

AIX Version 7.3

Commands, l - q

IBM

Note

Before using this information and the product it supports, read the information in [“Notices” on page 1485](#).

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.....	xiii
Highlighting.....	xiii
Case-sensitivity in AIX.....	xiii
ISO 9000.....	xiii
L.....	1
labck Command.....	1
last Command	2
lastcomm Command.....	5
lastlogin Command.....	6
lbxproxy Command.....	7
ld Command	8
ldapgetusrattr command.....	37
ldd Command.....	39
ldedit Command.....	39
learn Command.....	42
leave Command.....	43
lecstat Command.....	44
lex Command.....	45
line Command.....	51
link Command.....	52
lint Command.....	53
listdgrp Command.....	58
listvgbackup Command.....	59
listX11input Command.....	61
livedumpstart Command.....	62
lkdev Command.....	66
ln Command.....	67
locale Command.....	69
localedef Command.....	71
lock Command	73
lockd Daemon.....	74
locktrace Command	76
logevent Command.....	77
logform Command.....	78
logger Command.....	80
login Command	81
logins Command.....	84
logname Command	87
logout Command	88
look Command.....	88
lookbib Command	89
loopmount Command	90
loopumount Command	91
lorder Command.....	92
lp Command	93
lp.cat Command.....	99
lpacl Information.....	101
Setting or Changing the Printerlpadmin Command.....	104
lpar_netboot Command.....	114

lparstat Command.....	118
lpc Command.....	126
lpd Command.....	129
lpfilter Command.....	131
lpforms Command.....	136
lphistory Command.....	140
lpmove Command.....	145
lppchk Command.....	145
lppmgr Command.....	148
lpq Command.....	150
lpr Command	153
lprm Command	158
lpsched Command.....	161
lpstat Command.....	162
lpsystem Command.....	166
lptest Command.....	168
lpusers Command.....	168
ls Command	170
ls-secldapclntd Command.....	176
lsactdef Command.....	177
lsallq Command.....	181
lsallqdev Command.....	182
lsarm command.....	183
lsassocmap Command.....	184
lsattr Command.....	186
lsaudrec Command.....	191
lsauth Command.....	196
lsauthent Command.....	198
lsC2admin Command.....	199
lsCCadmin Command.....	200
lscfg Command.....	200
lscifscred Command.....	203
lscifsmnt Command.....	204
lsclass Command.....	205
lscluster Command.....	207
lscomg Command.....	211
lscondition Command.....	214
lscondresp Command.....	219
lsconn Command.....	224
lscons Command.....	226
lscore Command.....	227
lscosi Command.....	228
lsdev Command.....	230
lsdisp Command.....	237
lsdom Command.....	238
lsevent Command.....	239
lsfilt Command.....	243
lsfont Command.....	244
lsfs Command.....	245
lsgroup Command.....	246
lsiscsi Command.....	248
lsitab Command.....	250
lskbd Command.....	251
lskst Command.....	251
lsldap Command.....	253
lslicense Command.....	257
lspclacl Command.....	258
lspcmd Command.....	263

lspp Command.....	267
lspracl Command.....	272
lspriacl Command.....	278
lsprsac1 Command.....	283
lslv Command.....	288
lsmaster Command	292
lsmcode Command	293
lsmkysyb Command.....	295
lsmpl Command.....	298
lsmplio Command.....	299
lsnamsv Command.....	307
lsnfsexp Command	308
lsnfmnt Command	309
lsnim Command.....	310
lsnlspath Command.....	313
lsparent Command.....	313
lspath Command.....	315
lspriv Command.....	320
lsprtsv Command.....	321
lspis Command.....	322
lspv Command	324
lspprc Command.....	328
lsque Command.....	330
lsquedev Command	331
lsresource Command.....	332
lsresponse Command.....	335
lsrole Command	340
lsrpdomain Command.....	343
lsrpnode Command.....	346
lsrset Command.....	350
lsrsrc Command.....	352
lsrsrcassoc Command.....	358
lsrsrcdef Command.....	361
lssavevg Command.....	366
lssavewpar Command.....	369
lssec Command.....	371
lssecattr Command.....	373
lssecmode Command.....	376
lssensor Command.....	378
lsslot Command.....	385
lssmbcmnt Command.....	388
lssmbcred Command.....	389
lssrad Command.....	390
lssrc Command.....	391
lsts Command.....	394
lstun Command.....	395
lsuser Command	396
lsusil Command.....	399
lsvfs Command.....	399
lsvg Command.....	400
lsvgfs Command.....	404
lsvirprt Command.....	405
lsvmode Command.....	408
lsvpd Command	409
lsvsd Command.....	413
lswlmconf Command.....	416
lswpar Command.....	420
luit Command.....	431

lvmo Command.....	433
lvostat Command.....	435
lvupdateInit Command.....	438
lvupdateRegKE Command.....	440
lvupdateRegScript Command.....	441
lvupdateSafeKE Command.....	443
lvupdateSetProcs Command.....	445

m..... 447

m4 Command.....	447
mach Command.....	451
machstat Command.....	452
macref Command	453
mail Command.....	454
mailq Command.....	469
mailstats Command.....	471
make Command.....	472
makedbm Command.....	480
makedepend Command.....	481
makedev Command.....	483
makekey Command.....	484
makemap Command.....	485
man Command	486
manage_disk_drivers Command.....	491
managefonts Command.....	492
mant Command.....	494
mark Command.....	496
mesg Command.....	498
mhl Command	499
mhmail Command.....	502
mhpath Command.....	503
migratelp Command.....	505
migratepv Command.....	506
migwpar Command.....	507
mirrorvg Command.....	509
mirscan Command.....	512
mkauth Command.....	515
mkboot Command.....	518
mkC2admin Command.....	520
mkcatdefs Command.....	521
mkCCadmin Command.....	522
mkcd Command.....	523
mkcfsmnt Command.....	530
mkcifscred Command.....	531
mkcifsmnt Command.....	533
mkcimreg Command.....	535
mkclass Command.....	538
mkclient Command.....	541
mkcluster Command	542
mkcomg Command.....	546
mkcondition Command.....	551
mkcondresp Command.....	557
mkcosi Command.....	560
mkdev Command.....	561
mkdir Command.....	564
mkdirhier Command.....	566
mkdom Command.....	566

mkdvd Command.....	568
mkfifo Command.....	574
mkfilt Command.....	575
mkfont Command.....	576
mkfontdir Command.....	577
mkfs Command.....	578
mkgroup Command	583
mkhosts Command.....	586
mkiba Command.....	587
mkinstallp Command.....	589
mkiscsi Command.....	590
mkkitab Command.....	592
mkkeyserv Command.....	594
mkkrb5clnt Command.....	595
mkkrb5srv Command.....	597
mklost+found Command.....	599
mklpcmd Command.....	600
mklv Command.....	604
mklvcopy Command.....	613
mkmaster Command.....	616
mknamsv Command.....	617
mknetid Command.....	618
mknfs Command.....	619
mknfsexp Command.....	620
mknfsmnt Command.....	624
mknfsproxy Command.....	628
mknod Command.....	630
mknotify Command.....	632
mkpasswd Command	633
mkpath Command.....	635
mkprojldap Command.....	637
mkproto Command.....	639
mkprtldap Command.....	644
mkprtsv Command.....	647
mkps Command.....	650
mkqos Command.....	652
mkque Command.....	653
mkquedev Command.....	654
mkramdisk Command.....	656
mkresponse Command.....	658
mkrole Command.....	663
mkrpdomain Command.....	665
mkrset Command.....	673
mkrsrc Command.....	674
mkrtc Command.....	679
mkseckrb5 Command.....	680
mksecldap Command	682
mksecpki Command.....	689
mksensor Command.....	691
mkserver Command.....	697
mkslave Command	698
mksmbcmnt Command.....	699
mk smbcred Command.....	702
mkssys Command.....	703
mkstr Command.....	706
mksysb Command.....	707
mksysb_iso Command.....	712
mkyszfile Command.....	715

mktcpip Command.....	717
mkts Command.....	719
mktun Command.....	721
mkuser Command.....	722
mkuser.sys Command	726
mkusil Command.....	727
mkvg Command.....	728
mkvgdata Command.....	734
mkvirprt Command.....	735
mkwpar Command.....	738
mkwpardata Command.....	748
mm Command	749
mmt Command.....	751
mmtu Command.....	753
mobip6ctrl Command.....	754
mobip6reqd Daemon.....	756
monacct Command.....	756
mon-cxma Command.....	757
monitord Daemon.....	759
moo Command.....	760
more Command.....	760
mosy Command.....	766
mount Command.....	768
mountd Daemon.....	785
mpcstat Command.....	786
mpstat Command.....	788
mrouted Daemon.....	794
msgchk Command	797
msh Command.....	798
mt Command (BSD).....	800
mtrace Command.....	801
multibos Command.....	804
mv Command.....	808
mmdir Command.....	812
mvfilt Command.....	812
mvt Command	813
mwm Command.....	814
n.....	867
named Daemon – BIND Version 9.16.....	867
named-checkconf Command.....	867
named-checkzone Command.....	868
named9 Daemon.....	870
namerslv Command.....	873
ncheck Command.....	876
nddctl Command.....	877
ndp Command.....	878
ndpd-host Daemon.....	879
ndpd-router Daemon.....	880
ndx Command.....	885
neqn Command.....	886
netcd Daemon.....	887
netcdctl Command.....	889
netpmon Command	891
netrule Command.....	900
netstat Command.....	905
newaliases Command.....	920

newform Command.....	920
newgrp Command.....	923
newkey Command.....	924
news Command.....	925
next Command.....	926
nfs.clean Command	928
nfs4cl Command.....	929
nfs4smctl Command.....	931
nfsauthreset Command.....	931
nfsd Daemon.....	932
nfshostkey Command.....	934
nfshostmap Command.....	935
nfso Command.....	936
nfsrgyd daemon.....	945
nfsstat Command.....	946
nice Command.....	951
nim Command.....	953
nim_clients_setup Command.....	969
nim_master_recover Command.....	970
nim_master_setup Command.....	973
nim_move_up Command.....	976
nim_update_all Command.....	986
nimadapters Command.....	988
nimadm Command.....	994
nimclient Command.....	1001
nimconfig Command.....	1005
nimdef Command.....	1008
niminit Command.....	1013
niminv Command.....	1017
nimol_backup Command.....	1023
nimol_config Command.....	1024
nimol_install Command.....	1027
nimol_lslpp Command.....	1029
nimol_update Command.....	1031
nimquery Command.....	1032
nistoldif Command.....	1034
nl Command.....	1036
nlssrc Command.....	1039
nm Command.....	1041
nmon Command.....	1045
no Command.....	1070
nohup Command.....	1101
notifyevent Command.....	1103
nroff Command.....	1105
nslookup Command.....	1108
nsupdate Command.....	1110
ntpd4 Daemon.....	1114
ntpdate4 Command.....	1117
ntpd4 Command.....	1119
ntp-keygen4 Command.....	1126
ntpq4 Daemon.....	1130
ntptrace4 Command.....	1135
nulladm Command.....	1136
number Command.....	1137
nvmesed Command.....	1137
nxstat Command.....	1145

o.....	1149
od Command.....	1149
odmadd Command.....	1154
odmchange Command.....	1155
odmcreate Command.....	1156
odmdelete Command.....	1158
odmdrop Command.....	1158
odmget Command.....	1159
odmshow Command.....	1161
on Command.....	1161
openpts Command.....	1162
OS_install Command.....	1164
oslevel Command.....	1170
ospf_monitor Command.....	1172
p.....	1177
pac Command.....	1177
pack Command.....	1178
packf Command.....	1180
pagdel Command.....	1182
pagesize Command.....	1183
paginit Command.....	1184
paglist Command.....	1184
panel20 Command.....	1185
passwd Command.....	1186
paste Command.....	1189
patch Command.....	1191
pathchk Command.....	1196
pax Command.....	1197
pcat Command.....	1213
pdelay Command.....	1214
pdisable Command.....	1215
pdlink Command.....	1216
pdmkdir Command.....	1217
pdmode Command.....	1218
pdrmdir Command.....	1219
pdset Command.....	1220
penable Command.....	1221
perfwb Command.....	1222
pg Command.....	1223
phold Command.....	1226
pic Command.....	1227
pick Command.....	1233
ping Command.....	1237
pioattred Command.....	1242
piobe Command.....	1244
pioburst Command.....	1246
piocvt Command.....	1247
piodigest Command.....	1248
piodmgr Command.....	1250
piofontin Command.....	1251
pioformat Command.....	1252
piofquote Command.....	1254
piolsvp Command.....	1254
piomgpdev Command.....	1257
piomkapqd Command.....	1258

piomkpg Command.....	1260
piomsg Command.....	1262
pioout Command.....	1263
piopredef Command.....	1265
pkgadd Command.....	1267
pkgask Command.....	1269
pkgchk Command.....	1272
pkginfo Command.....	1273
pkgmk Command.....	1275
pkgparam Command.....	1278
pkgproto Command.....	1279
pkgrm Command.....	1281
pkgtrans Command.....	1282
platform_dump Command.....	1284
plotgbe Command.....	1286
plotlbe Command.....	1287
pmctl Command.....	1288
pmcycles Command.....	1294
pmlist Command.....	1295
pmtu Command.....	1297
pop3d Daemon.....	1298
pop3ds Daemon.....	1299
portmap Daemon.....	1300
portmir Command.....	1301
post Command.....	1303
pppattachd Daemon.....	1304
pppcontrold Daemon.....	1307
pppdial Command.....	1312
pppstat Command.....	1313
pprof Command.....	1315
pr Command.....	1317
praliases Command.....	1320
prctmp Command.....	1320
prdaily Command.....	1321
preparevsd Command.....	1322
preprnode Command.....	1323
prev Command.....	1325
printenv Command.....	1327
printf Command.....	1327
probevctrl Command.....	1332
probevue Command.....	1336
proccred Command.....	1340
procfiles Command.....	1341
procflags Command.....	1343
procldd Command.....	1344
procmap Command.....	1345
procrun Command.....	1349
procsig Command.....	1350
procstack Command.....	1351
procstop Command.....	1353
proctree Command.....	1354
procwait Command.....	1357
procwdx Command.....	1358
prof Command.....	1359
proff Command.....	1361
projctl Command.....	1362
prompter Command.....	1370
proto Command.....	1372

proxymngr Command.....	1372
prs Command (SCCS).....	1374
prtacct Command.....	1378
prtconf Command.....	1379
prtgblconfig Command.....	1383
ps Command	1384
ps4014 Command.....	1404
ps630 Command.....	1405
psc Command.....	1406
pshare Command.....	1409
psplot Command	1410
psrasc Command.....	1411
psrev Command.....	1412
psroff Command.....	1413
pstart Command.....	1416
pstat Command.....	1417
ptpd Daemon.....	1418
ptsc Command.....	1425
ptsevt Command.....	1426
ptsevtd Command.....	1427
ptx Command.....	1427
pvcauth command.....	1429
pvi Command.....	1431
pwchange Command.....	1434
pwck Command.....	1436
pwd Command.....	1437
pwdadm Command.....	1438
pwdck Command.....	1440
pwtokey Command.....	1443
pxed Command.....	1446
q.....	1449
qadm Command.....	1449
qcan Command.....	1450
qchk Command.....	1451
qdaemon Command.....	1453
qhld Command.....	1454
qmov Command.....	1455
qosadd Command.....	1457
qoslist Command.....	1458
qosmod Command.....	1459
qosremove Command.....	1461
qosstat Command.....	1462
qpri Command.....	1463
qprt Command.....	1465
qstatus Command.....	1475
quiz Command.....	1477
quot Command.....	1479
quota Command.....	1480
quotacheck Command.....	1482
quotaon Command.....	1483
Notices.....	1485
Privacy policy considerations.....	1486
Trademarks.....	1487
Index.....	1489

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 *l*.

labck Command

Purpose

Checks for the consistency of the label-encodings file.

Syntax

```
labck [ -l ] [ -f encodings_file ]
```

```
labck [ { -c | -r } encodings_file ]
```

Description

You can use the **labck** command to verify that a label-encodings file is internally consistent. When specified without any flags, the **labck** command verifies the consistency of the system-default, label-encodings file and no message is displayed if the file is proper.

Flags

Item	Description
-l	Lists the system high sensitivity label, system low sensitivity label, system high integrity label, and system low integrity label as defined in the label-encodings file.
-f <i>encodings_file</i>	Uses the value that you specify for <i>encodings_file</i> instead of the system-default, label-encodings file.
-c <i>encodings_file</i>	Copies the contents of the system-default, label-encodings file into the <i>encodings_file</i> that you specify. If the file already exists, the command exits with an error.
-r <i>encodings_file</i>	Replaces the contents of the system-default, label-encodings file with the contents in the file that you specify using <i>encodings_file</i> . The contents are replaced only if the file name that you specify is a valid label-encodings file.

Security

Only authorized users can run the **labck** command.

Item	Description
aix.mls.lef	Required to perform the above operations on the label encodings file.

