binary values into a directory that is specified with **-d** flag, enter the following command:

```
zic -d /usr/share/lib/zoneinfo /usr/share/lib/tz/format.zi
```

The `format.zi` file contains information about the input time zone in the DATAFORM format. The `format` has the following valid options:

- `main.zi`: This option is the default DATAFORM format for the `.zi` file.
- `vanguard.zi`: This option is the latest DATAFORM format that supports negative offset values.
- `rearguard.zi`: This DATAFORM format does not support negative offset values.

For more information about DATAFORM format, see the [DATAFORM format](#) section.

DATAFORM format

The new DATAFORM format macro in the makefile provides three source data formats to minimise the downstream disruption when data formats are improved.

The DATAFORM format macro offers the following source data formats:

- **main**: This DATAFORM format installs the data format from files similar to the `africa` data format file or other similar data format files. This option is the default format. Elements of the **main** format must move to the **rearguard** format in the future.

- **vanguard:** This DATAFORM format installs the data format from the bleeding-edge format, which is released but is not reliably tested. Elements of the **vanguard** format must move to the **main** format in the future.
- **rearguard:** This DATAFORM format installs the data format from the trailing-edge format that had more time to mature through iterative improvement.

Currently, the **main** and **rearguard** formats are identical and match the 2018 calendar. The **vanguard** format has a special feature that supports negative offset values. This feature of the **vanguard** format improves support for Ireland, where they follow Irish Standard Time (IST, UTC+01) in summer and Greenwich Mean Time (UTC) in winter. The tzcode library has supported negative offset values for a long time and must move to the **main** format soon. However, the tzcode library might not move to the **rearguard** format soon because some downstream parsers do not support negative offset values.

The build procedure for the DATAFORM format macro creates three files that are named `vanguard.zi`, `main.zi`, and `rearguard.zi`. Although the `vanguard.zi`, `main.zi`, and `rearguard.zi` files represent the same data, they might have minor discrepancies. These files are intended for the downstream data users. The zoneinfo parsers that do not support negative offset values must start using the `rearguard.zi` data format file so that they are not affected when the negative offset feature is moved from the **vanguard** data format to the **main** data format. The bleeding-edge zoneinfo parsers that support the new negative offset values can use the `vanguard.zi` data format file.

Exit Status

- 0**
 - The command was completed successfully.
- >0**
 - An error occurred.

Files

Table 59. Files

Item	Description
<code>/usr/sbin/zic</code>	Contains the SystemV zic command.
<code>/usr/share/lib/zoneinfo</code>	Standard directory used for files that are created by the zic command.
<code>/usr/share/lib/tz/</code>	Contains <code>.zi</code> files to support DATAFORM formats by the zic command.

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as the customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at [Copyright and trademark information at www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

The registered trademark Linux is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Red Hat®, JBoss®, OpenShift®, Fedora®, Hibernate®, Ansible®, CloudForms®, RHCA®, RHCE®, RHCSA®, Ceph®, and Gluster® are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Index

Special Characters

- .hash pseudo-op
 - sendmail [376](#)
- /etc/filesystems file
 - removing entries using rmfs command [173](#)
- /etc/inittab file
 - removing records
 - using rmitab command [177](#)
- /etc/utmp
 - monitor [1102](#)
- /etc/uucp/Permissions
 - checking [1103](#)
- /etc/vfs file
 - removing entries
 - using rmvfs command [231](#)

Numerics

- 128-port asynchronous controller
 - querying characteristics [594](#)
 - setting characteristics [594](#)

A

- accounting system
 - correcting format problems [1303](#)
 - deleting files
 - using remove command [72](#)
 - running
 - using runacct command [280](#)
 - starting up
 - using startup command [547](#)
 - summarizing records
 - using sa command [303](#)
 - turning off
 - using shutacct command [414](#)
 - using turnacct command [1028](#)
 - turning on
 - using turnacct command [1028](#)
- acct/* commands
 - remove [72](#)
 - shutacct [414](#)
 - startup [547](#)
 - turnacct [1028](#)
- administration program
 - for SCCS commands
 - using sccs command [331](#)
- aliases
 - locating [1266](#)
 - removing [1058](#)
- aliases file
 - sendmail [376](#)
- alters priorities of running using renice command [75](#)
- analyzing virtual memory snapshot
 - memory management
 - using svmon command [623](#)

- archive
 - using tar command [687](#)
- arithmetic
 - converting units [1071](#)
- ARPANET
 - user of whois command [1274](#)
- ATE program
 - transferring files with xmodem [1369](#)
 - xmodem command [1369](#)
- auditing
 - file installation in a secure system
 - using sysck command [700](#)
- authorization
 - trace
 - using traceauth command [901](#)
- authorization data
 - changing authorization data
 - using tpm_changeauth command [854](#)
- authorization information
 - editing and displaying
 - using xauth command [1326](#)

B

- backing up files
 - onto a remote machines' device using rdump command [43](#)
- bibliographic database
 - printing
 - using roffbib command [237](#)
 - sorting
 - using sortbib command [503](#)
- binary data
 - storing in a file
 - using sa1 command [305](#)
- binary file
 - decoding for mail transmission
 - using uuencode command [1116](#), [1122](#)
 - encoding for mail transmission
 - using uuencode command [1116](#), [1122](#)
 - finding the printable strings
 - using strings command [580](#)
- binary recording
 - CEC metrics [841](#)
 - local system metrics [841](#)
 - topasrec [841](#)
- bindings
 - configures virtual key
 - using xmbind command [1366](#)
- block count
 - displaying a file's
 - using sum command [610](#)
- BNU
 - checking status of operations
 - using uustat command [1136](#)
 - commands
 - executing remotely [1146](#)

BNU (continued)

commands (continued)

- running remotely [1142](#)

- communication between TCP/IP [1116](#)

- configuration information, entering

- using uucpadm command [1113](#)

- copying files between operating systems [1110](#)

- debugging mode

- using [1124](#)

- debugging remote connections [1104](#)

- files

- completing transfer of [1128](#)

- converting to ASCII [1122](#)

- copying between systems [1139](#)

- scheduling transfers [1133](#)

- transferring between systems [1104](#)

- initiating transport calls

- using uudemmon.hour command [1120](#)

- log files

- cleaning [1125](#)

- displaying [1125](#)

- networked computers

- listing [1127](#)

- polling remote systems

- using uudemmon.poll command [1121](#)

- required files

- checking for [1103](#)

- spooling directories

- cleaning [1106](#), [1108](#)

- deleting files from [1108](#)

- removing files from [1106](#)

- status

- obtaining [1135](#)

- tip command

- escape signals [766](#)

- variables [766](#)

- uucp command [1103](#)

- uucico daemon [1104](#)

- uuclean command [1106](#)

- uucleanup command [1108](#)

- uucp [1110](#)

- uucpadm command [1113](#)

- uucpd daemon [1116](#)

- uudemmon.admin command [1117](#)

- uudemmon.cleau command [1118](#)

- uudemmon.hour command [1120](#)

- uudemmon.poll command [1121](#)

- uukick command [1124](#)

- uulog command [1125](#)

- uupoll command [1130](#)

- uuq command [1131](#)

- uusched daemon [1133](#)

- uusend command [1134](#)

- uusnap command [1135](#)