Files Accessed:

Item	Description
Mode	File
r	/etc/security/enc/LabelEncodings

Exit Status

The **labck** command returns the following exit values:

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

Examples

1. To check the consistency of the system-default, label-encodings file, enter the following command:

```
labck
```

2. To check the consistency of the system-default, label-encodings file and print the system the high and low labels, enter the following command:

```
labck -l
```

3. To check the consistency of a label-encodings file that is stored in the current directory, enter the following command:

```
labck -f ./labelencodingsfile
```

4. To copy the system-default, label-encodings file to a file with the name of **/tmp/lef**, enter the following command:

```
labck -c /tmp/lef
```

5. To replace the contents of system-default, label-encodings file with the contents of the **/tmp/lef** file, enter the following command:

```
labck -r /tmp/lef
```

Files

Item	Description
/usr/sbin/labck	Contains the labck command.
/etc/security/enc/LabelEncodings	System default label encodings file.

last Command

Purpose

Displays information about previous logins.

Syntax

```
last [ -X ] [ -f FileName ] [ -t Time ] [ -n Number | -Number ] [ Name ... ] [ Terminal ... ]
```

Description

The **last** command displays, in reverse chronological order, all previous logins and logoffs still recorded in the **/var/adm/wtmp** file. The **/var/adm/wtmp** file collects login and logout records as these events occur and holds them until the records are processed by the **acctcon1** and **acctcon2** commands as part of the daily reporting procedures. When the time daemon, **timed**, changes the system time, it logs entries in **wtmp** under the pseudo-user "date". An entry starting with "date |" is logged before the change, and one

starting with "date {" is logged after the change. This allows for accurate accounting of logins that span a time change.

The list can be restricted to:

- The number of lines specified either with the *-Number* parameter or with the **-n** flag.
- Logins or logoffs by the users specified by the *Name* parameter.
- Logins or logoffs from the terminals specified by the *Terminal* parameter.
- A terminal can be named fully or abbreviated as a tty. For example, you can specify either the `tty0` terminal or the `0` terminal.

Note: If you specify both a *Name* and *Terminal* parameter, the **last** command displays all logins and logoffs meeting either criterion.

For each process, the **last** command displays the:

- Time the session began
- Duration
- Terminal (tty) used

If applicable, the following information is included:

- Terminations due to rebooting
- Sessions that are still continuing

If the **last** command is interrupted, it indicates how far the search has progressed in the `/var/adm/wtmp` file. If interrupted with a **quit** signal, the command indicates how far the search has progressed and then continues the search. The **quit** signal can be any one of the following:

```
#define SIGQUIT 3 /* (*) quit,
generated from terminal special char */

#define SIGKILL 9 /* kill (cannot be caught or ignored) */

#define SIGTERM 15 /* software termination signal */
```

The **kill** command sends the default SIGTERM signal when it is invoked without any option. If you want to send the SIGQUIT signal, enter the following:

```
kill -3 (Process ID)
```

Flags

Item	Description
-f <i>FileName</i>	Specifies an alternate file from which to read logins and logoffs.
-n	Specifies the number of lines to be displayed on the list.

Item	Description
-t Time	Displays users logged in at the given Time value. The Time variable is specified in the decimal form [[CC]YY]MMDDhhmm[.SS] where: <ul style="list-style-type: none"> CC Specifies the first two digits of the year. YY Specifies the last two digits of the year. 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).
-X	Prints all available characters of each user name instead of truncating to the first 8 characters.

Examples

1. To display all the recorded logins and logoffs by user `root` or from the console terminal, type:

```
last root console
```

2. To display the time between reboots of the system, type:

```
last reboot
```

The `reboot` pseudo-user logs in when the system starts again.

3. To display all the users still logged in at 10.30 am on April 15th, enter:

```
last -t 04151030
```

4. To display 10 lines in the list, type:

```
last -n 10
```

5. To display all the recorded logins and logoffs without truncating the user name, type:

```
last -X
```

Files

Item	Description
/usr/bin/last	Contains the last command.
/var/adm/wtmp	Contains connect-time accounting data, including login, logoff, and shutdown records.

lastcomm Command

Purpose

Displays information about the last commands executed.

Syntax

```
lastcomm [ -X ][ Command ] [ Name ] [ Terminal ]
```

Description

The **lastcomm** command displays information, in reverse chronological order, about all previously executed commands that are still recorded in the **/var/adm/pacct** summary file. You need to run the **/usr/sbin/acct/startup** command before you can execute the **lastcomm** command.

The list the **lastcomm** command displays can be restricted to:

- Commands specified by the *Command* parameter.
- Commands executed by the user specified by the *Name* parameter.
- Commands from the terminal specified by the *Terminal* parameter.

A terminal can be named fully or abbreviated as a tty. For example, you can specify either the `tty0` terminal or the `0` terminal.

For each process, the following information is displayed:

- The name of the user who ran the process.
- Any flags the accounting facilities collected when the command executed. The following are valid flags:

Item	Description
------	-------------

- | | |
|----------|---|
| S | The root user executed the command. |
| F | The command ran after a fork, but without a following subroutine. |
| C | The command ran in PDP-11 compatibility mode. |
| D | The command terminated with the generation of a core file. |
| X | The command was terminated with a signal. |

- The name of the command under which the process was called.
- The seconds of CPU time used by the process.
- The time the process was started.

Flags

Item	Description
------	-------------

- | | |
|----|--|
| -X | Prints all available characters of each user name instead of truncating to the first 8 characters. |
|----|--|

Examples

1. To display information about all previously executed commands recorded in the **/var/adm/pacct** file, enter:

```
lastcomm
```

2. To display information about commands named `a.out` executed by the `root` user on the `tttyd0` terminal, enter:

```
lastcomm a.out root ttyd0
```

3. To display information about all previously executed commands recorded in the `/var/adm/pacct` file without truncating the user name, enter:

```
lastcomm -X
```

Files

Item	Description
<code>/usr/bin/lastcomm</code>	Contains the lastcomm command.
<code>/var/adm/pacct</code>	The directory that contains the current accounting summary files.

lastlogin Command

Purpose

Reports the last login date for each user on the system.

Syntax

```
/usr/sbin/acct/lastlogin [ -X ]
```

Description

The **lastlogin** command updates the `/var/adm/acct/sum/loginlog` file to show the last date each user logged in. Normally, the **runacct** command, running under the **cron** daemon, calls this command and adds the information to the daily report. However, the **lastlogin** command can also be entered by a user who is a member of the ADM group.

Note: You should not share accounting files among nodes in a distributed environment. Each node should have its own copy of the various accounting files.

Flags

Item	Description
<code>-X</code>	Processes all available characters for each user name instead of truncating to the first 8 characters. This flag will also cause the lastlogin command to write to the <code>/var/adm/acct/sumx/loginlog</code> file instead of the <code>/var/adm/acct/sum/loginlog</code> file.

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/wtmp</code>	The login and logout history file.
<code>/var/adm/acct/sum</code>	Cumulative directory for daily accounting records.

lbxproxy Command

Purpose

Low BandWidth X proxy.

Syntax

```
lbxproxy [ :<display> ] [ -help ] [ -display Display ] [ -motion Number ] [ -terminate | -reset ] [ -reconnect ] [ -I ] [ -nolbx ] [ -nocomp ] [ -nodelta ] [ -notags ] [ -nogfx ] [ -noimage ] [ -nosquish ] [ -nointernsc ] [ -noatomsfile ] [ -atomsfiles File ] [ -nowinattr ] [ -nograbcmap ] [ -norgbfile ] [ -rgbfile Path ] [ -tagcachesize ] [ -zlevel Level ] [ -compstats ] [ -nozeropad ] [ -cheaterrors ] [ -cheatevents ]
```

Description

The **lbxproxy** command accepts client connections, multiplexes them over a single connection to the X server, and performs various optimizations on the X protocol to make it faster over low bandwidth and/or high latency connections. Applications that would like to take advantage of the Low Bandwidth extension to X (LBX) must make their connections to an **lbxproxy**. These applications need to know nothing about LBX, they simply connect to the **lbxproxy** as if were a regular server.

For authentication/authorization, **lbxproxy** passes the credentials presented by the client along to the server. Since X clients connect to **lbxproxy**, it is important that the user's **.Xauthority** file contain entries with valid keys associated with the network ID of the proxy. **lbxproxy** does not get involved with how these entries are added to the **.Xauthority** file. The user is responsible for setting it up.

The **lbxproxy** program has various flags, all of which are optional.

If **:<Display>** is specified, the proxy uses the *Display* port when listening for connections. The display port is an offset from port 6000, identical to the way in which regular X display connections are specified. If no port is specified on the command line, **lbxproxy** defaults to port 63. If the port that the proxy tries to listen on is in use, the proxy exits with an error message.

At startup, **lbxproxy** pre-interns a configurable list of atoms. This allows **lbxproxy** to intern a group of atoms in a single round trip and immediately store the results in its cache. While running, **lbxproxy** uses heuristics to decide when to delay sending window property data to the server. The heuristics depend on the size of the data, the name of the property, and whether a window manager is running through the same **lbxproxy**. Atom control is specified in the **AtomControl** file, set up during installation of **lbxproxy**, with command line overrides.

The file is a simple text file. There are three forms of lines: comments, length control, and name control. Lines starting with a **!** (exclamation point) are treated as comments. A line of the form **z length** specifies the minimum length in bytes before property data is delayed. A line of the form **options atomname** controls the given atom, where **options** is any combination of the following characters: **i** means the atom should be pre-interned; and **w** means data for properties with this name should be delayed only if a window manager is also running through the same **lbxproxy**.

Flags

Item	Description
-atomsfile <i>File</i>	Overrides the default AtomControl file.
-cheaterrors	Allows cheating on X protocol for the sake of improved performance. The X protocol guarantees that any replies, events or errors generated by a previous request are sent before those of a later request. This puts substantial restrictions on when lbxproxy can short circuit a request. The -cheaterrors flag allows lbxproxy to violate X protocol rules with respect to errors. Use at your own risk.
-cheatevents	The -cheatevents flag allows lbxproxy to violate X protocol rules with respect to events as well as errors. Use at your own risk.

Item	Description
-compstats	Reports stream compression statistics every time the proxy resets or receives a SIGHUP signal.
-display <i>Display</i>	Specifies the address of the X server supporting the LBX extension. If this flag is not specified, the display is obtained by the DISPLAY environment variable.
-help	Prints a brief help message about the command line flags.
-I	Causes all remaining arguments to be ignored.
-motion <i>Number</i>	Specifies the maximum <i>Number</i> of events that can be in flight. A limited number of pointer motion events are allowed to be in flight between the server and the proxy at any given time. The default is 8.
-noatomsfile	Disables reading of the AtomControl file.
-nocomp	Disables stream compression.
-nodelta	Disables delta request substitutions.
-nogfx	Disables reencoding of graphics requests (not including image related requests).
-nograbcmap	Disables colormap grabbing.
-noimage	Disables image compression.
-nointernsc	Disables short circuiting of InternAtom requests.
-nolbx	Disables all LBX optimizations.
-norgbfile	Disables color name to RGB resolution in proxy.
-nosquish	Disables squishing of X events.
-notags	Disables usage of tags.
-nowinattr	Disables GetWindowAttributes/GetGeometry grouping into one round trip.
-nozeropad	Indicates to not zero out unused pad bytes in X requests, replies, and events.
-reconnect	Causes lbxproxy to reset (see -reset) and attempts to reconnect to the server when its connection to the server is broken. The default behavior of lbxproxy is to exit.
-rgbfile <i>Path</i>	Specifies an alternate RGB database <i>Path</i> for color name to RGB resolution.
-tagcachesize	Sets the size of the proxy's tag cache (in bytes).
-[terminate reset]	The default behavior of lbxproxy is to continue running as usual when it's last client exits. The -terminate option will cause lbxproxy to exit when the last client exits. The -reset option will cause lbxproxy to reset itself when the last client exits. Resetting causes lbxproxy to clean up it's state and reconnect to the server.
-zlevel <i>Level</i>	Set the Zlib compression level (used for stream compression). The default is 9. 1 = worst compression, fastest. 9 = best compression, slowest.

ld Command

Purpose

Links object files.

Syntax

```
ld [ -D Number ] [ -e Label ] [ -G ] [ -H Number ] [ -K ] [ -m ] [ -M ] [ -o Name ] [ -r ] [ -s ] [ -S Number ] [ -T Number ] [ -u Name ] ... [ -v ] [ -V ] [ -z ] [ -Z String ] ... [ -b Option ] ... [ -L Directory ] ... { -f FileID ... -I Name ... InputFile ... }
```

or

```
ld -bsvr4 [ -d[y | n] ] [ -D Number ] [ -e Label ] [ -G ] [ -H Number ] [ -K ] [ -m ] [ -M ] [ -o Name ] [ -r ] [ -R Path ] [ -s ] [ -S Number ] [ -T Number ] [ -u Name ] ... [ -v ] [ -V ] [ -z [ defs | nodefs ] ] [ -z multidefs ] [ -z [text | nowarntext | warntext] ] [ -Z String ] ... [ -b Option ] ... [ -L Directory ] ... { -f FileID ... -I Name ... InputFile ... }
```

Description

The **ld** command, also called the linkage editor or binder, combines object files, archives, and import files into one output object file, resolving external references. It produces an executable object file that can be run. In addition, if you specify the **ld** command without the **-s** flag, you can use the output file as an *InputFile* parameter in another call to the **ld** command. By default, the **ld** command creates and places its output in the **a.out** file.

The **ld** command can relink a program without requiring you to again list all input object files. For example, if one object file from a large program has changed, you can relink the program by listing the new object file and the old program on the command line, along with any shared libraries required by the program. See “Examples” on page 36.

The **ld** command links input files in the order you specify on the command line. If you specify a file more than once, only the first occurrence of the file is processed. You must specify at least one input file, either with the **-bI** (uppercase letter i), **-bimport**, **-bkeepfile**, **-f**, or **-l** (lowercase letter L) flag or as an *InputFile* parameter. (The **-bI**, **-bimport**, or **-bkeepfile** flag is the **-b** flag used with the **I**, **import**, or **keepfile** option.)

Use the **cc** command to link files when you are producing programs that run under the operating system. Because the **cc** command calls the **ld** command with common options and necessary support libraries, you do not need to specify them on the command line. (This information is read from the **/etc/xlC.cfg** or **/etc/vac.cfg** configuration file.)

Linking Mode

The **ld** command can link 32-bit objects and programs as well as 64-bit objects and programs, but 32-bit and 64-bit objects may not be linked together. To specify the mode for linking, you can use the **OBJECT_MODE** environment variable or the **-b32** or **-b64** options.

Archive Files

Archive files are composite objects, which usually contain import files and object files, including shared objects. If an archive file contains another archive file or a member whose type is not recognized, the **ld** command issues a warning and ignores the unrecognized member. If an object file contained in an archive file has the **F_LOADONLY** bit set in the XCOFF header, the **ld** command ignores the member. This bit is usually used to designate old versions of shared objects that remain in the archive file to allow existing applications to load and run. New applications link with the new version of the shared object, that is, another member of the archive.

Shared Objects

A shared object, usually created by another call to the **ld** command, is an object file with the **F_SHROBJ** bit set in the XCOFF header. A shared object defines external symbols that are resolved at run time. If you specify the **-bnso** or **-bnoautoimp** option, the **ld** command processes a shared object as an ordinary object file, and if the file is stripped, the link fails.

Ordinarily, a shared object used as input is only listed in the loader section of the output file if a symbol in the shared object is actually referenced. When the run-time linker is used, however, you might want shared objects to be listed even if there are no symbols referenced. When the **-brtl** option is used, all shared objects listed on the command-line that are not archive members are listed in the output file.

The system loader loads all such shared objects when the program runs, and the symbols exported by these shared objects may be used by the run-time linker. Shared objects that are archive members are not loaded automatically unless automatic loading is enabled by an import file in the archive. To enable automatic loading, see [“Import and export File Format \(-bI: and -bE: Flags\)”](#) on page 32.

Import and Export Files

Import files are ASCII files that identify the external symbols to resolve at run time. An import file identifies the shared object defining the imported symbols. The system loader finds and resolves those symbols at run time. If the first line of an import file begins with `#!` (`#`, exclamation point), you can specify the file on the command line as an ordinary *InputFile*. Otherwise, you must use the **-bI** or **-bimport** option to specify the import file.

Export files are ASCII files that identify external symbols that are made available for another executable object file to import. The file format of an export file is the same as the file format of an import file.

Libraries

Libraries are files whose names end in `.a`, or possibly `.so`. To designate a library, you can specify an absolute or relative path name or use the **-l** (lowercase letter L) flag in the form **-l Name**. The last form designates a **libName.a** file, or when the **rtl** option is used, a **libName.so** file to be searched for in several directories. These search directories include any directories that are specified by **-L** flags and the standard library directories `/usr/lib` and `/lib`.

Note: If you specify a shared object, or an archive file that contains a shared object, with an absolute or relative path name, instead of with the **-l Name** flag, the path name is included in the import file ID string in the loader section of the output file. You can override this behavior with the **-bnoipath** option.

Processing

The **ld** command processes all input files in the same manner, whether they are archives or not. It includes the symbol tables of all objects, discarding only symbol definitions that duplicate existing symbols. Unlike some other versions of the **ld** command, you do not need to order archive files so references precede definitions. Furthermore, you do not need to list an archive file more than once on the command line.

The order of the **ld** command flags does not affect how they are processed, except for the flags used with input object files, libraries, and import files. These flags are **-L**, **-f**, **-l** (lowercase letter L), **-bkeepfile**, and **-bI** (uppercase letter i). The flags are processed in the following order:

1. The **-L** flag adds a directory to the list of search directories to locate libraries specified by the **-l** (lowercase letter L) flag. The directories are searched in the order specified. All **-L** flags are processed before any **-l** flags are processed.
2. The **ld** command processes the *InputFile* parameters, the files specified by the **-f** flag and libraries specified by the **-l** (lowercase letter L) flag in the order specified.
3. The **ld** command processes import files specified by the **-bI** (uppercase letter i) flag in the order specified after processing all other object files and libraries. You can specify an import file as an input file without the **-bI** flag if it is necessary to process the file before processing some object files. In this case, the first line of the import file must begin with the `#!` (`#`, exclamation point) symbols, and the import file is processed with other input files as described in step 2.
4. The **-bkeepfile** option names an input file on which the **ld** command does not perform garbage collection. If the specified input file is also specified as an *InputFile* parameter or listed in a file that is specified by the **-f** flag, the **-bkeepfile** option does not affect the order in which the file is processed. Otherwise, the file is processed in order along with other input files, as described in step 2.

An output file that is produced by the **ld** command has execute permission set, unless you specify the **-r** flag or **-bnox** option or errors were reported while linking. An existing output file is not

overwritten if any severe errors occurred, or if the output file was specified as an input file and any errors occurred.

Symbols

The **ld** command uses the following predefined symbols to provide special address locations and can be declared in C syntax as **extern char** *name* [*value*]. The symbol names are:

Item	Description
<code>_text</code>	Specifies the first location of the program.
<code>_etext</code>	Specifies the first location after the program.
<code>_data</code>	Specifies the first location of the data.
<code>_edata</code>	Specifies the first location after the initialized data
<code>_end</code> or <code>end</code>	Specifies the first location after all data.

The only way to use these symbols is to take their addresses. If an input file redefines any of these symbols, there might be unpredictable results. An additional predefined symbol, `_ptrgl`, is used by compilers to implement calls by using function pointers.

Garbage Collection

By default, the **ld** command performs garbage collection, deleting control sections (CSECTs) that are not referenced when generating the output file.

A CSECT is an indivisible unit of coding or data. A CSECT references another CSECT if it contains a relocation entry (RLD) referring to a symbol contained in the other CSECT. A referenced CSECT causes all CSECTs it references to be referenced as well. In addition, a CSECT is referenced if it contains exported symbols, symbols that are specified with the **-u** flag, or the symbol designated as the entry point with the **-e** flag.

If a symbol is not referenced but is needed in the output file, you can export the symbol, specify the symbol with the **-u** flag, or suppress garbage collection. To suppress garbage collection, use the **-r** flag or **-bnogc** option. To suppress garbage collection for individual object files, use the **-bkeepfile** option or the **-bgcbypass** option. Even when garbage collection is suppressed, unreferenced internal symbols are deleted.

Ignored and Unsupported Flags

For compatibility with other versions of the **ld** command, some flags are recognized but ignored. These flags produce a message that states that the flag and its operand were ignored. An ignored flag does not cause the **ld** command to stop without further processing. The following flags are ignored:

```
-ANumber      -bnostricmpct  -n
-bfilelist    -bstricmpct   -N
-bfl          -BNumber      -Q
-bforceimp    -d            -RNumber
-bi           -i            -VNumber
-binsert      -j[Key:]Number -x
-bnoforceimp  -kKey:Path    -YNumber
```

Note: When the **-bsvr4** option is present on the **ld** command line, the **-R** and **-z** options are redefined.

Flags that the **ld** command does not support result in an error message. After all unsupported flags are diagnosed, the **ld** command stops without further processing.


Flags

The **ld** command conforms to the XPG Utility Syntax Guidelines, except that the argument `-` applies only to the next operand, not to the remaining operands on the command line. For example, in the command line if you type `ld -- -s -v`, the **-s** is treated as a file name and the **-v** is treated as a flag. To treat `-v` as a file name, type `ld -- -s -- -v .`

Note: Enter a flag with an operand with or without a space between the flag and the operand. You can specify numeric values in decimal, octal (with a leading 0), or hexadecimal (with a leading 0x or 0X) format. If you specify conflicting flags on the command line, the **ld** command accepts the latest flag and ignores earlier ones.

Item	Description
-b <i>Option</i>	Sets special processing options. This flag can be repeated. For more information about these options, see “Options (-bOptions)” on page 15.
-d [<i>y</i> <i>n</i>]	When -dy is specified, ld uses dynamic linking; this option is equivalent to the -b so option. When -dn is specified, ld uses static linking; this option is equivalent to the -b nso option. The default is -dy . This option is valid only when the -bsvr4 option is specified.
-D <i>Number</i>	Sets the starting address for the initialized data (the data section) of the output file to <i>Number</i> . If the specified number is -1, the data section starts immediately after the text section. By default, the data section begins at location 0. If both the -D and -bpD flags are specified, the latter flag takes precedence. Note: The system loader relocates the data section at run time, so the specified number affects the addresses that are listed in address maps or printed by utilities such as the dump or nm command.
-e <i>Label</i>	Sets the entry point of the executable output file to <i>Label</i> . The default entry point is <code>__start</code> (double underscore start).
-f <i>FileID</i>	Specifies a file that contains a list of input files to process. <i>FileID</i> must contain a list of input file names. Each line in <i>FileID</i> is treated as if it were listed separately on the ld command line. Lines in the file can contain shell pattern characters * (asterisk), [(left bracket),] (right bracket), and ? (question mark), which are expanded by using the glob subroutine and can designate multiple object files.
-G	The -G produces a shared object that is enabled for use with the runtime linker. The -G flag is equivalent to specifying the erok , rtl , nortllib , nosymbolic , noautoexp , and M:SRE options with the -b flag. Subsequent options can override these options.
-H <i>Number</i>	Aligns the text, data, and loader sections of the output file so that each section begins on a file offset that is a multiple of <i>Number</i> . If the specified number is 1, no alignment occurs. If the specified number is 0, the loader section is aligned on a word boundary, and the text and data sections are aligned on a boundary to satisfy the alignment of all CSECTS in the sections. The default value is 0. If the specified <i>Number</i> causes any CSECTS to be unaligned within the output file, the ld command issues a warning and the output executable file might not load or run.
-K	Aligns the header, text, data, and loader sections of the output file so that each section begins on a page boundary. This flag is equivalent to specifying -H <i>Number</i> , where <i>Number</i> is the page size of the machine on which ld is running.

Item	Description
-l <i>Name</i>	<p>Processes the <i>libName</i>.a file. In dynamic mode, with the rtl option, processes the <i>libName</i>.a or <i>libName</i>.so file. In all cases, directories that are specified by the -L flag or in the standard library directories (/usr/lib and /lib) are searched to find the file. In dynamic mode with the rtl option, the first directory that contains either <i>libName</i>.so or <i>libName</i>.a satisfies the search. If both files are found in the same directory, the <i>libName</i>.so file is used. You can repeat this flag. For more information about dynamic mode, see “Run-time Linking” on page 31.</p> <p>Note: The first definition of a symbol is kept, even if no reference to the symbol is seen when the archive is read. In other versions of the ld command, a symbol that is defined in an archive is ignored if no reference to the symbol is seen when the archive is read.</p>
-L <i>Directory</i>	<p>Adds <i>Directory</i> to the list of search directories that are used for finding libraries that are designated by the -l (lowercase letter L) flag. The list of directories, including the standard library directories, is also recorded in the output object file loader section for use by the system loader unless you use the -blibpath, -bno libpath, or -bsvr4 option. You can repeat this flag.</p>
-m or -M	<p>Lists to standard output the names of all files and archive members that are processed to create the output file. Shared objects and import files are not listed.</p>
-o <i>Name</i>	<p>Names the output file <i>Name</i>. By default, the name of the output file is a.out.</p>
-r	<p>Produces a nonexecutable output file to use as an input file in another ld command call. This file might also contain unresolved symbols. The -r flag is equivalent to specifying the erok, noglink, nox, and nogc options with the -b flag. (Subsequent options can override these options.)</p>
-R <i>Path</i>	<p>Valid only when the -bsvr4 option is present on the ld command line. It defines a colon-separated list of directories that are used to specify library search directories to the runtime linker. <i>Path</i>, if present and not NULL, is recorded in the output file's loader section. Then it is used when linking an executable file with shared libraries at run time. Multiple instances of this option are concatenated together with each <i>Path</i> separated by a colon.</p>
-s	<p>Strips the symbol table, line number information, and relocation information when creating the output file. Stripping saves space but impairs the usefulness of the debuggers. You can also strip an existing executable file by using the strip command.</p> <p>Note: Nonshared objects cannot be linked if they are stripped. A shared object can be stripped, but a stripped, shared object cannot be used when linking statically.</p>
-S <i>Number</i>	<p>Sets the maximum size (in bytes) allowed for the user stack when the output executable program file is run. This value is saved in the auxiliary header and used by the system loader to set the soft ulimit. The default value is 0.</p> <p>For more information about large user stacks and 32-bit programs, see Large Program Support under General programming concepts topic.</p>

Item	Description
-T <i>Number</i>	<p>Sets the starting address of the text section of the output file to <i>Number</i>. The default value is 0.</p> <p>If both the -T and -bpT flags are specified, the latter flag takes precedence.</p> <p>Note: The system loader relocates the text section at run time, so the specified number affects addresses that are listed in address maps or printed by utilities such as the nm or the dump command.</p> <p> Attention: If <i>Number</i> is 0x1000xxxxxxx and the program that is linked is a 64-bit program, the system loader loads the executable text into memory that starts from the segment at the address 0x10000000000.</p>
-u <i>Name</i>	This option prevents garbage collection of the external symbol <i>Name</i> . If the specified symbol does not exist, a warning is reported. You can repeat this flag.
-v	Writes additional information about binder command execution to the loadmap file.
-V	Writes the version string of ld command to standard error (stderr).
-z	In the absence of the -b svr4 option, the -z flag functions the same as the -K flag.
-z defs	This option forces a fatal error if any undefined symbols remain at the end of the link. This option is the default when an executable file is built. It is also useful when building a shared library to assure that the object is self-contained, that is, that all its symbolic references are resolved internally. This option is valid only when the -b svr4 option is specified. It is equivalent to the -b erno option.
-z nodefs	This flag allows undefined symbols. This option is the default when a shared library is built. When used with executable files, the behavior of references to such undefined symbols is unspecified. This option is valid only when the -b svr4 option is specified. It is equivalent to the -b erok option.
-z multidefs	This option allows multiple symbol definitions. By default, multiple symbol definitions that occur between relocatable objects (.o files) results in a fatal error condition. This option suppresses the error condition and allows the first symbol definition to be taken. This option is valid only when the -b svr4 option is specified.
-z text	In dynamic mode only, forces a fatal error if any relocations against the .text section remain. This option is valid only when the -b svr4 option is specified.
-z nowarntext	In dynamic mode only, allows relocations against all mappable sections, including the .text section. This option is the default when building a shared library. This option is valid only when the -b svr4 option is specified.
-z warntext	In dynamic mode only, warns if any relocations against the .text section remain. This option is the default when building an executable file. This option is valid only when the -b svr4 option is specified.
-Z <i>String</i>	Prefixes the names of the standard library directories with <i>String</i> when searching for libraries specified by the -l (lowercase letter L) flag. For example, with the -Z test and -lxyz flags, the ld command looks for the /test/usr/lib/libxyz.a and /test/lib/libxyz.a files. When the -Z <i>String</i> flag is used, the standard library directories are not searched. This flag has no effect on the library path information that is saved in the loader section of the output file. This flag is useful when developing a new version of a library. You can repeat this flag.

The Binder

The **ld** command verifies the command-line arguments and calls the binder (by default the `/usr/ccs/bin/bind` file), passing a generated list of binder subcommands. The binder program links the files. Although the binder is called by the **ld** command, you can start the binder directly. In this case, the binder reads commands from standard input.

Two options affect the calling of the binder. The **binder** option specifies which binder to call, and the **nobind** option prevents the **ld** command from calling a binder. Other binder options affect the binder subcommands that are generated.

If the **ld** command does not detect any errors in the options or command-line arguments, it calls the binder. The binder is called with a command line of the form:

```
bind [quiet_opt] [loadmap_opt]
```

The default value for *quiet_opt* is **quiet** and the default value for the *loadmap_opt* is the null string, so the default command line is:

```
/usr/ccs/bin/bind quiet
```

Options (-bOptions)

The following values are possible for the *Options* variable of the **-b** flag. You can list more than one option after the **-b** flag, separating them with a single blank.

Note:

1. In the following list of binder options, two option names that are separated by the word *or* are synonymous.
2. The *FileID* indicates a path name. You can use either a relative or a full path name.
3. For a nonrepeatable option that is followed by an argument, you can negate the option by using a null argument. That is, specify only the option and the colon.
4. If you specify conflicting options, the last one takes precedence.

Item	Description
32	Specifies 32-bit linking mode. In this mode, all input object files must be XCOFF32 files, or an error is reported. XCOFF64 archive members are ignored. For import or export files that specify the mode of certain symbols, 64-bit symbols are ignored. If both -b32 and -b64 options are specified, the last specified option is used. If neither option is specified, the mode is determined from the value of the environment variable <u>OBJECT_MODE</u> .
64	Specifies 64-bit linking mode. In this mode, all input object files must be XCOFF64 files, or an error is reported. XCOFF32 archive members are ignored. For import or export files that specify the mode of certain symbols, 32-bit symbols are ignored. If both -b32 and -b64 options are specified, the last specified option is used. If neither option is specified, the mode is determined from the value of the environment variable <u>OBJECT_MODE</u> .
asis	Processes all external symbols in mixed case. This option is the default. To process all external symbols in uppercase, see the caps option that follows.

Item	Description
aslr or aslr:[tdsmp]* or aslr:-	<p>Specifies the address space layout randomization for the program. The aslr option enables all the randomization attributes when only this option is used, and -baslr:-, disables all of the randomization attributes.</p> <p>If the aslr option is followed by a colon, individual attributes can be enabled. The following attributes denote text, data, stack, mmap, and private-libraries: <i>t</i>, <i>d</i>, <i>s</i>, <i>m</i>, and <i>p</i>. For the attributes that are not listed, the randomization settings remain disabled.</p> <p>Note: The <i>m</i> and <i>p</i> attributes cannot be specified for 32-bit programs.</p> <p>Some programs are compiled and linked such that relocatable addresses are mapped into the text section. This option requires the relocation of the addresses when the program is run. These programs fail if randomization is enabled for text or data. As a result, when text-section relocations exist, text and data randomization are not enabled unless the <i>t</i> and <i>d</i> attributes are explicitly specified when by using the aslr option.</p>
autoexp	<p>Automatically exports some symbols from the output module without having to list them in an export file. (This option does not export all symbols from the output module. Use the bexpall option to export all symbols.) This option is the default. Use this option when linking a main program. The linker assumes that you are linking a main program when you do not specify a module type (with the M or modtype option) beginning with S and you do not use the noentry option.</p> <p>When you use the autoexp option, if any shared object that is listed on the command-line imports a symbol from the special file <code>.</code> (dot), and the module that is linked contains a local definition of the symbol, the symbol is exported automatically.</p> <p>Other symbols are also exported automatically when you link with the rtl option. If a symbol defined in the module that is linked has one or more definitions that are exported from a shared object that are listed on the command-line, and if any of the definitions is a BSS symbol, the symbol is exported automatically. If the definition in the module that is linked is a BSS symbol, the symbol is exported with the nosymbolic attribute. Otherwise, the symbol is exported with the symbolic attribute. If the symbol is listed in an export file with another export attribute, the explicit attribute is used.</p> <p>If the autoexp option would automatically export a symbol, but the symbol is listed in an export file with the list attribute, the symbol is not exported.</p>
autoimp or so	<p>Imports symbols from any shared objects that are specified as input files. The shared objects are referenced but not included as part of the output object file. This option is the default.</p>
autoload: <i>path/file(member)</i>	<p>Automatically load the archive member.</p>
bigtls	<p>Generates extra code if the size of thread-local storage in the output object or program is larger than 64 KB and a compiler was used that generates direct references to local-exec or local-dynamic thread-local variables. Extra code is needed for every direct reference to a thread-local variable that cannot be addressed with a 16-bit offset. Because a program that contains extra code might have poor performance, it is better to reduce the number of thread-local variables that use direct references than to use the option. The default option is the nobigtls option.</p>
bigtoc	<p>Generates extra code if the size of the table of contents (TOC) is greater than 64 KB. Extra code is needed for every reference to a TOC symbol that cannot be addressed with a 16-bit offset. Because a program that contains generated code might have poor performance, reduce the number of TOC entries that are needed by the program before you use this option. The default is the nobigtoc option. When this option is used, the <code>glink</code> file <code>/usr/lib/glink1.o</code> is used in 32-bit mode and <code>/usr/lib/glink164.o</code> is used in 64-bit mode.</p>

Item	Description
>> bimpctr	Optimizes performance of the function calls in large 64-bit executable object files where a called function, which is defined in the same module as the calling function, is located at an offset of more than 32 MB from the calling location. This option works only when the -bpT flag that states the text origin is specified and when the aslr option for text is not enabled. <<<
>> bimpptrgl	Optimizes performance of the function calls in large 64-bit executable object files where the called function requires the <code>_ptrgl</code> symbol as the target function and the definition of the <code>_ptrgl</code> symbol is located at an offset of more than 32 MB from the calling location. The function call might have been made by using the function pointer or the called function is defined in a different module than the calling function. This option works only when the compiler flag -qnoinlg1ue is specified. <<<
bindcmds:FileID	Writes a copy of the binder commands that are generated by the ld command to <i>FileID</i> . You can redirect the resultant file as standard input to the binder program when the binder program is called as a stand-alone program. By default, no file is produced.
binder:FileID	Uses <i>FileID</i> as the binder called by the ld command. The default binder is the <code>/usr/ccs/bin/bind</code> file.
bindopts:FileID	Writes a copy of the binder program arguments to <i>FileID</i> . You can use the resultant file to start the binder program as a stand-alone program. By default, no file is produced.
C:FileID or calls:FileID	Writes an address map of the output object file to <i>FileID</i> . Symbols are sorted by section and then by address. For each symbol listed in the map, references from the symbol to other symbols are listed. By default, no file is produced. To learn more about the calls option, see “Address Maps” on page 34 .
caps	Processes all external symbols in uppercase. The default is the asis option.