- uuxqt daemon [1146](#)

BNU job queue

- deleting entries

- using uuq command [1131](#)

Bourne shell [268](#)

C

C programming language

- compiling patterns into declarations [69](#)

CEC metrics

- binary recording

- topasrec [841](#)

- changing TPM active states

- using tpm_activate command [854](#)

- changing TPM enable states

- using tpm_enable command [858](#)

- changing TPM ownership operation settings

- using tpm_ownable command [860](#)

- changing TPM physical presence settings

- using tpm_present command [860](#)

- character translation [909](#)

- characters

- reversing in each line of a file

- using rev command [118](#)

- translating

- using tr command [886](#)

- checking

- file installation in a secure system

- using tcbck command [700](#)

- checksum

- displaying a file's

- using sum command [610](#)

- code set maps

- setting [391](#)

- code sets [391](#)

- command

- printing the time of execution

- using time command [759](#)

- wparprnterr [1298](#)

- X [1307](#)

- Command

- update_iscsi [1079](#)

- commands

- event response resource manager (ERRM)

- wallevent [1254](#)

- ras_logger [13](#)

- rc [20](#)

- rc.powerfail [22](#)

- rcp [24](#)

- recfgct [50](#)

- refsrc [64](#)

- refsensor [66](#)

- removevsd [73](#)

- remsh [269](#)

- rendev [74](#)

- resetsrc [85](#)

- resumevsd [117](#)

- rev [118](#)

- rmaudrec [135](#)

- rmcctrl [146](#)

- rmcomg [156](#)

- rmcondition [158](#)

- rmcondresp [161](#)

- rmddel [165](#)

- rmdev [166](#)

- rmdir [168](#)

- rmlpcmd [178](#)

- rmm [184](#)

- rmnamsv [185](#)

- rmnfsexp [186](#)

- rmramdisk [197](#)

- rmresponse [198](#)

- rmrpdomain [202](#)

commands (*continued*)

- [rmrpnod](#) [204](#)
- [rmsensor](#) [213](#)
- [rmsmbcmnt](#) [216](#)
- [rmsmbcred](#) [217](#)
- [rmss](#) [219](#)
- [rmvirprt](#) [231](#)
- [rsh](#) [269](#)
- [runact](#) [283](#)
- [runlpcmd](#) [288](#)
- [ruser](#) [293](#)
- [rvsdrestrict](#) [296](#)
- [sccs](#) [331](#)
- [sccshelp](#) [335](#)
- [slattach](#) [432](#)
- [sliplogin](#) [434](#)
- [smdemon.cleanu](#) [446](#)
- [snapshot](#) [465](#)
- [snmpevent](#) [477](#)
- [starttrpdomain](#) [536](#)
- [starttrpnod](#) [539](#)
- [starttrsrc](#) [541](#)
- [stopcondresp](#) [557](#)
- [stoptrpdomain](#) [560](#)
- [stoptrpnod](#) [562](#)
- [stoptrsrc](#) [564](#)
- [stty](#) [598](#)
- [su](#) [607](#)
- [sum](#) [610](#)
- [svmon](#) [623](#)
- [tbl](#) [695](#)
- [tcbck](#) [700](#)
- [timedc](#) [763](#)
- [tip](#) [766](#)
- [topas](#) [841](#)
- [topasrec](#) [841](#)
- [touch](#) [850](#)
- [tr](#) [886](#)
- [tracesoff](#) [906](#)
- [trcevgrp](#) [914](#)
- [troff](#) [929](#)
- [trustchk](#) [1004](#)
- [tset](#) [1011](#)
- [tsh](#) [1014](#)
- [tsm](#) [1016](#)
- [tunchange](#) [1020](#)
- [tuncheck](#) [1022](#)
- [tvi](#) [1030](#)
- [type](#) [1035](#)
- [unfencevsd](#) [1064](#)
- [updatevsdnod](#) [1080](#)
- [updatevsdtab](#) [1082](#)
- [updatevsdvg](#) [1083](#)
- [usrck](#) [1093](#)
- [uucpadm](#) [1113](#)
- [uudemon.admin](#) [1117](#)
- [uudemon.cleanu](#) [1118](#)
- [uudemon.hour](#) [1120](#)
- [uudemon.poll](#) [1121](#)
- [uuid_get](#) [1123](#)
- [uuq](#) [1131](#)
- [uustat](#) [1136](#)
- [vacation](#) [1149](#)
- [vpdadd](#) [1234](#)

commands (*continued*)

- [vsdchgserver](#) [1239](#)
- [vsdelnod](#) [1240](#)
- [vsdnod](#) [1242](#)
- [vsdsklst](#) [1244](#)
- [w](#) [1251](#)
- [wall](#) [1253](#)
- [wallevnt](#) [1254](#)
- [who](#) [1268](#)
- [wlmassign](#) [1277](#)
- [wlmcntrl](#) [1281](#)
- [wlmstat](#) [1284](#)
- [xdm](#) [1334](#)
- [xmbind](#) [1366](#)
- [xmodmap](#) [1371](#)
- [xterm](#) [1399](#)
- [ypcat](#) [1437](#)
- [ypinit](#) [1438](#)
- [yppasswd](#) [1441](#)
- [yppoll](#) [1443](#)
- [yppush](#) [1444](#)
- [ypset](#) [1446](#)
- [ypwhich](#) [1448](#)
- [ypxfr](#) [1450](#)
- communication channel
 - receiving mail in a secure
 - using xget command [1352](#)
 - sending mail in a secure
 - using xsend command [1390](#)
- conditional expressions
 - evaluating [746](#)
- configuring
 - virtual key bindings
 - using xmbind command [1366](#)
- control scripts
 - topsvcsctrl [847](#)
- control, limited
 - of BNU operations
 - using uustat command [1136](#)
- conversing with other users
 - using talk command [683](#)
- core dump size limits [1049](#)
- core file
 - gathering core file [464](#)
- creating a message source file [287](#)

D

- daemon
 - utmpd [1102](#)
- daemons
 - pcnfsd [251](#)
 - rshd [273](#)
 - tftpd [756](#)
 - ypbind [1436](#)
 - yppasswdd [1442](#)
 - ypserv [1445](#)
 - ypupdated [1447](#)
- daily report
 - writing in a file
 - using sa2 command [306](#)
- data area size limits [1049](#)
- databases, system
 - controlling foreign host access, manipulating

- databases, system (*continued*)
 - controlling foreign host access, manipulating (*continued*)
 - using ruser command [293](#)
- deleting entries
 - BNU job queue
 - using uuq command [1131](#)
- delta files
 - removing
 - using rmdel command [165](#)
- description of command type and arguments
 - using type command [1035](#)
- device configuration commands
 - restbase [95](#)
 - restbase command [95](#)
 - savebase [318](#)
- devices
 - customized
 - saving information about [318](#)
- Devices file format
 - setting up
 - using uucpadm command [1113](#)
- Dialcodes file format
 - setting up
 - using uucpadm command [1113](#)
- directories
 - deleting [131](#)
 - removing [131](#)
- directory
 - unmounting
 - using umount command [1054](#)
- disabling TPM clear operations
 - using tpm_clearable command [857](#)
- displaying TPM's endorsement key public part
 - using tpm_getpubek command [859](#)
- dump device
 - changing the primary [663](#)
 - changing the secondary [663](#)
 - starting a kernel dump to the primary [669](#)
 - starting a kernel dump to the secondary [669](#)
- dump file
 - formatting for printer output
 - using xpr command [1379](#)