Item**Description**

cdtors[:*incl*][:*nnn*][:*order*]]

The linker gathers information about C++ static constructor or destructor functions and saves this information in the output file. The *incl* suboption tells the linker which archive members to search when creating the saved information. Following are the possible values:

all

Searches all members of all archives for constructor or destructor functions. This option is the default.

mbr

Searches for constructor or destructor functions only if the member is included in the output file. Using the **mbr** value is equivalent to by using the **-qtwoLink** and **-bsvr4** options.

csect

Searches for the constructor or destructor function only in **csects** included from the archive. Using the **csect** value is equivalent to by using the **-qtwoLink** option without the **-bsvr4** option.

The *nnn* suboption specifies the priority of the output module. If a program loads multiple modules at the same time, the *nnn* suboption is used to control the order in which modules are initialized. (The priority is ignored if the output file is a program and not a shared object.) The priority can be in the range from -2^{31} to $2^{31}-1$. The default priority is 0. Values in the range from -2^{31} to $-2^{31}+1023$ are reserved for C++ runtime initialization.

The *order* suboption specifies the order in which individual constructor or destructor functions are called, for all functions with the same priority. Following are the possible values:

s

Sorts in an arbitrary order based on function names. This option is the default, and is compatible with the order that the XL C++ compiler uses.

c

Sorts in link order. Constructor or destructor functions in the first input file are initialized first. In archives, functions in the first member are initialized first.

r

Sorts in reverse link order. Constructor or destructor functions in the last input file are initialized first. In archives, functions in the last member are initialized first.

You can specify this option multiple times, but the last suboption that you specify is used. An unspecified suboption does not affect the current or default value. For example, **-bcdtors:csect:20:s -bcdtors:::r** is the same as **-bcdtors:csect:::20:r**. The default is **-bnocdtors**. If **-bcdtors** is specified, it is the same as **-bcdtors:all:0:s**.

Note:

1. If necessary, the XL C++ compiler produces this option automatically.
2. Functions that are specified with the **-binitfini** option are invoked independently of static constructor or destructor functions. For more information about the order of initialization, see the **dlopen** subroutine.

comprld or **crld**

Combines multiple relocation entries (RLDs) at the same address into a single RLD when possible. This option is the default.

cror15

Uses the **cror 15,15,15** (0x4def7b82) instruction as the special no-op instruction that follows a call instruction. The default value is **ori 0, 0, 0** (0x60000000). See the **nop** option.

Use this option when linking object files on the current level of the system that you intend to relink on AIX 3.1.

Item	Description
cror31	<p>Uses the cror 31, 31, 31 (0x4ffffb82) instruction as the special no-op instruction that follows a call instruction. The default value is ori 0, 0, 0 (0x60000000). See the nop option.</p> <p>Use this option when linking object files on the current level of the system that you intend to relink on AIX 3.2.</p>
D: Number [/dsa] or maxdata: Number [/dsa]	<p>Sets the maximum size (in bytes) that is allowed for the user data area (or user heap) when the executable program is run. This value is saved in the auxiliary header and used by the system loader to increase the soft data limit, if required. The default value is 0. When this option is used, the specified number of bytes are reserved for the user data area. The program might not explicitly map objects, by using shmat or mmap functions to virtual addresses that are reserved for the user data area.</p> <p>For 32-bit programs, the maximum value that is allowed by the system is 0x80000000 for programs that are running under Large Program Support and 0xD0000000 for programs that are running under Very Large Program Support. For more information, see Large Program Support under General programming concepts topic. When a nonzero value is specified, the user data area begins in segment 3, and the program uses as many segments as necessary to satisfy the specified maxdata value.</p> <p>For 64-bit programs, the maxdata option provides a guaranteed maximum size for the programs data heap. Any value can be specified but the data area cannot extend past 0x06FFFFFFFFF8 regardless of the maxdata value specified.</p>
datapsize: psize	<p>Requests <i>psize</i> page sizes in bytes for data. The value can be specified as a decimal, hexadecimal, or octal number. The number specifications are the same as in the C programming language. Also, the page size can be specified as a number followed by a one-character suffix:</p> <ul style="list-style-type: none"> • K or k for kilo or 0x400 bytes. • M or m for mega or 0x100000 bytes. • G or g for giga or 0x40000000 bytes. • T or t for tera or 0x1000000000 bytes. • P or p for peta or 0x400000000000 bytes. • X or x for exo or 0x100000000000000 bytes. <p>For example, either -b datapsize:16k or -b datapsize:0x4000 requests 0x4000 for data and set the F_VARPG bit in the XCOFF header.</p>
dbg: Option or debugopt: Option	<p>Sets a special debugging or control option. By default, no debug option is set.</p> <p>The dbg: loadabs or debugopt: loadabs option is used to indicate that the output program is loaded at the same address as the address specified by the -T and -D flags. In this case, a branch-absolute instruction is never changed to a (relative) branch instruction even if its target is a relocatable symbol. Similarly, a branch instruction is never changed to a branch-absolute instruction.</p>
delcsect	<p>Deletes all symbols in a CSECT if any symbol in the CSECT was defined by a previously read object file. This option prevents more than one instance of the same function from existing in the same program. For example, if a .o file defines function a() and b.o file defines functions a() and b(), linking a.o and b.o with the -bdelcsect option deletes symbols a() and b() from b.o. Thus, two instances of a() do not exist. The default is the nodelcsect option.</p>
dynamic or shared	<p>Cause the linker to process subsequent shared objects in dynamic mode. This option is the default. In dynamic mode, shared objects are not statically included in the output file. Instead, the shared objects are listed in the loader section of the output file. When you specify the rtl option and dynamic mode is in effect, files that ends in .so as well as .a satisfy searches for libraries specified with the -l (lowercase L) flag. When both are in effect, preference is given to .so instead of .a when present in the same directory. When you specify the rtl option and static mode is in effect, files that ends in .a are processed.</p>

Item	Description
E:FileID or export:FileID	Exports the external symbols that are listed in the file <i>FileID</i> . Exported symbols are listed in the loader section of the output file. There is no default export file. When the svr4 option is used, the E:FileID option cancels any expall or expfull options.
ernotok or f	Reports an error if there are any unresolved external references. This option is the default.
erok	Produces the output object file without errors even if there are unresolved external references. The default is the ernotok option.
errmsg	Writes error messages to standard error if the error level of the message is greater than or equal to the value of the halt option and the quiet option is used or standard output is redirected. This option is the default.
ex1:FileID , ex2:FileID , ex3:FileID , ex4:FileID , and ex5:FileID	Provide user exits in the typical binder subcommand sequence. Each file that is specified by <i>FileID</i> must contain a list of binder subcommands that runs as follows: ex1:FileID Before reading any <i>InputFiles</i> ex2:FileID Immediately before symbol resolution ex3:FileID Immediately after symbol resolution ex4:FileID Immediately before writing the output file ex5:FileID Immediately after writing the output file
expall	This option exports all global symbols, except imported symbols, unreferenced symbols that are defined in archive members, and symbols that begin with an underscore (_). You can export more symbols by listing them in an export file or by using the expfull option. This option does not affect symbols that are exported by the autoexp option. When you use this option, you might be able to avoid this by using an export file. However, by using an export file provides explicit control over which symbols are exported, and allows the usage of other global symbols within the shared object without worrying about conflicting with names that are exported from other shared objects. The default is noexpall .
expfull	This option exports all global symbols, except the imported symbols and the symbols that are defined in the archive members. Unlike the -bexpall option, the -bexpfull option exports symbols that begin with an underscore (_) and also exports the module entry point of the symbol. The actual behavior of the -bexpfull option might differ depending on the specific use case and the configuration of the runtime linker. The -bexpfull option does not affect symbols that are exported by the autoexp option. The default value is noexpfull unless the svr4 option is used with the -b flag.
export:FileID	Functions the same as the E:FileID option.
f	Functions the same as the ernotok option.
forceimprw	Forces read-only CSECTs that contain references to imported symbols to become read or write. The default is noforceimprw .
forkpolicy:policy	Sets the _AOUT_FORK_POLICY and _AOUT_FORK_COR flags in the XCOFF auxiliary header, when linking a 64-bit program. If <i>policy</i> is cor , the _AOUT_FORK_COR flag is also set, requesting the use of the copy-on-reference forktree policy when the program is run. If <i>policy</i> is cow , the _AOUT_FORK_COR flag is reset, requesting the use of the copy-on-write forktree policy when the program is run. When linking a 32-bit program, this flag is ignored. The default value is noforkpolicy .
gc	Performs garbage collection. Use the nogc , gcbypass , or keepfile option to prevent garbage collection for some or all object files. This option is the default.

Item	Description
gcbypass: <i>Number</i>	Specifies the number of files to bypass when garbage is collected if the gc option is specified. This option is ignored if the nogc option is used. If <i>Number</i> is 0, this option is equivalent to the gc option and garbage collection is performed for all files. The default value is 0 .
glink: <i>FileID</i> or glink=large or glink=small	Uses the global linkage prototype code specified by <i>FileID</i> . Global-linkage interface code is generated for each imported or undefined function. In 32-bit mode, the default is the /usr/lib/glink.o file, or /usr/lib/glinkl.o if the bigtoc option is used. In 64-bit mode, the default is the /usr/lib/glink64.o file, or /usr/lib/glinkl64.o if the bigtoc option is used. You can also use glink=large to specify /usr/lib/glinkl.o in 32-bit mode and /usr/lib/glinkl64.o in 64-bit mode, or glink=small to override a previous glink option.
h: <i>Number</i> or halt: <i>Number</i>	Specifies the maximum error level for binder command processing to continue. The default value is 4 . If any binder subcommand has a return value greater than <i>Number</i> , no additional binder subcommands are processed. If the halt level value is 8 or greater, the output file might not be an executable file if it is produced at all. Return values are: <p>0 No error</p> <p>4 Warning</p> <p>8 Error</p> <p>12 Severe error</p> <p>16 Internal program error</p>
I: <i>FileID</i> or import: <i>FileID</i>	(Uppercase i) Imports the symbols that are listed in <i>FileID</i> . There is no default import file.
initfini: [<i>Initial</i>] [: <i>Termination</i>] [: <i>Priority</i>]	Specifies a module initialization and termination function for a module, where <i>Initial</i> is an initialization routine, <i>Termination</i> is a termination routine, and <i>Priority</i> is a signed integer, with values from -2,147,483,648 to 2,147,483,647. Specify at least one of <i>Initial</i> and <i>Termination</i> , and if you omit both <i>Termination</i> and <i>Priority</i> , you must omit the colon after <i>Initial</i> as well. If you do not specify <i>Priority</i> , 0 is the default. This option can be repeated. <p>This option sorts routines by priority, starting with the routine with the smallest (most negative) priority. It invokes initialization routines in order, and termination routines in reverse order.</p> <p>This option invokes routines with the same priority in an unspecified order, but if multiple initfini options specify the same priority and both an initialization and termination routine, it preserves the relative order of the routines. For example, if you specify the options initfini:i1:f1 and initfini:i2:f2, then function i1 and i2 are invoked in an unspecified order, but if i1 is invoked before i2 when the module is loaded, f2 will be invoked before f1 when the module is unloaded.</p> <p>Note:</p> <ol style="list-style-type: none"> The priorities in the following inclusive ranges are reserved: <div style="background-color: #f0f0f0; padding: 10px; margin: 10px 0;"> <pre style="margin: 0;">-2,147,483,640 to -2,147,000,000 -1,999,999,999 to -1,000,000,000 -99,999,999 to -50,000,000 0 50,000,000 to 99,999,999 1,000,000,000 to 1,999,999,999 2,147,000,000 to 2,147,483,640</pre> </div> Functions that are specified with the -binitfini option are invoked independently of static constructor or destructor functions. For more information about the order of initialization, see the dlopen subroutine.

Item	Description
ipath	For shared objects listed on the command-line, rather than specified with the -l flag, use the path component when listing the shared object in the loader section of the output file. This option is the default.
keepfile:FileID	This option prevents garbage collection of <i>FileID</i> . By default, the binder deletes unreferenced CSECTS in all files. You can repeat this option.
lazy	<p>This option enables lazy loading of a module's dependent modules. This option adds a -lrtl option that follows other flags and options. If the -brtl option is specified, the -blazy option is ignored and lazy loading is not enabled.</p> <p>When a module is linked, a list of its dependent modules is saved in the module's loader section. The system loader automatically loads the dependent modules after the module is loaded. When lazy loading is enabled, loading is deferred for some dependents until a function is called in the module for the first time.</p> <p>A module is lazy loaded when all references to the module are function calls. If variables in the module are referenced, the module is loaded in the typical way.</p> <p>Note: Be careful while comparing function pointers if you are by using lazy loading. Usually a function has a unique address to compare two function pointers to determine whether they refer to the same function. When by using lazy loading to link a module, the address of a function in a lazy loaded module is not the same address that is computed by other modules. Programs that depend upon the comparison of function pointers must not use lazy loading.</p> <p>For more information about lazy loading, see Shared Libraries and Lazy Loading.</p>
l:FileID or loadmap:FileID	(Lowercase L) Writes each binder subcommand and its results to <i>FileID</i> . By default, no file is produced.
libpath:Path	<p>Uses <i>Path</i> as the library path when writing the loader section of the output file. <i>Path</i> is neither checked for validity nor used when searching for libraries specified by the -l flag. <i>Path</i> overrides any library paths that are generated when the -L flag is used.</p> <p>If you do not specify any -L flags, or if you specify the no libpath option, the default library path information is written in the loader section of the output file. The default library path information is the value of the LIBPATH environment variable if it is defined, and <code>/usr/lib:/lib</code>, otherwise.</p>
loadmap:FileID	Functions the same as the l:FileID option.
lpdata	Sets the F_LPDATA bit in the XCOFF header of the executable file. When this bit is set, the process is going to request large pages for its data.

Item	Description
M : <i>ModuleType</i> or modtype : <i>ModuleType</i>	Sets the two-character module-type field and the shared object flag in the object file. The module type is not checked by the binder, but it must be set to one of the following values: <p>1L Single use. Module requires a private copy of the data section for each load.</p> <p>RE Reusable. Module requires a private copy of the data area for each process dependent on the module.</p> <p>RO Read only. The module is read-only, and can be used by several processes at one time. If an S prefix is used on any of the preceding options, the shared flag in the object file is set. The system loader attempts to share a single instance of the data section of an RO module. Otherwise, the module type is ignored by the system loader. The default value is 1L.</p> <p>UR Sets the SGETUREGS flag for the linker. When the SGETUREGS flag is set, the contents of the registers are stored in a buffer. This option is used by coredump system call.</p>
map : <i>FileID</i> or R : <i>FileID</i>	Writes an address map of the output object file to <i>FileID</i> . Symbols are sorted by section and then by address. By default, no file is produced. To learn more about the map option, see “Address Maps” on page 34.
maxdata : <i>Number</i> [/ <i>dsa</i>]	Functions the same as the D : <i>Number</i> [/ <i>dsa</i>] option.
maxstack : <i>Number</i> or S : <i>Number</i>	Functions the same as the -S flag.
modtype : <i>ModuleType</i>	Functions the same as the M : <i>ModuleType</i> option.
n1 or noloadmap	This option does not write the binder subcommands and their results to a load map file. This option is the default.
noaslr	Cancels the effect of a previous aslr option on the command line.
noautoexp	Prevents automatic exportation of any symbols. The default is the autoexp option.
noautoimp or nso	Links any unstripped, shared objects as ordinary object files. When you use this option, the loader section of shared objects is not used. The default is the autoimp or so option. <p>Note: By using either of these flags, you can statically link a shared object file into an application. Any application that is statically linked is not binary portable from any fix or release level to any other fix or release level.</p>
nobigtls	This option generates a severe error message if the compiler generates direct references to thread-local variables and the relocation to one of the variables overflows because the size of the thread-local storage is greater than 64 KB. If an output file is produced, it does not run correctly. The nobigtls option is the default option.
nobigtoc	This option generates a severe error message if the size of the TOC is greater than 64 KB. If an output file is produced, it does not execute correctly. This option is the default.
nobind	Omits calling the binder. Instead, the ld command writes the generated list of binder subcommands to standard output. By default, the ld command calls the binder.

Item	Description
nocdtors	This option does not gather static constructor or destructor functions. This option is the default.
nocmprld or nocrld	This option does not combine multiple relocation entries (RLDs) at the same address into a single RLD. The default is the comprld or crld option.
nodekcsect	This option allows all symbols in the CSECT to be considered during symbol resolution, even if some symbol in the CSECT is defined in a previously read object file. For more information, see the delkcsect option. The nodekcsect option is the default.
noexpall	This option does not export symbols unless you list them in an export file or you export them with the autoexp option. This option is the default.
noexpfull	This option does not export symbols unless you list them in an export file or you export them with the autoexp option. This option is the default, unless the svr4 option is used.
noentry	This option indicates that the output file has no entry point. To retain any needed symbols, specify them with the -u flag or with an export file. You can also use the -r flag or the nogc or gcbtpass options to keep all external symbols in some or all object files. If neither the noentry nor the nox option is used and the entry point is not found, a warning is issued. This warning is suppressed when the svr4 option is used.
noerrmsg	This option does not write error messages to standard error. Use this option if you specify the noquiet option and you pipe standard output to a command such as tee or pg .
noforceimprw	This option allows read-only CSECTs to reference imported symbols. This option is the default.
noforkpolicy	Clears the _AOUT_FORK_POLICY and _AOUT_FORK_COR flags in the XCOFF auxiliary header, when linking a 64-bit program. The default forktree policy is used, unless a forktree policy is specified with the VMM_CNTRL environment variable. When linking a 32-bit program, this flag is ignored. This option is the default.
nogc	This option prevents garbage collection. CSECTs in all object files that contain global symbols are kept, whether they are referenced or not. The default is the gc option.
noglink	Prevents the ld command from inserting global linkage code. By default, the binder inserts the global linkage code.
noipath	For shared objects listed on the command-line, rather than specified with the -l flag, use a null path component when listing the shared object in the loader section of the output file. A null path component is always used for shared objects that are specified with the -l flag. This option does not affect the specification of a path component by using a line beginning with #! in an import file. The default is the ipath option.
nolibpath	Overrides any previous library path that is generated by the -L flag or specified by the libpath option. Instead, the default library path information is written in the loader section of the output file. The default library path information is the value of the LIBPATH environment variable if it is defined, and /usr/lib:/lib otherwise.
noloadmap	Functions the same as the nl option.
nolpdata	Clears the F_LPDATA bit in the XCOFF header of the executable file. When this bit is not set, the process is going to use small (regular) pages for its data.

Item	Description
nom	This option does not list the object files that are used to create the output file. This option overrides the -m flag. This option is the default.
noobjreorder	<p>This option does not use the depth-first CSECT reordering logic. The CSECTs in the output file are arranged in the same order that the object files and library files were specified on the command line, except as follows:</p> <ul style="list-style-type: none"> • CSECTs are placed in their correct text, data, or BSS section of the object file, based on the storage-mapping class field of each CSECT. • All CSECTs with a storage-mapping class of XMC_TC (TOC address constant) or XMC_TD (TOC variable) are grouped. <p>If both the noobjreorder and noreorder options are specified, the noreorder option takes precedence. The default is the reorder option.</p>
noorder_file	This option does not map symbols in a specified order. This flag negates the effect of a previous -border_file flag. This option is the default.
nop:Nop	<p>Specifies the no-op instruction that is used after branches to local routines. <i>Nop</i> can be one of the special values cror15, cror31, ori, or an eight-digit hexadecimal number. The ori instruction is the default. Specifying the -bnop:cror15 option is equivalent to specifying the -bcror15 option; specifying the -bnop:cror31 option is equivalent to specifying the -bcror31 option. If you specify one of the special nop options, all previous nop options are overridden.</p> <p>If <i>Nop</i> is an eight-digit hexadecimal number, it specifies an arbitrary machine instruction. This machine instruction overrides any previously specified special value for <i>Nop</i> instruction. When you use this form, you can repeat this option.</p> <p>The last machine instruction that is specified is the instruction that is generated by the binder after intra-module branches. Other specified machine instructions are recognized as no-op instructions, but are converted to the preferred no-op instruction.</p>
> > noplugin_opt	Discards any previous plug-in options that are specified with the plugin_opt option. <<<
noquiet	Writes each binder subcommand and its results to standard output. The default is the quiet option.
noreorder	This option does not reorder CSECTs, except to combine all XMC_TC (TOC address constant) and XMC_TD (TOC variable) CSECTs and place them in the data section, and combine all BSS symbols and place them in the bss section. All other CSECTs are placed in the text section, so text and data are mixed in the output file. When the noreorder option is used, the text section of the output file might no longer be position-independent, and the system loader will not load a module if the text section is not position-independent. Therefore, avoid by using this option for programs and kernel extensions. If both noobjreorder and noreorder options are specified, the noreorder option takes precedence. The default is the reorder option.
nortl	Disables runtime linking for the output file. This option implies the nortllib and nosymbolic - options. Furthermore, more actions that are described under the rtl option are not taken. This option is the default unless the svr4 option is used.

Item	Description
nortllib	This option does not include a reference to the runtime linker. If a main program is linked with this option, no runtime linking takes place in the program, regardless of the way any shared modules are linked that are used by the program. This option is the default unless the svr4 option is used.
norwexec	Specifies that if the system's sed_config setting is not off, the process' private data areas have nonexecute permission.
noshrsymtab	Prevents the _AOUT_SHR_SYMTAB flag from being set in the output object. This option is the default state.
nostabsplit	Prevents the debug section to be written to an alternate output file with the extension .stab . This option is the default setting.
nostrip	This option does not generate a stripped output file. Thus, the symbol table and relocation information are written in the output file. This option overrides the -s flag. This option is the default.
nosymbolic	Assigns the nosymbolic attribute to most symbols exported without an explicit attribute. For more information, see “Attributes of Exported Symbols” on page 34 . The default is the nosymbolic- option.
nosymbolic-	Assigns the nosymbolic- attribute to most symbols exported without an explicit attribute. For more information, see Attributes of Exported Symbols . This option is the default.
notextro or nro	This option does not check to ensure that there are no load-time relocation entries for the text section of the output object file. This option is the default.
notmprelname	The binder does not check for general instantiations. Note: This option is only needed for 32-bit mode. This option is ignored when you build 64-bit objects.
notypchk	This option does not check function-parameter types between external functional calls. The default is the typchk option.
nov	This option does not write additional information to the load map file. This option is the default and overrides the -v flag.
noweaklocal	Resolves weak symbols by using normal search order. This option overrides the weaklocal option. It is the default option.
nox	This option does not make the output file an executable file. Neither the auxiliary header nor the loader section is written. Flags and options that specify values that are written in the auxiliary header or loader section have no effect when this option is used. The default is the x option.
nro	Functions the same as the notextro option.
nso	Functions the same as the noautoimp option.
order_file:FileID	Maps symbols that are listed in <i>FileID</i> in the specified order. The symbols that are listed in the file are mapped before other symbols of the same storage-mapping class. Function names that are specified in the file must start with a dot because a function name without a dot denotes a function descriptor.

Item	Description
order : <i>Specification</i>	<p>Controls the order in which some symbols are mapped in the output file. The specifications are as follows:</p> <p>toc:fileref If the -border_file flag is used, then the TOC symbols that are referenced by any function that are listed in the order file, are mapped before other TOC symbols. Otherwise, this specification is ignored.</p> <p>toc:nofileref This option does not order TOC symbols that are based on the order file.</p> <p>notoc This option does not order TOC symbols in any special way.</p>
pD : <i>Origin</i>	<p>Specifies <i>Origin</i> as the address of the first byte of the file page that contains the beginning of the data section. For example, if the data section begins at an offset 0x22A0 in the object file, and pD: 0x20000000 is specified, the first byte of the data section is assigned address 0x200002A0. This option assumes a page size of 4096 (0x1000) bytes.</p> <p>Note: If both the -bpD and -D flags are specified, the latter flag takes precedence.</p>
>> plugin : <i>path</i>	<p>Specifies the full path of a compiler plug-in. This option is typically generated by a compiler, if required.</p> <p>If you want to cancel a previously specified plug-in path, do not specify the <i>path</i> variable that is -bplugin:.</p> <p>If an input file is a bitcode file or an archive that contains a bitcode file, and a plug-in path is not specified, the ld command fails.</p> <p><<<</p>
>> plugin_opt : <i>plugin-option</i>	<p>Specifies an option to be passed to the compiler plug-in. You can specify the <i>plugin_opt</i> option multiple times. This option is typically generated by a compiler.<<<</p>
pT : <i>Origin</i>	<p>Specifies <i>Origin</i> as the address of the first byte of the file page that contains the beginning of the text section. For example, if the text section begins at an offset 0x264 in the object file, and pT: 0x10000000 is specified, the first byte of the text section is assigned address 0x10000264.</p> <p>Note: If both the -bpT and -T flags are specified, the latter flag takes precedence. For more information, see the -T flag for additional information.</p>
quiet	<p>This option does not write binder subcommands and their results to standard output. This option is the default.</p>
Item	Description
R : <i>FileID</i>	<p>Functions the same as the map: <i>FileID</i> option.</p>
r or reorder	<p>Reorders CSECTs as part of the save command processing. The reorder process arranges CSECTs of the same storage-mapping class by proximity of reference. This option is the default.</p>
ras	<p>Sets a flag in the output module's auxiliary header to signify that the module is both storage key safe and recovery safe. For more information about how to make a kernel extension be key safe and recovery safe, see Kernel Extensions and Device Support Programming Concepts .</p>

Item	Description
rename : <i>Symbol</i> , <i>NewName</i>	Renames the external symbol <i>Symbol</i> to <i>NewName</i> . In effect, it is as if all definitions and references to <i>Symbol</i> in all object files were renamed to <i>NewName</i> before the files were processed. By default, symbols are not renamed.
reorder	Functions the same as the r option.
ro or textro	Ensures that there are no load-time relocation entries for the text section of the resultant object file. The default is the nro option.
rtl	<p>Enables runtime linking for the output file. This option implies the rtllib and symbolic options.</p> <p>When dynamic mode is in effect (see the dynamic and static options), the rtl option allows input files that are specified with the -l flag to end in .so as well as in .a.</p> <p>All input files that are shared objects are listed as dependents of your program in the output files loader section. The shared objects are listed in the same order as they were specified on the command line.</p> <p>A shared object that is contained in an archive is only listed if the archive specifies automatic loading for the shared object member. You can specify automatic loading for an archive member foo.o by creating an import file with the following lines:</p> <pre style="background-color: #f0f0f0; padding: 5px;"># autoload #! (foo.o)</pre> <p>After creating the import file, add the import file as a member to the archive. You might also specify automatic loading for an archive member foo.o by using the -bautoload option:</p> <pre style="background-color: #f0f0f0; padding: 5px;">-bautoload:<archive_name>(foo.o)</pre> <p>You might specify more archive members with -bautoloads option.</p> <p>If the first line of an import file begins with #! (number sign, exclamation point), you can specify the file on the command line as an ordinary <i>InputFile</i>. Otherwise, you must use the -bI or -bimport option to specify the import file.</p>
rtllib	This option includes a reference to the runtime linker. The runtime linker is defined in librtl.a , and an implicit -lrtl flag is added automatically to the command line. This option (implied by the rtl option) must be used when linking a main program or no runtime linking occurs. Shared objects do not have to be linked with this option. The default is the nortllib option unless the svr4 option is used.
rwexec	Specifies that the execute permissions of the process' private data areas are determined according to the system's sed_config setting. This option is the default.
rwexec_must	Specifies that the process' private data areas have execute permission, regardless of the system's sed_config setting.
S : <i>Number</i>	Functions the same as the -S flag.
scalls : <i>FileID</i>	Writes an address map of the object file to <i>FileID</i> . Symbols are listed alphabetically. For each symbol listed in the map, references from the symbol to the other symbols are listed. By default, no file is produced. To learn more about the scalls option, see “Address Maps” on page 34 .

Item	Description
shared	Functions the same as the dynamic option.
> shmpsize: <i>psize</i> <	<p>> Requests <i>psize</i> page sizes in bytes for a shared memory. The value that is specified must be a power of two. If not, the value is rounded down to the nearest power of two. The value can be specified as a decimal, hexadecimal, or octal number. Also, the page size can be specified as a number followed by a one-character suffix:</p> <ul style="list-style-type: none"> • K or k for kilo or 0x400 bytes. • M or m for mega or 0x100000 bytes. • G or g for giga or 0x40000000 bytes. • T or t for tera or 0x10000000000 bytes. • P or p for peta or 0x4000000000000 bytes. • X or x for exo or 0x1000000000000000 bytes. <p>For example, either <code>-b shmpsize:64k</code> or <code>-b shmpsize:0x10000</code> request a page size of 64 KB for shared memory. If the specified value is not 0, the <code>F_VARPG</code> bit of the XCOFF header's <code>f_flags</code> member is set. If 0 is specified, <code>F_VARPG</code> is cleared. The program uses the default page size for shared memory. In 32-bit mode, this flag is ignored and a warning message is printed.<</p>
shrsymtab	<p>In 64-bit mode, sets the <code>_AOUT_SHR_SYMTAB</code> flag in the XCOFF auxiliary header. If <i>File</i> is a 64-bit program, a shared symbol table is created when the program is run. If <i>File</i> is a 64-bit object but not a 64-bit program, the <code>_AOUT_SHR_SYMTAB</code> flag can be set, but has no effect at run time. In 32-bit mode, this flag is ignored. The default is noshrsymtab.</p>
smap: <i>FileID</i>	<p>Writes an address map of the object file to <i>FileID</i>. Symbols are listed alphabetically. By default, no file is produced. To learn more about the smap option, see Address Maps that follows.</p>
so	Functions the same as the autoimp option.
stabcmpct: <i>Level</i>	<p>Specifies the level of compaction for stabstrings in the debug section. Stabstrings are strings that are longer than eight characters. Each substring in the symbol table has its own offset in the debug section. The following values are valid for <i>Level</i>:</p> <p>0</p> <p>This option does not compact. Separate copies of duplicate stabstrings are written to the debug section.</p> <p>1</p> <p>Deletes duplicates. Each stabstring is written once to the <code>.debug</code> section. Duplicate stabstrings in the symbol table specifies the same offset into the debug section.</p> <p>2</p> <p>Renumbers the stabstrings and deletes most duplicates. (In some instances, multiple stabstrings can exist. They describe the same type but use different type numbers.) The scope of a type number is the entire output file, rather than a single input file as indicated by a <code>C_FILE</code> symbol table entry.</p> <p>If the binder does not recognize a stabstring, it returns an error message and the resulting executable file does not have valid stabstrings. The rest of the file is unaffected by the error.</p>

Item	Description
stabsplit	This option causes the debug section to be written to an alternate output file with the extension <i>.stab</i> .
stacksize:psize	Requests <i>psize</i> page sizes in bytes for process main thread stack. The value can be specified as a decimal, hexadecimal, or octal number. The number specifications are the same as in the C programming language. Also, the page size can be specified as a number followed by a one-character suffix: <ul style="list-style-type: none"> • K or k for kilo or 0x400 bytes. • M or m for mega or 0x100000 bytes. • G or g for giga or 0x40000000 bytes. • T or t for tera or 0x10000000000 bytes. • P or p for peta or 0x4000000000000 bytes. • X or x for exo or 0x1000000000000000 bytes. For example, either <code>-b stacksize:16k</code> or <code>-b stacksize:0x4000</code> requests 0x4000 for process main thread stack and set the F_VARPG bit in the XCOFF header.
static	This option causes the linker to process subsequent shared objects in static mode. In static mode, shared objects are statically linked in the output file.
svr4	This option changes the meaning of some other options on the command line and the standard behavior of the linker. It has the following effect on the linker: <ul style="list-style-type: none"> • <code>-b rtl</code> is set. • <code>-b rtllib</code> is set only when building an executable file or if not set explicitly to <code>-b nortllib</code>. • <code>-b symbolic</code> is set only when building an executable file or if not set explicitly by one of <code>-b symbolic</code>, <code>-b nosymbolic</code>, or <code>-b nosymbolic-</code>. • <code>-b expfull</code> is set only when neither <code>-b E</code> nor <code>-b export</code> are present. • <code>-b noexpall</code> is set. • <code>-d</code>, instead of being ignored, is redefined and can assume two values: -dy or -dn. • <code>-R</code>, instead of being ignored, takes one suboption that defines the runtime library search path. • <code>-z</code>, instead of being a synonym of the -K option, takes either defs, nodefs, multidefs, text, nowarntext, or warntext, as a suboption. • Directories that are specified with the -L option are not included in the runtime library search path.
sxref:FileID	Writes an address map of the object file to <i>FileID</i> . Symbols are listed alphabetically. For each symbol listed in the map, references to the symbol from other symbols are listed. By default, no file is produced. To learn more about the sxref option, see Address Maps that follows.
symbolic	Assigns the symbolic attribute to most symbols exported without an explicit attribute. For more information, see Attributes of Exported Symbols that follows. This option is the default when the svr4 option is used. Otherwise, the default is the symbolic- option.
textro	Same as the ro option.