E

- edit status
 - displaying [306](#)
- endorsement key pair on TPM
 - using tpm_createek command [858](#)
- ERRM commands
 - elogevent [1254](#)
 - snmpevent [477](#)
- ERRM scripts
 - snmpevent [477](#)
 - wallevent [1254](#)
- errors
 - fixing in file
 - using tcbck command [700](#)
- escape signals
 - using tip command [766](#)
- event response resource manager (ERRM)
 - commands
 - wallevent [1254](#)
 - scripts

- event response resource manager (ERRM) (*continued*)
 - scripts (*continued*)
 - wallevent [1254](#)
- exit values
 - returning [999](#)
- expressions
 - evaluating conditional [746](#)

F

- fastboot command [48](#)
- file
 - copying created by the backup command
 - using restore command [96](#)
 - deleting repeated lines in a
 - using uniq command [1069](#)
 - fixing errors in
 - using tcbck command [700](#)
 - reversing characters in each line of a
 - using rev command [118](#)
 - splitting into pieces
 - using split command [517](#)
 - unmounting
 - using umount command [1054](#)
- file inclusion
 - processing
 - using soelim command [495](#)
- file mode creation masks [1051](#)
- file size limits [1049](#)
- file system
 - copying backup from remote machine to local machine
 - using rrestore command [265](#)
 - unmounting
 - using umount command [1054](#)
- file systems
 - removing unwanted files
 - using skulker command [431](#)
- files
 - comparing two
 - using sdiff command [354](#)
 - compression [1061](#), [1454](#)
 - copying between systems [1110](#)
 - decompression [1061](#)
 - deleting [131](#)
 - displaying block count
 - using sum command [610](#)
 - displaying comparison side-by-side of two
 - using sdiff command [354](#)
 - displaying the checksum
 - using sum command [610](#)
 - executable
 - locating [1266](#)
 - expanding
 - using unpack command [1077](#)
 - expansion [1453](#)
 - locating sections [1265](#)
 - merging
 - using sort command [497](#)
 - removing [131](#)
 - removing ifdef'ed lines [1067](#)
 - SCCS
 - canceling specified versions [1066](#)
 - comparing two versions [334](#)
 - displaying edit status [306](#)

- files (*continued*)
 - SCCS (*continued*)
 - displaying identifying information [1259](#)
 - validating [1151](#)
 - sorting
 - using sort command [497](#)
 - sorting unordered lists [1017](#)
 - transferring with tftp command [751](#)
 - writing
 - from specified point [681](#)
- finding literature references in
 - using refer command [58](#)
- firmware-assisted system dump modifying
 - sysdumpdev [663](#)
- folder
 - displaying messages in a
 - using scan command [328](#)
- folders
 - deleting
 - using rmf command [170](#)
 - removing messages within
 - using rmf command [170](#)
- font directories
 - adding [1321](#)
- font path element
 - removing [1321](#)
- font servers
 - adding [1320](#)
- fonts
 - supplying to X Window display servers [1350](#)
- FORTTRAN
 - translating programs to RATFOR [593](#)

G

- games
 - hunt the wumpus [1304](#)
 - tic-tac-toe [1018](#)
- games directory permissions [1029](#)
- groups
 - resetting for the current login session
 - using setgroups command [387](#)

H

- help
 - describing command functions [1260](#)
- hlptcpdump [706](#)
- hlpuil [1046](#)
- host
 - showing the status on a
 - using ruptime command [292](#)
- host machine
 - controlling access
 - using xhost command [1354](#)
- hosts
 - connecting local with remote
 - using rlogin command [127](#)
 - using telnet command [730](#)
 - using tn command [730](#)
 - using tn3270 command [730](#)

I

- i-node table
 - updating
 - using sync command [649](#)
- iconv library
 - generating conversion table for [1039](#)
- ID, user
 - associated with session
 - using su command [607](#)
- image
 - displaying
 - using xwud command [1430](#)
 - dumping
 - using xwd command [1428](#)
 - retrieving
 - using xwud command [1430](#)
- incoming messages
 - notifying user [30](#)
- inetd daemon
 - uucpd daemon and [1116](#)
- init command [726](#)
- initialization, startup
 - using rc command [20](#)
- initiating transport calls
 - using BNU program
 - using uudemon.hour command [1120](#)
- inserting literature references in
 - using refer command [58](#)
- installing
 - files in a secure system
 - verifying using sysck command [700](#)
- Internet
 - tracing network packets [903](#)
- ip security crypto module [1074](#)

K

- kernel messages
 - writing to terminal [1085](#)
- kernel name list
 - generating a [915](#)
- key bindings
 - configures virtual
 - using xmbind command [1366](#)
- keywords
 - SCCS
 - substituting values [1157](#)
- Korn shell
 - enhanced [124](#)
 - invoking [124](#)
 - restricted [124](#)

L

- libraries
 - converting archives [5](#)
- lines
 - deleting repeating
 - using uniq command [1069](#)
- links
 - removing [131](#)
- local system metrics

- local system metrics (*continued*)
 - binary recording
 - topasrec [841](#)
- locking X display until password is entered
 - using xlock command [1362](#)
- log files (BNU)
 - cleaning up [1125](#)
- log, trace
 - formatting a report from
 - using trcrpt command [916](#)
- logged in, users
 - identifying
 - using who command [1268](#)
- logical volume
 - copying one volume to a new volume [55](#), [649](#)
 - removing from a volume group
 - using rmlv command [181](#)
 - split and copy [518](#)
 - synchronizing mirrors that are not current
 - using syncvg command [652](#)

M

- mail
 - receiving in a securing communication channel
 - using xget command [1352](#)
 - sending in a secure communication channel
 - using xsend command [1390](#)
- mail bug report
 - mailing of [375](#)
- Mail commands
 - rmail [133](#)
 - sendbug [375](#)
 - sendmail [376](#)
 - smdemon.cleanu [446](#)
- mail, incorporation into folder, MH [29](#)
- man pages
 - rmccli [141](#)
- management information base variables
 - managing with snmpinfo command [480](#)
- managing trusted computing resources
 - using tcscd command [721](#)
- managing Trusted Signature Database (TSD)
 - trustchk [1004](#)
- memory management
 - analyzing virtual memory snapshot
 - using svmon command [623](#)
 - reporting virtual memory statistics [1220](#)
 - updating the super block [1078](#)
- message facility commands
 - runcat [287](#)
- message routing [526](#)
- message source file
 - creating [287](#)
- messages
 - filing in other folders
 - using refile command [60](#)
 - listing lines of [328](#)
 - listing the addresses of recipients of
 - using whom command [1275](#)
 - logs system [672](#)
 - prompting for the disposition of
 - using whatnow command [1261](#)
 - receiving from a remote system