Item	Description
textpsize: <i>psize</i>	<p>Requests <i>psize</i> page sizes in bytes for text. The value can be specified as a decimal, hexadecimal, or octal number. The number specifications are the same as in the C programming language. Also, the page size can be specified as a number followed by a one-character suffix:</p> <ul style="list-style-type: none"> • K or k for kilo or 0x400 bytes. • M or m for mega or 0x100000 bytes. • G or g for giga or 0x40000000 bytes. • T or t for tera or 0x10000000000 bytes. • P or p for peta or 0x400000000000 bytes. • X or x for exo or 0x100000000000000 bytes. <p>For example, either <code>-b textpsize:16k</code> or <code>-b textpsize:0x4000</code> requests 0x4000 for text and set the F_VARPG bit in the XCOFF header.</p>
tplrename	<p>Specifies that the binder must check for general instantiations. The binder checks for any symbol of the form <code>__tfnNxxx_name</code> and renames the symbol to <i>name</i>. The default is -bnotplrename.</p> <p>Note: This option is only needed for 32-bit mode. This option is ignored when building 64-bit objects.</p>
typchk	<p>Performs function-parameter type checking between external functional calls. The information of the Parameter-type checking can be included in object files by compilers and assemblers. This option is the default.</p> <p>For more information about type checking, see XCOFF Object File Format.</p>
weaklocal	<p>Specifies that weak symbols are searched for first in the object files where they are referenced. If the symbols are not found there, the normal search order is resumed.</p>
x	<p>Makes the output file an executable file, if no errors exist. This option is the default option.</p>
X or xref: <i>FileID</i>	<p>Writes an address map of the object file to <i>FileID</i>. Symbols are sorted by section and then by address. For each symbol listed in the map, references to the symbol from other symbols are listed. By default, no file is produced. To learn more about the xref option, see Address Maps that follows.</p>

Run-time Linking

By default, references to symbols in shared objects are bound at link time. That is, the output module associates an imported symbol with a definition in a specific shared object. At load-time, the definition in the specified shared object is used even if other shared objects export the same symbol.

You can cause your program to use the runtime linker, allowing some symbols to be rebounded at load-time. To create a program that uses the runtime linker, link the program with the **-brtl** option. The way that shared modules are linked affects the rebinding of symbols.

You can build shared objects that are enabled for runtime linking by using the **-G** flag. You can fully enable runtime linking for existing shared objects by relinking them with the **rtl_enable** command, unless they are stripped.

Symbol Visibility

Global and weak symbols in an input object file can be marked with a visibility symbol. Four symbol visibilities are defined.

<i>Table 1. Symbol Visibility</i>	
Symbol	Visibility
Internal	The symbol is not exported. The address of the symbol must not be provided to other programs or shared objects, but the linker does not verify this.
Hidden	The symbol is not exported.
Protected	The symbol is exported but cannot be rebounded (or preempted), even if runtime linking is being used.
Exported	The symbol is exported with the global export attribute.

The visibility of a symbol can be specified in an assembler source file. Some compilers support visibility as well. Consult your compiler documentation for details.

Export files can also be used to specify the visibility for a symbol. Ordinarily, the visibility that is specified in an export file takes precedence over the visibility that is specified in the object file. This linker considers symbol visibility when creating the export list for a program or shared object.

Import and export File Format (-bI: and -bE: Flags)

Each line within an import or export file contains the name of a symbol, optionally followed by an address or a keyword. Primary keywords are **svc**, **svc32**, **svc3264**, **svc64**, **syscall**, **syscall32**, **syscall3264**, **syscall64**, **symbolic**, **nosymbolic**, **nosymbolic-**, **list**, **cm**, **bss**, **internal**, **hidden**, **protected**, and **export**. A few more keywords are **weak** and **required**, which can be used along with another keyword.

In an import file, specifying an address allows a symbol to be mapped to a fixed address, such as an address in a shared memory segment. You can also use one of the keywords **cm**, **bss**, or **weak** to specify the storage class of an imported symbol. When the **autoexp** option is used, the storage class of an imported symbol affects which symbols are automatically exported. If any other keyword is specified in an import file, the keyword is ignored.

In an export file, you can use the **svc**, **svc32**, **svc3264**, **svc64**, **syscall**, **syscall32**, **syscall3264**, or **syscall64** keyword after a function name to indicate that the function is a system call. This option is needed when linking kernel extensions. If the output file is not a kernel extension, these keywords are equivalent to the **symbolic** keyword.

You can use the **list** keyword to cause a symbol to be listed in the loader section of the output file, although it is not marked as an exported symbol. The **list** keyword can be used for applications that want to process some symbols at run time. Listed symbols are not processed by the system loader or the runtime linker.

You can use the **symbolic**, **nosymbolic**, or **nosymbolic-** keyword to associate an attribute with an exported symbol. A symbol address in an export file is ignored. In an export file, the keywords **cm** and **bss** are equivalent to the **nosymbolic** keyword. The visibility of a symbol can be specified with the **internal**, **hidden**, **protected**, or **export** keywords. For more information, see [“Attributes of Exported Symbols” on page 34](#)

The **weak** keyword can be used to specify a weak symbol binding, and might be used with another attribute.

Use the **required** keyword to verify that a symbol is defined and not imported. An error is printed for symbols not meeting these criteria.

The **ld** command treats import and export files according to the following guidelines:

- A blank line is ignored.

- A line beginning with an * (asterisk) is a comment and is ignored.
- A line beginning with a # (#, blank space) provides operands to the **setopt** binder subcommand (**-bdbg:Option**). For example, a line that contains `# verbose` causes the binder to list each symbol as it is read from the file. These option settings are active only while processing the file. The `# 32`, `# 64`, `# no32`, and `# no64` options can be used to specify whether the listed symbols must be used for 32-bit links, 64-bit links, or both.

32-bit and 64-bit Import File Options

Item	Description
32	This option is used in an import or export file to specify that subsequent symbols must be processed when linking in 32-bit mode, but ignored when linking in 64-bit mode. If no 32 or 64 option is specified, all symbols are processed in both 32- and 64-bit modes.
64	This option is used in an import or export file to specify that subsequent symbols must be processed when linking in 64-bit mode, but ignored when linking in 32-bit mode. If no 32 or 64 option is specified, all symbols are processed in both 32- and 64-bit modes.
no32 or no64	Override a previous 32 or 64 . Subsequent symbols are processed in both 32- and 64-bit modes.

- When processing an import file, a line beginning with a `#!` (`#!`, exclamation point) provides the shared library name to be associated with subsequent import symbols. The line can occur more than once and applies to subsequent symbols until the next line beginning with `#!` is read. This file name information is placed in the loader section of the XCOFF object file. It is used by the system loader to locate the appropriate object file at execution time. If the import file name is **ipath/ifile** (**imember**), the file name placed in the loader section is determined based on the import file name and the contents of the `#!` line of the import file, as follows:

Item	Description
#!	(Nothing after the <code>#!</code> .) Use null path, null file, and null number. This option is treated as a deferred import by the system loader.
#! ()	Use ipath , ifile , and imember . This line can be used if the import file is specified as an <i>InputFile</i> parameter on the command line. The file must begin with <code>#!</code> in this case. This line can also be used to restore the default name if it was changed by another <code>#!</code> line.
#! path/file (member)	Use the specified path, file, and member.
#! path/file	Use the specified path and file, and a null member.
#! file	Use a null path, the specified file, and a null member. At run time, a list of directories is searched to find the shared object.
#! (member)	Use ipath , ifile , and the specified member. At run time, a list of directories is searched to find the shared object.
#! file (member)	Use a null path and specified file and member. At run time, a list of directories is searched to find the shared object.
#! .	(A single dot) This name refers to the main executable file. Use this file name when you are creating a shared object that imports symbols from multiple main programs with different names. The main program must export symbols that are imported by other modules, or loading fails. This import file name can be used with or without the runtime linker.
#! ..	(Two dots) Use this name to list symbols that are resolved by the runtime linker. Use this file name to create shared objects that are used by programs that makes use of the runtime linker. If you use a module that imports symbols from <code>..</code> in a program that was not linked with the rtllib option, the symbols are unresolved, and references to such symbols result in undefined behavior.

To automatically load archive members when the **-brtl** option is used, you can create an import file as follows. If `shr.so` is a shared object in an archive, create an import file:

```
# autoload
#! (shr.so)
```

You can list extra member names on more lines, if appropriate. You do not need to list symbol names in the import file because the symbols imported from `shr.so` are read from `shr.so` itself.

For more information about creating a shared library, see [create_shared_lib](#) subroutine. For more information about loading and binding, see the [load](#) subroutine.

Attributes of Exported Symbols

When you use runtime linking, a reference to a symbol in the same module can be rebounded only if the symbol is exported with the proper attribute. References to symbols with the **symbolic** attribute cannot be rebounded. References to symbols with the **nosymbolic** attribute can be rebounded. References to symbols with the **nosymbolic-** attribute can be rebounded if the symbols are variables. For function symbols, calls by using a function pointer can be rebounded, while direct function calls cannot be rebounded. The **nosymbolic-** attribute is the default and is provided for compatibility with previous versions of the operating system, but its use is not recommended.

If you are not by using the runtime linker, avoid by using the **nosymbolic** attribute because intra-module function calls are made indirectly through a function descriptor by using global-linkage code. Otherwise, the attribute of exported symbols has no effect for modules that are used with programs that do not use the runtime linker.

You can specify an explicit export attribute for symbols that are listed in an export file. Most symbols without an explicit attribute are exported with the default export attribute, as specified with the **symbolic**, **nosymbolic**, or **nosymbolic-** options.

If a symbol is listed in an export file without a keyword, and the visibility of the symbol is specified in an input file, the symbol's visibility is preserved. An input symbol's visibility can be overridden by using the **internal**, **hidden**, **protected**, or **export** keyword.

The **weak** export attribute marks the associated symbol's mapping type with `L_WEAK` in the loader section.

Imported symbols might have only the **weak** export attribute. If a symbol is imported from another module, all references to the symbol can be rebounded. However, if a symbol is imported at a fixed address, all references are bound to this fixed address and cannot be rebounded by the runtime linker. The system loader must resolve deferred imports. The runtime linker never resolves or rebinds references to deferred imports.

For exports of nonimported symbols, the following rules are used.

- If a symbol has the **list** attribute, it is listed in the loader section symbol table, but the **L_EXPORT** flag is not set in the symbol table entry. The runtime linker ignores such symbols.
- If a symbol was exported with an explicit attribute, or with an explicit visibility, the explicit attribute, or visibility is used.
- If the symbol is a BSS symbol, it is exported with the **nosymbolic** attribute.
- Otherwise, the symbol is exported with the global attribute, as specified by the **symbolic**, **nosymbolic**, or **nosymbolic-** option. The default global attribute is **nosymbolic-**.

Address Maps

The **ld** command generates address maps, listing the layout of symbols in the output object file. If you use the **map** (or **R**) option, unresolved symbols and imported symbols are listed first, followed by the symbols in each section in address order. If you use the **calls** (or **C**) option, each symbol that is listed is followed by a list of references from that symbol to other symbols. If you use the **xref** (or **X**) option,

each symbol that is listed is followed by a list of references to that symbol from other symbols. If you use the **smap**, **scalls**, or **sxref** option, the address map contains the same information as listed by the **map**, **calls**, or **xref** option, but symbols are listed in alphabetical order.

Internal symbols, with a storage class `C_HIDEXT`, are printed with the characters `<` and `>` (angle brackets) surrounding the symbol name. Names of external symbols, with a storage class `C_EXT`, are printed without the angle brackets, and symbols with a storage class of `C_WEAKEXT`, are printed with the characters `{` and `}` surrounding the symbol name.

The following information is listed about each symbol:

- An indication of whether the symbol is imported, exported, or the entry point. An `*` (asterisk) is used to mark the entry point, **I** is used to mark imported symbols, and **E** is used to mark exported symbols.
- Its address (except for imported symbols)
- Length and alignment (for CSECTs and BSS symbols)
- Storage-mapping class
- Symbol type
- Symbol number (used to differentiate between symbols of the same name)
- Symbol name
- Input file information

Storage-mapping classes and symbol types are defined in the `/usr/include/syms.h` file. In the address maps, only the last two characters are shown, except that the storage-mapping class **XMC_TC0** is shown as **TO**.

The input file information depends on the type of input file. For object files, source file names that are obtained from `C_FILE` symbols table entries are listed. If the object is from an archive file, the object file name is listed in the following format:

```
ArchiveFileName[ObjectName]
```

A shared object name is listed between `{ }` (braces). If a shared object is defined by an import file, the name of the import file is listed before the shared object name.

Import symbols have a symbol type of `ER`, but they have associated file input information. Undefined symbols are also listed with a symbol type of `ER`, but all other columns, except the symbol number, are left blank.

The **-T** and **-D** flags (or **pT** or **pD** options) affect the addresses that are printed in these address maps. For machine-level debugging, it is helpful to choose address so that symbols are listed with the same addresses that they have at run time. For a 32-bit program that does not use privately loaded shared objects, you can choose the proper addresses by specifying the `-bpT:0x10000000` and `-bpD:0x20000000` options. These options are defined by default in the `/etc/x1C.cfg` or `/etc/vac.cfg` file.

Environment Variables

The following environment variables affect the execution of the **ld** command:

Item	Description
<code>LIBPATH</code>	If <code>LIBPATH</code> is defined, its value is used as the default library path information. The default library path is <code>/usr/lib:/lib</code> . If no -L flags are specified and no -bllibpath option is specified, the default library path information is written in the loader section of the output file. Regardless of any options specified, <code>LIBPATH</code> is not used when searching for libraries that are specified from the command line.

Item	Description
<i>TMPDIR</i>	If the output file exists or it is on a remote file system, the ld command generates a temporary output file. The temporary output file is created in the directory that is specified by <i>TMPDIR</i> . If <i>TMPDIR</i> is not defined, the temporary output file is created in the <code>/tmp</code> directory if the output file is remote, or in the same directory as the existing output file.
<i>OBJECT_MODE</i>	If neither the -b32 nor -b64 option is used, the <i>OBJECT_MODE</i> environment variable is examined to determine the linking mode. If the value of <i>OBJECT_MODE</i> is 32 or 64 , 32-bit or 64-bit mode is used. If the value is 32_64 or any other value, the linker prints an error message and exits with a nonzero return code. Otherwise, a 32-bit mode is used.

Examples

1. To link several object files and produce a `a.out` file to run under the operating system, enter the following command:

```
ld /usr/lib/crt0.o pgm.o subs1.o subs2.o -lc
```

The **-lc** (lowercase letter L) links the `libc.a` library. A simpler way to link the `libc.a` library is to use the **cc** command (the compiler) to link the files as follows:

```
cc pgm.o subs1.o subs2.o
```

2. To specify the name of the output file, enter the following command:

```
cc -o pgm pgm.o subs1.o subs2.o
```

This creates the output in the file `pgm`.

3. To relink `pgm` if only the object file `subs1.o` changed, enter the following command:

```
cc -o pgm subs1.o pgm
```

The CSECTs that originally came from object files `pgm.o` and `subs2.o` are read from the file `pgm`. This technique can speed the linking process if a program consists of many input files, but only a few files change at a time.

4. To link with library subroutines, enter the following command:

```
cc pgm.o subs1.o subs2.o mylib.a -ltools
```

This links the object modules `pgm.o`, `subs1.o`, and `subs2.o`, the subroutines from the `mylib.a` archive, and the subroutine from the library that is specified by **-l** (lowercase letter L) flag (This means the `/usr/lib/libtools.a` file).

5. To generate a shared object, enter the following command:

```
ld -o shrsub.o subs1.o subs2.o -bE:shrsub.exp -bM:SRE -lc
```

This links the object files `subs1.o`, `subs2.o`, and the subroutines from the library `libc.a` specified by **-lc** flag. It exports the symbols that are specified in the file `shrsub.exp` and stores the linked shared object in the file `shrsub.o`. The **-bM:SRE** sets the shared object flag in the linked object file.