- messages (*continued*)
 - receiving from a remote system (*continued*)
 - using writesrv command [1303](#)
 - removing from active status
 - using rmm command [184](#)
 - replying to
 - using repl command [78](#)
 - saving in packed files [28](#)
 - sending
 - using send command [372](#)
 - sending from a remote system
 - using writesrv command [1303](#)
 - sending to other users
 - using write command [1298](#)
 - showing
 - using show command [411](#)
 - sorting
 - using sortm command [504](#)
 - verifying the addresses of recipients of
 - using whom command [1275](#)
- messages, SCCS
 - displaying help information
 - using sccshelp command [335](#)

MH

- rcvpack command [28](#)
- rcvstore command [29](#)
- rcvttty [30](#)
- slocal command [438](#)
- spost command [526](#)
- MH commands
 - invoking a visual interface for use with
 - using vmh command [1202](#)
- MultiPath I/O
 - rmpath command [190](#)

N

- NDBM database
 - sendmail [376](#)
- network information service [1436](#)
- networked computers
 - displaying list of [1127](#)
- NFS commands
 - rmnfs [186](#)
 - rmnfsexp [186](#)
 - rmnfsmnt [188](#)
 - rpcgen [252](#)
 - rpcinfo [253](#)
 - rup [291](#)
 - rusers [295](#)
 - rwall [297](#)
 - showmount [413](#)
 - spray [528](#)
- NFS daemons
 - pcnfsd [251](#)
 - rexid [120](#)
 - rstatd [275](#)
 - rusersd [296](#)
 - rwalld [298](#)
 - sprayd [529](#)
 - statd [553](#)
- NIS commands
 - rmkeyerv [177](#)
 - rmyp [233](#)

NIS commands (*continued*)

- [ypcat 1437](#)
- [ypinit 1438](#)
- [ypmatch 1440](#)
- [yppasswd 1441](#)
- [yppoll 1443](#)
- [yppush 1444](#)
- [ypset 1446](#)
- [ypwhich 1448](#)
- [ypxfr 1450](#)

NIS daemons

- [ypbind 1436](#)
- [yppasswdd 1442](#)
- [ypserv 1445](#)
- [ypupdated 1447](#)

notify object class (SRC)

- removing a notify method definition [189](#)

nroff command

- formatting table for
using [tbl command 695](#)

O

object file

- finding the printable strings
using [strings command 580](#)

object files

- displaying section sizes of XCOFF [429](#)
- displaying symbol information with [stripnm command 584](#)
- reducing size of XCOFF [582](#)

P

paging

- specifying additional devices for
using [swapon command 644](#)

paging space

- removing [193](#)

parameter

- [Name 74](#)
- [NewName 74](#)

parser

- creating with the [yacc command 1433](#)

path names

- executable files
finding [1266](#)

pcnfsd daemon [251](#)

performing TPM self-test

- using [tpm_selftest command 862](#)

Permissions file format

- setting up
using [uucpadmin command 1113](#)
- verifying [1103](#)

phones file format

- setting the phones variable [771](#)

physical memory size limits [1049](#)

physical volume [82](#)

plotter queue

- removing from the system
using [rmqudev command 196](#)

Poll file format

- setting up

Poll file format (*continued*)

- setting up (*continued*)
using [uucpadmin command 1113](#)

polling

- remote systems
using [uudemon.poll command 1121](#)

printer

- changing driver settings
using [splp command 523](#)
- displaying driver settings
using [splp command 523](#)

printer queue

- removing from the system
using [rmque command 195](#)
using [rmqudev command 196](#)

printer, virtual

- removing
using [rmvirprt command 231](#)

process

- initializing
using [init command 726](#)
using [telinit command 726](#)

process resource allocation

- removing unused modules [434](#)

process suspension

- suspending execution [1252](#)
- suspending execution for an interval [433](#)

processing incoming mail, MH [438](#)

processor

- reporting usage [864](#)

program

- copying output into a file [725](#)
- monitoring
using [watch command 1256](#)

program listing

- formatting
using [vgrind command 1164](#)

program loops

- returning exit values [999](#)

program, administration

- for SCCS commands
using [scs command 331](#)

programs

- creating a Makefile from an Imakefile [1367](#)

Q

querying characteristics

- terminals
using [stty command 598](#)

R

raddbm command [1](#)

radiusctl command [4](#)

RAM disk [197](#)

RAM disk, virtual

- removing
using [rmramdisk command 197](#)

rand command [234](#)

ras_logger command [13](#)

raso command [6](#)

rbacqry Command [14](#)

- rbactoldif command [18](#)
- rc command [20](#)
- rc.mobip6 [20](#)
- rc.powerfail command [22](#)
- rc.wpars [24](#)
- rcp command [24](#)
- rcvdist command
 - incoming messages
 - sending copy to additional recipients [28](#)
- MH
 - rcvdist command [28](#)
- rcvpack command [28](#)
- rcvstore command [29](#)
- rcvttty command [30](#)
- rdistd command [42](#)
- rdump command [43](#)
- readlvcopy Command [47](#)
- realtime messages
 - sending to other users
 - using write command [1298](#)
- reboot command [48](#)
- rebootwpar command [49](#)
- recfgct command [50](#)
- recreatevg command [52](#)
- recsh command [54](#)
- redefinevg command [55](#)
- reducevg command [56](#)
- refer command [58](#)
- refile command [60](#)
- refresh command [62](#)
- refsrc command [64](#)
- refsensor command [66](#)
- Reliable Scalable Cluster Technology (RSCT) topology
 - services
 - control scripts
 - topsvcsctrl [847](#)
 - scripts
 - topsvcs [845](#)
- rembak Command [70](#)
- remote command requests
 - executing [1146](#)
- remote file format
 - setting the remote variable
 - using tip command [771](#)
- remote machines' device
 - backing up files onto using rdump command [43](#)
- remote systems
 - executing commands on [1146](#)
 - polling
 - using uudemmon.poll command [1121](#)
- remove command [72](#)
- removevsd command [73](#)
- removing messages
 - from active status
 - using rmm command [184](#)
- remsh Command [269](#)
- rendev command [74](#)
- renice command [75](#)
- reorgvg command [76](#)
- repetitive responses
 - generating [1435](#)
- repl command [78](#)
- replacepv command [82](#)
- Reporting and management tool for TNC, SUMA

- Reporting and management tool for TNC, SUMA (*continued*)
 - using tnconconsole command [773](#)
- repquota command [83](#)
- reset command [84](#)
- resetsrc command [85](#)
- resize Command [89](#)
- resource database
 - performing utilities for X server
 - using xrdb command [1387](#)
- resource limits [1049](#)
- resource_data_input information file [90](#)
- restart-secdapclntd [94](#)
- restore Command [96](#)
- restorevgfiles Command [107](#)
- restricted shell [268](#)
- restricting endorsement key public part display
 - using tpm_restrictpubek command [861](#)
- restvg command [108](#)
- restwparfiles command [115](#)
- resumevsd command [117](#)
- rev command [118](#)
- revnetgroup Command [119](#)
- rex daemon [120](#)
- rexc command [120](#)
- rexecd daemon [122](#)
- rgb command [122](#)
- ripquery Command [123](#)
- rksh command [124](#)
- rlogin command [127](#)
- rlogind daemon [129](#)
- rmail command [133](#)
- rmaudrec command [135](#)
- rmauth Command [133](#)
- rmC2admin command [139](#)
- rmCCadmin command [140](#)
- rmcli man page [141](#)
- rmcctrl command [146](#)
- rmcdomainstatus command [149](#)
- rmcifscred command [152](#)
- rmcifsmnt command [153](#)
- rmclass command [154](#)
- rmcluster Command [155](#)
- rmcomg command [156](#)
- rmcondition command [158](#)
- rmcondresp command [161](#)
- rmcosi command [164](#)
- rmdd command [165](#)
- rmdev command [166](#)
- rmdir command [168](#)
- rmdom command [170](#)
- rmf command [170](#)
- rmfilt Command [172](#)
- rmfs command [173](#)
- rmgroup command [174](#)
- rmiscsi command [176](#)
- rmitab command [177](#)
- rmkeyserv command [177](#)
- rmipcnd command [178](#)
- rmlv command [181](#)
- rmm command [184](#)
- rmnamsv command [185](#)
- rmnfs command [186](#)
- rmnfsexp command [186](#)
- rmnfsmnt command [188](#)

- rmnfsproxy command [188](#)
- rmpath command [190](#)
- rmprtsv command [192](#)
- rmpps command [193](#)
- rmqos Command [194](#)
- rmqueue command [195](#)
- rmquedev command [196](#)
- rmramdisk command [197](#)
- rmresponse command [198](#)
- rmrole command [200](#)
- rmrpdomain command [202](#)
- rmrpnod command [204](#)
- rmrset [207](#)
- rmrsrc Command [207](#)
- rmsecattr command [211](#)
- rmsensor command [213](#)
- rmsmbcmnt command [216](#)
- rmsmbcred command [217](#)
- rmsock Command [218](#)
- rmss command [219](#)
- rmtcpip Command [225](#)
- rmts command [226](#)
- rmtun Command [227](#)
- rmuser command [229](#)
- rmusil command [228](#)
- rmvfs command [231](#)
- rmvirprt command [231](#)
- rmwpar command [232](#)
- rmyp command [233](#)
- rndc-confgen command [235](#)
- roffbib command [237](#)
- rolelist command [238](#)
- roleqry [239](#)
- rolerpt command [241](#)
- rollback command [242](#)
- route command [243](#)
- route mail for local or network delivery [376](#)
- rpcgen command [252](#)
- rpcinfo command [253](#)
- rpvutil command [261](#)
- rrestore Command [265](#)
- RSCT topology services
 - control scripts
 - topsvcsctrl [847](#)
 - scripts
 - topsvcs [845](#)
- rsh command [269](#)
- Rsh command [268](#)
- rshd daemon [273](#)
- rstatd daemon [275](#)
- rsyslogd Daemon [276](#)
- rt_enable command [279](#)
- rtcd Daemon [278](#)
- runacct command [280](#)
- runact command [283](#)
- runcat command [287](#)
- runlpcmd command [288](#)
- rup command [291](#)
- ruptime command [292](#)
- ruser command [293](#)
- rusers command [295](#)
- rusersd daemon [296](#)
- rvsrestrict command [296](#)
- rwall command [297](#)

- rwalld daemon [298](#)
- rwho command [299](#)

S

- sa command [303](#)
- sa1 command [305](#)
- sa2 command [306](#)
- sadc command [307](#)
- sar command [308](#)
- savebase command [318](#)
- savecore command [320](#)
- savevg command [321](#)
- savewpar command [324](#)
- scan command [328](#)
- SCCS
 - commands
 - administrating [331](#)
 - files
 - administrating [331](#)
 - canceling specified versions [1066](#)
 - comparing two versions [334](#)
 - displaying edit status [306](#)
 - displaying identifying information [1259](#)
 - validating [1151](#)
 - help information [335](#)
 - keywords
 - substituting values [1157](#)
 - removing delta files
 - using rmdel command [165](#)
- scs command [331](#)
- SCCS commands
 - administration program for
 - using scs command [331](#)
 - displaying help information
 - using sccshelp command [335](#)
 - rmdel [165](#)
 - sact [306](#)
 - scs [331](#)
 - sccsdiff [334](#)
 - sccshelp [335](#)
 - unget [1066](#)
 - val [1151](#)
 - vc [1157](#)
 - what [1259](#)
- SCCS messages
 - displaying help information
 - using sccshelp command [335](#)
- sccshelp command [335](#)
- schedo command [336](#)
- scls command [344](#)
- screen
 - copying display to a file [725](#)
 - creating a typescript [345](#)
- screen color characterization data
 - loading from root window
 - using xcmsdb command [1333](#)
 - queries
 - using xcmsdb command [1333](#)
 - removing from root window
 - using xcmsdb command [1333](#)
- screen lock
 - controlling
 - using xss Command [1397](#)

- scripts
 - event response resource manager (ERRM)
 - wallevent [1254](#)
 - snmpevent [477](#)
 - topsvcs [845](#)
 - topsvcsctrl [847](#)
 - wallevent [1254](#)
- sctpctrl command [346](#)
- sdiff command [354](#)
- secldapclntd [357](#)
- secldifconv command [359](#)
- sectoldif command [361](#)
- securetcip command [362](#)
- security
 - auditing the state of the system
 - using sysck command [700](#)
 - locking workstation screens
 - using xss Command [1397](#)
- sed command [363](#)
- sedmgr command [368](#)
- sendbug command [375](#)
- sendmail command [376](#)
- server function for rexec command, TCP/IP [122](#)
- setclock command [385](#)
- setea command [386](#)
- setgroups command [387](#)
- setkst command [389](#)
- setmaps command [391](#)
- setrunmode [394](#)
- setsecattr command [395](#)
- setsecconf [401](#)
- setsenv command [403](#)
- setsyslab [404](#)
- settime command [405](#)
- setting characteristics
 - terminals
 - using stty command [598](#)
- setting up an owner on TPM
 - using tpm_takeownership command [863](#)
- setuname command [407](#)
- sh command [408](#)
- shell
 - executing with log in credentials
 - using the shell command [410](#)
- shell command [410](#)
- shell scripts
 - program loops
 - returning exit values [999](#)
 - repetitive responses
 - generating [1435](#)
- shells
 - default [408](#)
 - locating executable files in C [1266](#)
- show command [411](#)
- showmount command [413](#)
- shutacct command [414](#)
- shutdown command [415](#)
- sisraidmgr command [418](#)
- sissasraidmgr command [422](#)
- size command [429](#)
- skctl command [431](#)
- skulker command [431](#)
- slattach command [432](#)
- sliplogin Command [434](#)
- slocal command [438](#)
- slp_srvreg command [439](#)
- smbcd daemon [441](#)
- smbcstat Command [442](#)
- smbctune Command [444](#)
- smdemon.cleanu command
 - Mail [446](#)
- smit command [447](#)
- smit.log file
 - redirecting [447](#), [450](#)
- smit.script file
 - redirecting [447](#), [450](#)
- smitty command [450](#)
- smrsh command [452](#)
- snap command [456](#)
- snapshot command [465](#)
- snapsplit command [469](#)
- SNMP
 - switching versions of snmpd agent daemon [487](#)
- SNMP version 1
 - Agent Applications
 - snmpdv1 command [471](#)
 - start SNMP version 1 agent as background process [471](#)
- SNMP version 3
 - Agent Applications
 - snmpdv3 command [474](#)
 - start SNMP version 3 agent daemon as background process [474](#)
- snmpd daemon [470](#)
- snmpdv1 daemon [471](#)
- snmpdv3 daemon [474](#)
- snmpevent command [477](#)
- snmpevent script [477](#)
- snmpinfo command [480](#)
- snmpmibd daemon [483](#)
- snmptrap command [486](#)
- snmpv3_ssw command [487](#)
- SNOBOL
 - compiling and interpreting [488](#)
- sntp4 [489](#)
- sodebug command [493](#)
- soelim command [495](#)
- soestat command [496](#)
- sort command [497](#)
- sortbib command [503](#)
- sortm command [504](#)
- spaces
 - changing from tabs
 - using untab command [1078](#)
 - changing into tabs
 - using tab command [677](#)
- spell command
 - maintain hash lists for [506](#)
- spellin command [508](#)
- spelling list
 - creating
 - example of [508](#)
 - using spellin command [508](#)
 - verifying the absence of a word on
 - example of [509](#)
 - using spellout command [509](#)
- spellout command [509](#)
- splat [509](#)
- split command [517](#)

- splitlvcopy command [518](#)
- splitvg command [521](#)
- splp command [523](#)
- spooling directories [1106](#)
- spost command [526](#)
- spray command [528](#)
- sprayd daemon [529](#)
- SRC
 - removing a subserver object definition [216](#)
 - removing a subsystem notification method [189](#)
- SRC configuration commands
 - rmnotify [189](#)
 - rmsrver [216](#)
- srcmstr daemon [529](#)
- standard input
 - copying to a file [725](#)
 - creating typescript [345](#)
- start-secdapclntd [531](#)
- starttrpdomain command [536](#)
- starttrpnode command [539](#)
- startsrc command [541](#)
- startsrc command [545](#)
- startup
 - performing initialization for a normal
 - using rc command [20](#)
- startup command [547](#)
- startwpar
 - activates workload partition [549](#)
- startx Command [551](#)
- statd daemon [553](#)
- status, reporting
 - of BNU operations
 - using uustat command [1136](#)
- stop-secdapclntd [556](#)
- stopcondresp command [557](#)
- stoprpdomain command [560](#)
- stoprpnnode command [562](#)
- stopsrc command [564](#)
- stopsrc command [568](#)
- stopwpar
 - deactivates an active workload partition [571](#)
- storage
 - protection keys
 - using skctl command [431](#)
- stpinet method [573](#)
- strace command [573](#)
- strchg command [575](#)
- strclean command [576](#)
- strconf command [577](#)
- STREAMS
 - displaying information [579](#)
 - tunable parameters [591](#)
- STREAMS command
 - strchg [575](#)
 - strconf [577](#)
 - strload [586](#)
- STREAMS commands
 - scls [344](#)
 - strace [573](#)
 - strclean [576](#)
- STREAMS facility
 - configuration
 - changing [575](#)
 - querying [577](#)
- STREAMS facility (*continued*)
 - driver names
 - listing [344](#)
 - error log
 - receiving messages [578](#)
 - error logger
 - cleaning up [576](#)
 - modules
 - listing [344](#)
 - portable environment
 - loading and configuring [586](#)
 - strerr daemon [578](#)
 - trace messages
 - printing [573](#)
- strerr daemon [578](#)
- strinfo command [579](#)
- strings command [580](#)
- strings, shared
 - using xstr command [1398](#)
- striprnm command [584](#)
- strload command [586](#)
- strreset command [590](#)
- strtune command [591](#)
- sttinet method [594](#)
- stty command [598](#)
- stty-cxma command [594](#)
- style command [606](#)
- su command [607](#)
- subj command [610](#)
- subject list
 - generating
 - using subj command [610](#)
- subroutine call interface program [657](#)
- subroutine calls
 - performing [657](#)
- subserver
 - starting
 - using startsrc command [545](#)
 - stopping
 - using stopsrc command [568](#)
 - turning off tracing
 - using tracesoff command [906](#)
 - turning on tracing
 - using traceson command [907](#)
- subservers
 - removing SRC object definition [216](#)
- subsystem
 - requesting a refresh of
 - using refresh command [62](#)
 - starting
 - using startsrc command [545](#)
 - stopping
 - using stopsrc command [568](#)
 - turning off tracing
 - using tracesoff command [906](#)
 - turning on tracing
 - using traceson command [907](#)
- subsystems
 - removing a notification method [189](#)
- sum command [610](#)
- suma command [611](#)
- super block
 - updating [1078](#)
- svmon command [623](#)

- swap
 - specifying additional devices for [644](#)
- swap command [642](#)
- swapon command [644](#)
- swapping [644](#)
- swcons command [645](#)
- swrole command [647](#)
- swts command [648](#)
- symbol table
 - sendmail [376](#)
- sync command [649](#)
- synclvdm command [649](#)
- syncroot
 - synchronizes non-share portion [651](#)
- syncvg command [652](#)
- syncwpar
 - Synchronizes software between global system and a workload partition [655](#)
- sysck command [658](#)
- syscorepath command [662](#)
- sysdumpdev command [663](#)
- sysdumpstart command [669](#)
- sysline command [670](#)
- syslogd daemon [672](#)
- system
 - displaying uptime for the
 - using uptime command [1086](#)
 - ending operation of the
 - using shutdown command [415](#)
 - restarting
 - using reboot command [48](#)
- system boot
 - boot image
 - boot image [95](#)
 - reading information from [95](#)
- system call interface program [657](#)
- system calls
 - performing [657](#)
- system console
 - redirecting temporarily to a device
 - using swcons command [645](#)
 - redirecting temporarily to a file
 - using swcons command [645](#)
- system dump
 - saving [320](#)
- system management
 - performing
 - using smit command [447](#)
 - using smitty command [450](#)
- system resource controller [189](#)
- system security state
 - auditing
 - trustchk [1004](#)
- system status
 - displaying on terminal status line [670](#)
- Systems file format
 - setting up
 - using uucpadm command [1113](#)

T

- tab command [677](#)
- tables
 - formatting for nroff command

- tables (*continued*)
 - formatting for nroff command (*continued*)
 - using tbl command [695](#)
 - formatting for troff command
 - using tbl command [695](#)
- tabs
 - changing from spaces
 - using tab command [677](#)
 - changing into spaces
 - using untab command [1078](#)
- talk command [683](#)
- talkd daemon [684](#)
- tape device
 - consistency checking
 - tapechk command [686](#)
 - copying
 - tcopy command [705](#)
 - giving subcommands to a streaming
 - using tctl command [722](#)
- tape devices
 - allowing remote access
 - rmt command [224](#)
- tar Command [687](#)
- tbl command [695](#)
- tc command [698](#)
- tcbck command
 - modes of operation [700](#)
- TCP Traffic Regulation (TR) policy
 - tcptr command [719](#)
- TCP/IP
 - configuration database managing entries
 - using ruser command [293](#)
 - daemon
 - rexecd [122](#)
 - hosts
 - listing logged in users [299](#)
 - setting time and date [385](#)
 - inet instance
 - disabling [573](#)
 - enabling [594](#)
 - internet instance
 - undefining [1041](#)
 - unloading [1037](#)
 - methods
 - undefinet [1041](#)
 - print services
 - unconfiguring [192](#)
 - routing tables
 - making manual entries [243](#)
 - security feature
 - enabling [362](#)
 - server function
 - providing [129](#)
 - support for talk command [684](#)
 - support for TELNET protocol [743](#)
 - server function for TFTP
 - using tftpd daemon [756](#)
 - TCP sockets
 - tracing [995](#)
 - time server daemon
 - invoking [761](#)
 - tracing Internet packets [903](#)
 - tracking packets [995](#)
- TCP/IP commands

TCP/IP commands (*continued*)

- executing on a remote host [120](#)
- rmnamsv [185](#)
- rmprtsv [192](#)
- route [243](#)
- ruser [293](#)
- rwho [299](#)
- securetcip [362](#)
- setclock [385](#)
- slattach [432](#)
- sliplogin [434](#)
- tftp [751](#)
- timedc [763](#)
- tracert [903](#)
- trpt [995](#)
- utftp [751](#)

TCP/IP daemons

- rlogind [129](#)
- rshd [273](#)
- talkd [684](#)
- telnetd [743](#)
- tftpd [756](#)
- timed [761](#)

TCP/IP methods

- stpinet [573](#)
- sttinnet [594](#)
- ucgif [1037](#)
- ucfginet [1037](#)
- udefinet [1041](#)

TCP/IP smit commands

- rmnamsv [185](#)
- rmprtsv [192](#)
- ruser [293](#)

tcpdump command

- [706](#)

tcptr command

- [719](#)

tcspd command

- [721](#)

TE, Trusted Execution

- [1004](#)

tee command

- [725](#)

telinit command

- [726](#)

telnet command

- [730](#)

TELNET protocol

- implementing
 - using telnet command [730](#)
 - using tn command [730](#)
 - using tn3270 command [730](#)

telnetd daemon

- telnet options [744](#)

termcap environment variable

- setting to current window size
 - using the resize Command [89](#)

termdef command

- [746](#)

terminal emulator

- providing for X Window System
 - using the xterm Command [1399](#)

terminal maps

- setting [391](#)

terminal sessions

- making a typescript [345](#)

terminal settings

- setting to current window size [89](#)

terminal state manager

- invoking
 - using tsm command [1016](#)

terminals

terminals (*continued*)

- initializing
 - using tset command [1011](#)
- initializing using reset command [84](#)
- manipulating kernel messages [1085](#)
- querying characteristics
 - using stty command [598](#)
 - using termdef command [746](#)
- setting characteristics
 - using stty command [598](#)
 - using tset command [1011](#)
- setting characteristics using reset command [84](#)
- setting tab stops [677](#)
- specifying baud rate
 - using tset command [1011](#)
- writing path names to standard output [1019](#)

terminfo descriptor files

- translating from source to compiled format [759](#)

test command

- [746](#)

tetoldif command

- [749](#)

tftp command

- [751](#)

tftpd daemon

- [756](#)

tic command

- [759](#)

time

- displaying the current
 - using xclock command [1330](#)

time command

- [759](#)

timed daemon

- manipulating with SRC [761](#)

timedc command

- variables [763](#)

timex command

- [765](#)

tip command

- escape signals [766](#)
- phones file format
 - setting the phones variable [771](#)
- remote file format
 - setting the remote variable [771](#)
- variables [766](#)

tn command

- [730](#)

tn3270 command

- [730](#)

tnconconsole command

- [773](#)

tninit command

- [779](#)

token-ring device driver

- displaying statistics [781](#)

tokstat command

- [781](#)

topasrec command

- [841](#)

topology services subsystem

- control scripts
 - topsvcsctrl [847](#)
- scripts
 - topsvcs [845](#)

topsvcs script

- [845](#)

topsvcsctrl script

- [847](#)

touch command

- [850](#)

TPM default state

- using tpm_clear command [856](#)

TPM version

- using tpm_takeownership command [863](#)

tpm_activate command

- [854](#)

tpm_changeauth command

- [854](#)

tpm_clear command

- [856](#)

tpm_clearable command

- [857](#)

tpm_createek command

- [858](#)

- tpm_enable command [858](#)
- tpm_getpubek command [859](#)
- tpm_ownable command [860](#)
- tpm_present command [860](#)
- tpm_restrictpubek command [861](#)
- tpm_selftest command [862](#)
- tpm_takeownership command [863](#)
- tpm_version command [863](#)
- tprof command [864](#)
- tput command [884](#)
- tr command [886](#)
- trace buffer
 - extracting from system dump image [912](#)
- trace log
 - formatting a report from
 - using trcrpt command [916](#)
- trace report
 - adding format templates
 - using trcupdate command [927](#)
 - deleting format templates
 - using trcupdate command [927](#)
 - replacing format templates
 - using trcupdate command [927](#)
- trace session
 - ending
 - using trcstop command [926](#)
- traceauth command [901](#)
- tracepriv command [902](#)
- traceroute command [903](#)
- tracesoff command [906](#)
- traceson command [907](#)
- tracing, turning off
 - subservers or subsystems
 - using tracesoff command [906](#)
- traditional system dump modifying
 - sysdumpdev [663](#)
- translating
 - characters
 - using tr command [886](#)
- trbsd command [909](#)
- trcctl [911](#)
- trcdead command [912](#)
- trcevgrp command [914](#)
- trcnm command [915](#)
- trcrpt command [916](#)
- trcstop command [926](#)
- trcupdate command [927](#)
- troff command
 - command output interpreter for
 - using tc command [698](#)
 - formatting table for
 - using tbl command [695](#)
- trpt command
 - output fields [995](#)
- trustchk command [1004](#)
- Trusted Execution
 - administration
 - trustchk [1004](#)
 - configuring policies
 - trustchk [1004](#)
 - enabling
 - trustchk [1004](#)
- trusted shell
 - interpreting commands in a

- trusted shell (*continued*)
 - interpreting commands in a (*continued*)
 - using tsh command [1014](#)
 - invoking [1015](#)
 - Korn shell
 - differences between [1015](#)
- Trusted Signature Database (TSD)
 - administration
 - trustchk [1004](#)
- TSD, Trusted Signature Database [1004](#)
- tset command [1011](#)
- tsh command [1014](#)
- tsig-keygen command [1015](#)
- tsm command [1016](#)
- tunchange command [1020](#)
- tuncheck command [1022](#)
- tundefault command [1024](#)
- tunrestore command [1025](#)
- tunsave command [1027](#)
- turnacct command [1028](#)
- tvi command [1030](#)
- tvi editor
 - customizing [1031](#)
 - limitations of [1030](#)
 - operating modes of [1030](#)
- twconvdict command [1033](#)
- twconvfont command [1034](#)
- type command [1035](#)

U

- ucgif method [1037](#)
- ucfginet method [1037](#)
- uconvdef command [1039](#)
- udeffif method [1041](#)
- udefinet method [1041](#)
- udfcheck command [1041](#)
- udfcreate command [1042](#)
- udflabel command [1043](#)
- ugrep command [1044](#)
- uil command [1046](#)
- UIL compiler
 - starting
 - using uil command [1046](#)
- uimx command [1047](#)
- ul command [1049](#)
- ulimit command [1049](#)
- umask command [1051](#)
- umcode_latest command [1053](#)
- umountall command [1057](#)
- unalias command [1058](#)
- uncompress command [1061](#)
- underline
 - performing
 - using ul command [1049](#)
- unexpand command [1063](#)
- unfencevsd command [1064](#)
- uniq command [1069](#)
- unlink command [1073](#)
- unlink subroutine [1073](#)
- unloadipsec command [1074](#)
- unmount command [1054](#)
- unpack command [1077](#)
- untab command [1078](#)

- update_iscsi command [1079](#)
- updatevsdnode command [1080](#)
- updatevsdtab command [1082](#)
- updatevsdvg command [1083](#)
- updating
 - files in a secure system
 - verifying using sysck command [700](#)
- uptime command [1086](#)
- user
 - changing session ID
 - using su command [607](#)
 - re-initializing login session
 - using shell command [410](#)
 - removing an account
 - using rmuser command [229](#)
 - resetting protected state environment
 - using setsenv command [403](#)
- user attributes
 - changing
 - usermod command [1091](#)
- user name directory
 - searching for ID
 - using whois command [1274](#)
 - searching for nickname
 - using whois command [1274](#)
- useradd command [1086](#)
- userdel command [1089](#)
- usermod command [1091](#)
- users
 - displaying compact list [1093](#)
 - displaying login name [1271](#)
 - identifying those logged in
 - using who command [1268](#)
- usrck command [1093](#)
- usrrpt command [1101](#)
- utftp command [751](#)
- utmpd [1102](#)
- uucheck command [1103](#)
- uucico command [1104](#)
- uucico daemon [1104](#)
- uuclean command [1106](#)
- uucleanup command [1108](#)
- uucp command [1110](#)
- uucpadm command [1113](#)
- uucpd command [1116](#)
- uucpd daemon [1116](#)
- uudecode command [1116](#)
- uudemon.admin command [1117](#)
- uudemon.admin shell script [1117](#)
- uudemon.cleau command [1118](#)
- uudemon.cleau shell script [1118](#)
- uudemon.hour command [1120](#)
- uudemon.hour shell script [1120](#)
- uudemon.poll command [1121](#)
- uudemon.poll shell script [1121](#)
- uencode command [1122](#)
- uuid_get command [1123](#)
- uukick command [1124](#)
- uulog command [1125](#)
- uname command [1127](#)
- uupick command [1128](#)
- uupoll command [1130](#)
- uuq command [1131](#)
- uusched command [1133](#)

- uusched daemon [1133](#)
- uuseend command [1134](#)
- uusnap command [1135](#)
- uustat command [1136](#)
- uuto command [1128](#), [1139](#)
- uux command [1142](#)
- uuxqt command [1146](#)
- uuxqt daemon [1146](#)

V

- vacation command [1149](#)
- variables
 - tip command
 - setting [766](#)
- varyoffvg command [1152](#)
- varyonvg command [1153](#)
- verifying
 - file installation in a secure system
 - using sysck command [700](#)
- vfcstat command [1160](#)
- vgrind command [1164](#)
- vi command [1166](#)
- vi editor
 - starting [1189](#)
- view command [1189](#)
- Virtual Frame Buffer (VFB)
 - starting X Server with [1307](#)
- virtual key bindings
 - configuring
 - using xmbind command [1366](#)
- virtual memory
 - reporting statistics [1220](#)
- virtual printer
 - removing
 - using rmvirprt command [231](#)
- virtual RAM disk
 - removing
 - using rmramdisk command [197](#)
- vmh command [1202](#)
- volume group
 - activating
 - using varyonvg command [1153](#)
 - deactivating
 - using varyoffvg command [1152](#)
 - recreating a volume group [52](#)
 - removing logical volumes
 - using rmlv command [181](#)
 - removing physical volumes
 - using reducevg command [56](#)
 - reorganizing the physical partition allocation using
 - reorgvg command [76](#)
- vpdadd command [1234](#)
- vsdata1st command [1237](#)
- vsdchgserver command [1239](#)
- vsdelnode command [1240](#)
- vsdnode command [1242](#)
- vsdsk1st command [1244](#)
- vsdvgt command [1248](#)

W

- w command [1251](#)

- wall command [1253](#)
- wallevent command [1254](#)
- wallevent script [1254](#)
- watch command [1256](#)
- wc command [1257](#)
- whatis command [1260](#)
- whatnow command [1261](#)
- which_fileset command [1267](#)
- who command [1268](#)
- whodo command [1272](#)
- whois command
 - ARPANET
 - use on [1274](#)
 - description of [1274](#)
 - example of [1275](#)
- wlmassign command [1277](#)
- wlmcheck command [1279](#)
- wlmcntrl command [1281](#)
- wlmstat command [1284](#)
- wol command [1289](#)
- workload partition
 - synchronizing software using command
 - syncwpar [655](#)
- workstation screens
 - locking
 - using xss Command [1397](#)
- wpar_reg_script [1297](#)
- wparerr command [1290](#)
- wparexec
 - creates an application workload partition or specification file [1291](#)
- wparprnterr command [1298](#)
- write command [1298](#)
- writesrv command [1303](#)
- writing
 - with tabs restored [1063](#)
- writing style
 - analyzing
 - using style command [606](#)
- wtmpfix command [1303](#)

X

- x command [1307](#)
- X display
 - displaying troff files on
 - using xpreview command [1381](#)
 - locking until password is entered
 - using xlock command [1362](#)
- X render extension
 - starting X Server with [1307](#)
- X Server
 - starting [1307](#)
- X session
 - initializing
 - using startx command [551](#)
- x_add_fs_fpe command [1320](#)
- x_add_nfs_fpe command [1321](#)
- x_rm_fpe command [1321](#)
- X-Windows
 - setting environment options [1391](#)
 - setting root window parameters [1395](#)
- xauth command [1326](#)
- xclock command

- xclock command (*continued*)
 - setting the defaults [1332](#)
- xcmsdb Command [1333](#)
- xdm command [1334](#)
- xfs command [1350](#)
- xget command [1352](#)
- xhost command [1354](#)
- xinit command [1355](#)
- xlock command [1362](#)
- xlsfonts command [1365](#)
- xmbind command [1366](#)
- xmodem command [1369](#)
- xmodem protocol [1369](#)
- xmodmap command [1371](#)
- xmpeek command [1373](#)
- xmscheck command [1375](#)
- xmwlm command [1368](#)
- xpr command [1379](#)
- xpreview command [1381](#)
- xprofiler [1384](#)
- xrdb command [1387](#)
- xsend command [1390](#)
- xsetroot command [1395](#)
- xss Command [1397](#)
- xstr command [1398](#)
- xterm command
 - providing terminal emulation
 - for X Windows system [1399](#)
- xwd command [1428](#)
- xwud command [1430](#)

Y

- yacc command [1433](#)
- ypbind daemon [1436](#)
- ypcat command [1437](#)
- ypinit command [1438](#)
- ypmatch command [1440](#)
- yppasswd command [1441](#)
- yppasswdd daemon [1442](#)
- yppoll command [1443](#)
- yppush command [1444](#)
- ypserv daemon [1445](#)
- ypset command [1446](#)
- ypupdated daemon [1447](#)
- ypwhich command [1448](#)
- ypxfr command [1450](#)

Z

- zcat command [1453](#)
- zdump command [1454](#)
- zic command [1457](#)