6. To link with the shared object `shrsub.o` generated previously, enter the following command:

```
cc -o pgm pgm.o shrsub.o -L '.'
```

This links the object file `pgm.o` with the exported symbols of `shrsub.o`. The linked output is stored in the object file `pgm`. The **-L '.'** adds the current directory to the library search path that the system loader uses to locate the `shrsub.o` shared object. At run time, this program is loaded only if it is run from a directory that contains an instance of the `shrsub.o` file or if the `shrsub.o` file is found

in the `/usr/lib` standard library directory. To allow the program to be run from anywhere, use the option `-L `pwd``.

The list of directories searched by the system loader can be seen by using the **dump** command.

7. To link a program by using the `libc.a` library as a nonshared library, enter the following command:

```
cc -o pgm pgm.o -bnso -bI:/lib/syscalls.exp
```

This links `pgm.o` with the necessary support libraries and names the output file `pgm`. For the **cc** command, the `libc.a` library is a necessary support library and is link-edited to the user's program as a shared library. In this example, the **-bnso** option directs the **ld** command to link with the `libc.a` library as a nonshared library, and the **-bI:/lib/syscalls.exp** directs the **ld** command to import the system call functions that are contained in the kernel or `/usr/lib/boot/unix` file. Whenever linking with the **-bnso** option, any symbols that were both imported and exported (that is, passed through) in a shared object must be explicitly imported, as is done by the **-bI:/lib/syscalls.exp** option in this example.

Note: Anytime that `/usr/lib/libc.a` is linked nonshared, the flag **-bI:/lib/syscalls.exp** must be used. The application can also have to be linked again whenever an updated release of the operating system is installed. Any application that is statically linked is not binary portable from any fix or release level to any other fix or release level.

8. To enable the randomization attributes, enter the following command:

```
-baslr
```

If text-section relocations exist, do not enable text and data randomization.

9. To enable randomization for text and stack only, enter the following command:

```
-baslr:ts
```

Files

Item	Description
<code>/usr/bin/ld</code>	Contains the ld command.
<code>/usr/lib/lib*.a</code>	Specifies libraries That are used for linking programs.
<code>a.out</code>	Specifies the default output file name.

Ldapgetusrattr Command

Purpose

Displays the value of an attribute for an LDAP user from the LDAP configured directory server.

Note: If an LDAP user is created with a UID value that is greater than 2^{31} , the **ldapgetusrattr** command displays it as a negative number.

Syntax

```
ldapgetusrattr <user_name> <ldap_attribute_name>
```

Description

The **ldapgetusrattr** command displays the value of an attribute for an LDAP user from the LDAP configured directory server. The **ldapgetusrattr** command queries the LDAP directory server by using the `secldaplntd` daemon and prints the result to the standard output (stdout) file.

Exit Status

This **ldapgetusrattr** command returns 0 after successful completion and returns nonzero value on failure. On failure, one for the following error messages are written to the standard error (stderr) file:

Item	Description
0	Indicates that the command completed successfully.
EIO	Indicates a connection error with LDAP.
EINVAL	Indicates that the command arguments are invalid or do not follow the expected usage.
EPERM	Indicates that you do not have permissions to run the command.
ENOMEM	Indicates insufficient memory to run the command.

Security

The **ldapgetusrattr** command is owned by the root user and the command has access permissions of 500. A root user or a user with the role that has the `aix.security.ldap` authorization can run the **ldapgetusrattr** command.

Example

1. To display the value of the `passwordminlength` attribute for the LDAP user `foo`, run the following command:

```
ldapgetusrattr foo passwordminlength
```

An output that is similar to the following example is displayed:

```
8
```

2. To display the value of the `sshPublicKey` attribute for the LDAP user `foo`, run the following command:

```
ldapgetusrattr foo sshPublicKey
```

An output that is similar to the following example is displayed:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGD1XAIquGStc6P07u7Y+3e5BeP608AxxCvCICGd/  
1V7jzzjKXI1o4ktFPqEUilHqw7Ragj  
zdXRg9jMeo2rg8oKye10CtswZGYunCDiFrBtw7cPSHcE1DCFw0yVu  
70I5pUwVgYeVzQIWI8t28PdAvJnfCm1QQZxQrgGk3RimNVrIFFHKgVbvG3Ck32K  
ChRSpz0FiI14ZaGgz1qvW1GAM4YD1zQ3pk/  
E5Gs80FaEuqxiDhmWow7joA5SmkBcmz4UZgPEns0nZnIPDAYPPHBD482rKf1e0qymr9F1p5gIPK70Qi6fr  
ilRdYK9e7ybyq16n8KzgJWgGbbZqkjyEJn/  
Xe0rLhMfiEeqcNC3Mq31g2M0tBGLojWyz4QSIUCQXsjeRV74E1SuB0zr14EBhiqJ8VQNr4sMfb1wXKPF6D0  
ivGY2w7tbthp7LE94fKAnYyMHEg67LQXVoaGw+EUcj6kJVnW1Hqly6Q2bMHmbiLHRudb+CAa8GUFuWsDxVmUn/  
PjyIAWc= vc17user@alp052-vc17.aus.stglabs.ibm.com
```

Restrictions

The **ldapgetusrattr** command is dependent on the `secldapc1ntd` daemon to query the LDAP server.

Idd Command

Purpose

Lists dynamic dependencies.

Syntax

ldd *FileName*

Description

The **ldd** command lists the path names of all dependencies. The command will report dependencies on only valid XCOFF files.

Parameters

Item	Description
<i>FileName</i>	Specifies the file whose dependencies will be listed.

Exit Status

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

Examples

- To display dependencies on **/usr/bin/dbx**, enter:

```
ldd /usr/bin/dbx
```

The output looks like the following:

```
/usr/bin/dbx needs:  
/usr/lib/libc.a(shr.o)  
/usr/lib/libdbx.a(shr.o)  
/unix  
/usr/lib/libcrypt.a(shr.o)  
/usr/lib/libpthdebug.a(shr.o)
```

Files

Item	Description
/usr/bin/ldd	Contains the ldd command.

Ldedit Command

Purpose

Modifies an **XCOFF** executable file header.

Syntax

ldedit **-b** *Option* [**-V**] *File*

Description

You can use the **ldedit** command to modify various fields in an **XCOFF** header or the auxiliary header of an executable file. The **ldedit** command makes it possible to mark or unmark an application as a 'large page data' program. The **ldedit** command also makes it possible to add or modify the values of MAXDATA and MAXSTACK without relinking.

The format of the **-b** flag is similar to the format used by the link editor, the **ld** command. The **-b** flag can be used multiple times on the command line.

If no flags are specified, the **ldedit** command displays a usage message using the standard error output.

Flags

Item	Description
-bOption	<p>Modifies an executable as specified by <i>Option</i>. The possible values for <i>Option</i> are:</p> <p>aslr or aslr:[+-[tdsmp]*]* aslr or aslr:[+-[tdsmp]*]* specifies the address space layout randomization for the program. The aslr option enables all of the randomization attributes when only this option is used.</p> <p>If the aslr option is followed by a colon, individual attributes can be enabled. The plus sign (+) enables randomization and the minus sign (-) disables randomization. The following attributes can be used to specify text, data, stack, mmap, and private-libraries: <i>t</i>, <i>d</i>, <i>s</i>, <i>m</i>, and <i>p</i>. For the attributes that are not listed, the randomization setting remains disabled.</p> <p>Note: The <i>m</i> and <i>p</i> attributes cannot be specified for 32-bit programs.</p> <p>Some programs are compiled and linked such that relocatable addresses are mapped into the text section. This requires the relocation of the addresses when the program is run. These programs fail if randomization is enabled for text or data. As a result, when text-section relocations exist, text and data randomization are not enabled unless the <i>t</i> and <i>d</i> attributes are explicitly specified when using the aslr option.</p>
noaslr	<p>Clears the randomization attribute settings of the program.</p>
datapsize:psize	<p>Requests <i>psize</i> page sizes in bytes for data. The value can be specified as a decimal, hexadecimal, or octal number. The number specifications are the same as in C programming language. Additionally, the page sizes can be specified as a number followed by a one-character suffix:</p> <ul style="list-style-type: none">• k or K for kilo or 0x400 bytes• m or M for mega or 0x100000 bytes• g or G for giga or 0x40000000 bytes• t or T for tera or 0x1000000000 bytes• p or P for peta or 0x400000000000 bytes• x or X for exo or 0x100000000000000 bytes <p>For example, either -b datapsize:16k or -b datapsize:0x4000 will request 0x4000 for data and set the F_VARPG bit in the XCOFF header. It is accomplished by setting the corresponding member of the auxiliary header to the logarithm base 2 of the given value <i>psize</i>. If the value is different from 0, the F_VARPG bit of the XCOFF header's <i>f_flags</i> member is also set. Otherwise, this bit is cleared.</p>
forkpolicy:policy	<p>Sets the _AOUT_FORK_POLICY flag in the XCOFF auxiliary header, if <i>File</i> is a 64-bit program. If <i>policy</i> is <i>cow</i>, the _AOUT_FORK_COW is also set, requesting the use of the copy-on-write forktree policy. If <i>policy</i> is <i>cow</i>, the _AOUT_FORK_COW flag is cleared, requesting the use of the copy-on-reference forktree policy. If <i>File</i> is a 32-bit program, no change is made.</p>
noforkpolicy	<p>Resets the _AOUT_FORK_POLICY and _AOUT_FORK_COW flags in the XCOFF auxiliary header, if <i>File</i> is a 64-bit program. The default forktree policy is used, unless a forktree policy is specified with the VMM_CNTRL environment variable. If <i>File</i> is a 32-bit program, no change is made.</p>
lpdata	<p>Marks a file as a 'large page data' executable.</p>
nolpdata	<p>Unmarks a file as a 'large page data' executable.</p>
noshrsymtab	<p>Clears the _AOUT_SHR_SYMTAB flag in the XCOFF auxiliary header. If <i>File</i> is a 32-bit object, no change is made.</p>
M: <modtype>	<p>Updates the module-type field and the shared object flag in the file. The F_SHROBJ flag is set in the XCOFF header when the module type begins with <i>S</i> character and is 3 characters long.</p>
maxdata:value	<p>Sets the MAXDATA value. <i>value</i> is an octal number when it starts with 0, a hexadecimal number when it starts with 0x, and a decimal number in all other cases.</p>
maxdata:value/dsa	<p>Sets the MAXDATA value and the DSA bit. <i>value</i> is an octal number when it starts with 0, a hexadecimal number when it starts with 0x, and a decimal number in all other cases.</p>

Item	Description
	<p>maxstack:value Sets the MAXSTACK value. <i>value</i> is an octal number when it starts with 0, a hexadecimal number when it starts with 0x, and a decimal number in all other cases.</p> <p>norwexec Marks a file's writable and mappable sections and stack as non-executable.</p> <p>shrsymtab Sets the _AOUT_SHR_SYMTAB flag in the XCOFF auxiliary header. If <i>File</i> is a 64-bit program, a shared symbol table is created when the program is run. If <i>Flag</i> is a 64-bit object but not a 64-bit program, the _AOUT_SHR_SYMTAB flag can be set, but has no effect at runtime. If <i>File</i> is a 32-bit object, no change is made.</p> <p>stacksize:psize Requests <i>psize</i> page sizes in bytes for process main thread stack. The value can be specified as a decimal, hexadecimal, or octal number. The number specifications are the same as in C programming language. Additionally, the page sizes can be specified as a number followed by a one-character suffix:</p> <ul style="list-style-type: none"> • k or K for kilo or 0x400 bytes • m or M for mega or 0x100000 bytes • g or G for giga or 0x40000000 bytes • t or T for tera or 0x1000000000 bytes • p or P for peta or 0x400000000000 bytes • x or X for exo or 0x100000000000000 bytes <p>For example, either <code>-b stacksize:16k</code> or <code>-b stacksize:0x4000</code> will request 0x4000 for process main thread stack and set the F_VARPG bit in the XCOFF header. It is accomplished by setting the corresponding member of the auxiliary header to the logarithm base 2 of the given value <i>psize</i>. If the value is different from 0, the F_VARPG bit of the XCOFF header's <code>f_flags</code> member is also set. Otherwise, this bit is cleared.</p> <p>textsize:psize Requests <i>psize</i> page sizes in bytes for text. The value can be specified as a decimal, hexadecimal, or octal number. The number specifications are the same as in C programming language. Additionally, the page sizes can be specified as a number followed by a one-character suffix:</p> <ul style="list-style-type: none"> • k or K for kilo or 0x400 bytes • m or M for mega or 0x100000 bytes • g or G for giga or 0x40000000 bytes • t or T for tera or 0x1000000000 bytes • p or P for peta or 0x400000000000 bytes • x or X for exo or 0x100000000000000 bytes <p>For example, either <code>-b textsize:16k</code> or <code>-b textsize:0x4000</code> will request 0x4000 for text and set the F_VARPG bit in the XCOFF header. It is accomplished by setting the corresponding member of the auxiliary header to the logarithm base 2 of the given value <i>psize</i>. If the value is different from 0, the F_VARPG bit of the XCOFF header's <code>f_flags</code> member is also set. Otherwise, this bit is cleared.</p>
-v	Prints the version of the ldedit command on the standard error output.

Examples

1. To request system-selected page sizes for text, data, and stacks, enter:

```
ldedit -b textsize:0 -b datapsize:0 -b stacksize:0
```

This clears the F_VARPG bit in the XCOFF header.

2. To enable all of the randomization attributes, enter:

```
-baslr
```

If text-section relocations exist, do not enable text and data randomization.

3. To enable the randomization for text, disable the randomization for stack, and leave the other values unchanged, enter:

```
-baslr:+t-s
```

learn Command

Purpose

Provides computer-aided instruction for using files, editors, macros, and other features.

Syntax

learn[- *Directory*] [*Subject*[*LessonNumber*]]

Description

The **learn** command provides computer-aided instruction for using files, editors, macros, and other features. The first time you invoke the command, the system provides introductory information about the **learn** command. Otherwise, the **learn** command begins at the point where you left the last **learn** command session.

You can bypass the default action of the **learn** command by specifying the *Subject* parameter. The **learn** command starts with the first lesson of the subject you specify. You can specify any of the following subjects:

- Files
- Editors
- More files
- Macros
- EQN (the enquiry character)
- C (the language)

Note: You can only run the EQN lesson on a hardcopy terminal that is capable of 1/2 line motion. The **/usr/share/lib/learn/eqn/Init** file contains a detailed list of the supported terminals.

When you enter the **learn** command, the system searches the **/usr/share/lib/learn** directory for the appropriate lesson file. Use the *-Directory* flag to identify a different search directory.

Subcommands

- The **bye** subcommand terminates a **learn** command session.
- The **where** subcommand tells you of your progress; the **where m** subcommand provides more detail.
- The **again** subcommand re-displays the text of the lesson.
- The **again LessonNumber** subcommand lets you review the lesson.
- The **hint** subcommand prints the last part of the lesson script used to evaluate a response; the **hint m** subcommand prints the entire lesson script.

Parameters

Item	Description
<i>-Directory</i>	Allows you to specify a different search directory. By default, the system searches for lesson files in the /usr/share/lib/learn directory.
<i>LessonNumber</i>	Identifies the number of the lesson.
<i>Subject</i>	Specifies the subject you want instruction on.

Examples

To take the online lesson about files, enter:

```
learn files
```

The system starts the **learn** program and displays instructions for how to use the program.

Files

Item	Description
<code>/usr/share/lib/learn</code>	Contains the file tree for all dependent directories and files.
<code>/tmp/pl*</code>	Contains the practice directories.
<code>\$HOME/.learnrc</code>	Contains the startup information.

leave Command

Purpose

Reminds you when you have to leave.

Syntax

```
leave [ [ + ] hhmm ]
```

Description

The **leave** command waits until the specified time and then reminds you that you have to leave. You are reminded at 5 minutes and at 1 minute before the actual time, again at that time, and at every minute thereafter. When you log off, the **leave** command exits just before it would have displayed the next message.

If you do not specify a time, the **leave** command prompts with `When do you have to leave?` A reply of `newline` causes the **leave** command to exit; otherwise, the reply is assumed to be a time. This form is suitable for inclusion in a **.login** or **.profile** file.

The **leave** command ignores `interrupt`, `quit`, and `terminate` operations. To clear the **leave** command, you should either log off or use the **kill-9** command and provide the process ID.

Flags

Item	Description
<code>+</code>	Specifies to set the alarm to go off in the indicated number of hours and minutes from the current time.
<code>hhm</code> <code>m</code>	Specifies a time of day in hours and minutes (based on a 12- or 24-hour clock) or, if preceded by the <code>+</code> , a set number of hours and minutes from the current time for the alarm to go off. All times are converted to a 12-hour clock and assumed to relate to the next 12 hours.

Examples

To remind yourself to leave at 3:45, enter:

```
leave 345
```

lecstat Command

Purpose

Displays operational information about an Asynchronous Transfer Mode network protocol (ATM) Local Area Network (LAN) Emulation Client.

Syntax

lecstat [**-a -c -q -r -s -t -v**] Device_Name

Description

This command displays ATM LAN Emulation Client (LEC) operational information gathered by a specified LEC device. If a LEC device name is not entered, statistics for all available LEC's appear. Select a flag to narrow down your search results. You can display specific categories of information such as Configuration, LE_ARP Cache Entries, Virtual Connections, and Statistics, or you can choose to display all of the information categories.

You can also toggle debug tracing on or off and reset statistics counters.

Parameters

Item	Description
<i>Device_Name</i>	The name of the LE Client device, for example, <i>ent1</i> .

Flags

Item	Description
-a	Requests all the LE Client information. This flag does not reset statistics counters or toggle trace.
-c	Requests the configuration.
-q	Requests the LE_ARP cache.
-r	Resets the statistics counters after reading.
-s	Requests the statistics counters.
-t	Toggles full debug trace on or off.
-v	Requests the list of virtual connections.

The following information appears for all valid calls and contains the following fields:

Device Type

Displays a description of the LAN Emulation Client (example: Ethernet or Token Ring)

LAN MAC Address

Displays the LAN Emulation Client's 6-byte Ethernet or Token Ring MAC address.

ATM Address

Displays the LAN Emulation Client's 20-byte Asynchronous Transfer Mode (ATM) address.

Elapsed Time

Displays the real time period which has elapsed since statistics were last reset.

Driver Flags

The current LAN Emulation Client(LEC) device driver NDD status flags. Example status flags:

Broadcast	Allowing broadcast packets.
Dead	Requires re-open.
Debug	Internal debug tracing enabled.
Limbo	Attempting ELAN recovery.
Running	Fully operational on the ELAN.
Up	Device has been opened.

Configuration Information

Selected with the **-a** or **-c** flags. Displays the network administrator's pre-configured attributes, as well as the current ELAN configuration values as defined by the LANE Servers.

Lane LE_ARP Table Entries

Selected with the **-a** or **-q** flags. Displays the current LE Client ARP cache. Included are the type of entry, it's state, the remote LAN MAC address or route descriptor, the remote ATM address and some descriptive values.

Example Types

BUS-PP	Broadcast and Unknown Server (point-to-point).
BUS-MP	Broadcast and Unknown Server (multi-point).
Data	Data (point-to-point).
LES-PP	LE Server (point-to-point).
LES-MP	LE Server (multi-point).

Example States

Arping	Attempting to locate remote client/server via LE_ARP.
Connected	Fully connected to the remote client/server.
Flushing	Flushing the data path to the client/server.
Known	Remote address is known but no connection yet.
Unknown	Remote address in unknown and not able to LE_ARP yet.

Lane Servers and Statistics

Selected with the **-a** or **-s** flags. Displays the current Transmit, Receive, and General statistics for this LE Client, as well as the ATM addresses of the current and available LANE Servers.

Lane connections

Selected with the **-a** or **-v** flags. Displays the current list of virtual connections in use by this LE Client. Included are virtual path and channel values, remote ATM address, and some descriptive values such as whether this connection was started by the remote, whether it is a duplicate connection, or whether the remote station is proxied by another LE Client.

Exit Status

If you specify an invalid *Device_Name*, this command produces error messages stating that it could not connect to the device. Examples of an invalid device error message might be:

```
LECSTAT: No LANE device configured.
LECSTAT: Device is not a LANE device.
LECSTAT: Device is not available.
```

lex Command

Purpose

Generates a C or C++ language program that matches patterns for simple lexical analysis of an input stream.

Syntax

```
lex [ -C ] [ -t ] [ -v | -n ] [ File... ]
```

Description

The **lex** command reads *File* or standard input, generates a C language program, and writes it to a file named **lex.yy.c**. This file, **lex.yy.c**, is a compilable C language program. A C++ compiler also can compile the output of the **lex** command. The **-C** flag renames the output file to **lex.yy.C** for the C++ compiler.

The C++ program generated by the **lex** command can use either `STDIO` or `IOSTREAMS`. If the `cpp` define `_CPP_IOSTREAMS` is true during a C++ compilation, the program uses `IOSTREAMS` for all I/O. Otherwise, `STDIO` is used.

The **lex** command uses rules and actions contained in *File* to generate a program, **lex.yy.c**, which can be compiled with the **cc** command. The compiled **lex.yy.c** can then receive input, break the input into the logical pieces defined by the rules in *File*, and run program fragments contained in the actions in *File*.

The generated program is a C language function called **yylex**. The **lex** command stores the **yylex** function in a file named **lex.yy.c**. You can use the **yylex** function alone to recognize simple one-word input, or you can use it with other C language programs to perform more difficult input analysis functions. For example, you can use the **lex** command to generate a program that simplifies an input stream before sending it to a parser program generated by the **yacc** command.

The **yylex** function analyzes the input stream using a program structure called a finite state machine. This structure allows the program to exist in only one state (or condition) at a time. There is a finite number of states allowed. The rules in *File* determine how the program moves from one state to another.

If you do not specify a *File*, the **lex** command reads standard input. It treats multiple files as a single file.

Note: Since the **lex** command uses fixed names for intermediate and output files, you can have only one program generated by **lex** in a given directory.

lex Specification File

The input file can contain three sections: *definitions*, *rules*, and *user subroutines*. Each section must be separated from the others by a line containing only the delimiter, `%%` (double percent signs). The format is:

```
definitions
%%
rules
%%
user subroutines
```

The purpose and format of each are described in the following sections.

Definitions

If you want to use variables in your rules, you must define them in this section. The variables make up the left column, and their definitions make up the right column. For example, if you want to define `D` as a numerical digit, you would write the following:

```
D [0-9]
```

You can use a defined variable in the rules section by enclosing the variable name in `{}` (braces), for example:

```
{D}
```

Lines in the definitions section beginning with a blank or enclosed in `%{`, `%}` delimiter lines are copied to the **lex.yy.c** file. You can use this construct to declare C language variables to be used in the **lex** actions or to include header files, for example:

```
%{
#include <math.h>
int count;
%}
```

Such lines can also appear at the beginning of the rules section, immediately after the first %% delimiter, but they should not be used anywhere else in the rules section. If the line is in the definitions section of *File*, the **lex** command copies it to the external declarations section of the **lex.yy.c** file. If the line appears in the rules section, before the first rule, the **lex** command copies it to the local declaration section of the **yylex** subroutine in **lex.yy.c**. Such lines should not occur after the first rule.

The type of the **lex** external, **ytext**, can be set to either a null-terminated character array (default) or a pointer to a null-terminated character string by specifying one of the following in the definitions section:

```
%array      (default)
%pointer
```

In the definitions section, you can set table sizes for the resulting finite state machine. The default sizes are large enough for small programs. You may want to set larger sizes for more complex programs.

Item Description

m

%a Number of transitions is *n* (default 5000)

n

%e Number of parse tree nodes is *n* (default 2000)

n

%h Number of multibyte character output slots (default is 0)

n

%k Number of packed character classes (default 1000)

n

%m Number of multibyte "character class" character output slots (default is 0)

n

%n Number of states is *n* (default 2500)

n

%o Number of output slots (default 5000, minimum 257)

n

%p Number of positions is *n* (default 5000)

n

%v Percentage of slots vacant in the hash tables controlled by **%h** and **%m** (default 20, range $0 \leq P < 100$)

p

%z Number of multibyte character class output slots (default 0)

n

If multibyte characters appear in extended regular expression strings, you may need to reset the output array size with the **%o** argument (possibly to array sizes in the range 10,000 to 20,000). This reset reflects the much larger number of characters relative to the number of single-byte characters.

If multibyte characters appear in extended regular expressions, you must set the multibyte hash table sizes with the **%h** and **%m** arguments to sizes greater than the total number of multibyte characters contained in the **lex** file.

If no multibyte characters appear in extended regular expressions but you want **!** to match multibyte characters, you must set **%z** greater than zero. Similarly, for inverse character classes (for example, **[^abc]**) to match multibyte characters, you must set both **%h** and **%m** greater than zero.

When using multibyte characters, the **lex.yy.c** file must be compiled with the **-qmbcs** compiler option.

Rules

Once you have defined your terms, you can write the rules section. It contains strings and expressions to be matched by the **yylex** subroutine, and C commands to execute when a match is made. This section is required, and it must be preceded by the delimiter **%%** (double percent signs), whether or not you have a definitions section. The **lex** command does not recognize your rules without this delimiter.

In this section, the left column contains the pattern in the form of an extended regular expression, which will be recognized in an input file to the **yylex** subroutine. The right column contains the C program fragment executed when that pattern is recognized, called an *action*.

When the lexical analyzer finds a match for the extended regular expression, the lexical analyzer executes the action associated with that extended regular expression.

Patterns can include extended characters. If multibyte locales are installed on your system, patterns can also include multibyte characters that are part of the installed code set.

The columns are separated by a tab or blanks. For example, if you want to search files for the keyword **KEY**, you can write the following:

```
(KEY) printf ("found KEY");
```

If you include this rule in *File*, the **yylex** lexical analyzer matches the pattern **KEY** and runs the **printf** subroutine.

Each pattern can have a corresponding action, that is, a C command to execute when the pattern is matched. Each statement must end with a **;** (semicolon). If you use more than one statement in an action, you must enclose all of them in **{ }** (braces). A second delimiter, **%%**, must follow the rules section if you have a *user subroutine* section. Without a specified action for a pattern match, the lexical analyzer copies the input pattern to the output without changing it.

When the **yylex** lexical analyzer matches a string in the input stream, it copies the matched string to an external character array (or a pointer to a character string), **yytext**, before it executes any commands in the rules section. Similarly, the external int, **yylen**, is set to the length of the matched string in bytes (therefore, multibyte characters will have a size greater than 1).

User Subroutines

The **lex** library defines the following subroutines as macros that you can use in the rules section of the **lex** specification file:

Item	Description
input	Reads a byte from yyin .
unput	Replaces a byte after it has been read.
output	Writes an output byte to yyout .
winput	Reads a multibyte character from yyin .
wunput	Replaces a multibyte character after it has been read.
woutput	Writes a multibyte output character to yyout .
yysetlocale	Calls the setlocale (LC_ALL, " ") ; subroutine to determine the current locale.

The **winput**, **wunput**, and **woutput** macros are defined to use the **yywinput**, **yywunput**, and **yywoutput** subroutines coded in the **lex.yy.c** file. For compatibility, these **yy** subroutines subsequently use the **input**, **unput**, and **output** subroutines to read, replace, and write the necessary number of bytes in a complete multibyte character.

You can override these macros by writing your own code for these routines in the user subroutines section. But if you write your own, you must undefine these macros in the definition section as follows:

```
%{  
#undef input  
#undef unput  
#undef output
```

```
#undef winput
#undef wunput
#undef woutput
#undef yysetlocale
%}
```

There is no **main** subroutine in **lex.yy.c**, because the **lex** library contains the **main** subroutine that calls the **yylex** lexical analyzer, as well as the **yywrap** subroutine called by **yylex()** at the end of *File*. Therefore, if you do not include **main()**, **yywrap()**, or both in the user subroutines section, when you compile **lex.yy.c**, you must enter `cc lex.yy.c -ll`, where **ll** calls the **lex** library.

External names generated by the **lex** command all begin with the preface **yy**, as in **yyin**, **yyout**, **yylex**, and **yytext**.

Finite State Machine

The default skeleton for the finite state machine is defined in `/usr/ccs/lib/lex/ncform`. The user can use a personally configured finite state machine by setting an environment variable **LEXER=PATH**. The *PATH* variable designates the user-defined finite state machine path and file name. The **lex** command checks the environment for this variable and, if it is set, uses the supplied path.

Putting Blanks in an Expression

Normally, blanks or tabs end a rule and therefore, the expression that defines a rule. However, you can enclose the blanks or tab characters in " " (quotation marks) to include them in the expression. Use quotes around all blanks in expressions that are not already within sets of [] (brackets).

Other Special Characters

The **lex** program recognizes many of the normal C language special characters. These character sequences are:

Sequence	Meaning
<code>\a</code>	Alert
<code>\b</code>	Backspace
<code>\f</code>	Form Feed
<code>\n</code>	Newline character (Do not use the actual newline character in an expression.)
<code>\r</code>	Return
<code>\t</code>	Tab
<code>\v</code>	Vertical Tab
<code>\\</code>	Backslash
<code>\digits</code>	The character with encoding represented by the one-, two-, or three-digit octal integer specified by digits .
<code>\xdigits</code>	The character with encoding represented by the sequence of hexadecimal characters specified by digits .
<code>\c</code>	Where c is none of the characters listed above, represents the character c unchanged.

Note: Do not use `\0` or `\x0` in **lex** rules.

When using these special characters in an expression, you do not need to enclose them in quotes. Every character, except these special characters and the operator symbols, is always a text character.

Matching Rules

When more than one expression can match the current input, the **lex** command chooses the longest match first. When several rules match the same number of characters, the **lex** command chooses the rule that occurs first. For example, if the rules

```
integer      keyword action...;
[a-z]+      identifier action...;
```

are given in that order, and `integer` is the input word, **lex** matches the input as an identifier, because `[a-z]+` matches eight characters while `integer` matches only seven. However, if the input is `integer`, both rules match seven characters. **lex** selects the keyword rule because it occurs first. A shorter input, such as `int`, does not match the expression `integer`, and so **lex** selects the identifier rule.

Matching a String Using Wildcard Characters

Because **lex** chooses the longest match first, do not use rules containing expressions like `.*`. For example:

```
'.*'
```

might seem like a good way to recognize a string in single quotes. However, the lexical analyzer reads far ahead, looking for a distant single quote to complete the long match. If a lexical analyzer with such a rule gets the following input:

```
'first' quoted string here, 'second' here
```

it matches:

```
'first' quoted string here, 'second'
```

To find the smaller strings, `first` and `second`, use the following rule:

```
'[^\\n]*'
```

This rule stops after `'first'`.

Errors of this type are not far reaching, because the `.` (period) operator does not match a new-line character. Therefore, expressions like `.*` (period asterisk) stop on the current line. Do not try to defeat this with expressions like `[.\\n]+`. The lexical analyzer tries to read the entire input file and an internal buffer overflow occurs.

Finding Strings within Strings

The **lex** program partitions the input stream and does not search for all possible matches of each expression. Each character is accounted for once and only once. For example, to count occurrences of both `she` and `he` in an input text, try the following rules:

```
she      s++
he       h++
\\n      |.
```

where the last two rules ignore everything besides `he` and `she`. However, because `she` includes `he`, **lex** does *not* recognize the instances of `he` that are included in `she`.

To override this choice, use the action **REJECT**. This directive tells **lex** to go to the next rule. **lex** then adjusts the position of the input pointer to where it was before the first rule was executed and executes the second choice rule. For example, to count the included instances of `he`, use the following rules:

```
she      {s++;REJECT;}
he       {h++;REJECT;}
\\n      |.
```

After counting the occurrences of `she`, **lex** rejects the input stream and then counts the occurrences of `he`. Because in this case `she` includes `he` but not vice versa, you can omit the **REJECT** action on `he`. In other cases, it may be difficult to determine which input characters are in both classes.

In general, **REJECT** is useful whenever the purpose of **lex** is not to partition the input stream but to detect all examples of some items in the input, and the instances of these items may overlap or include each other.

Flags

Item Description

- C** Produces the **lex.yy.C** file instead of **lex.yy.c** for use with a C++ compiler. To get the I/O Stream Library, use the macro, **_CPP_IOSTREAMS**, as well.
- n** Suppresses the statistics summary. When you set your own table sizes for the finite state machine, the lex command automatically produces this summary if you do not select this flag.
- t** Writes **lex.yy.c** to standard output instead of to a file.
- v** Provides a one-line summary of the generated finite-state-machine statistics.

Exit Status

This command returns the following exit values:

Item Description

- 0** Successful completion.
- >0** An error occurred.

Examples

1. To draw **lex** instructions from the file `lexcommands` and place the output in **lex.yy.c**, use the following command:

```
lex lexcommands
```

2. To create a **lex** program that converts uppercase to lowercase, removes blanks at the end of a line, and replaces multiple blanks by single blanks, including the following in a **lex** command file:

```
%%  
[A-Z] putchar(yytext[0]+ 'a' - 'A');  
[ ]+$ ;  
[ ]+ putchar(' ');
```

Files

Item	Description
<code>/usr/ccs/lib/libl.a</code>	Contains the run-time library.
<code>/usr/ccs/lib/lex/ncform</code>	Defines a finite state machine.

line Command

Purpose

Reads one line from the standard input.

Syntax

`line`

Description

The **line** command copies one line from standard input and writes it to standard output. It returns an exit value of 1 on an end-of-file and always writes at least a new-line character. Use this command within a shell command file to read from the workstation.

Exit Status

This command returns the following exit values:

Item	Description
------	-------------

m

0 Successful completion.

>0 End-of-file occurred on input.

Examples

To read a line from the keyboard and append it to a file, create a script file as follows:

```
echo 'Enter comments for the log:'
echo ': \c'
line >>log
```

This shell procedure displays the message:

```
Enter comments for the log:
```

and then reads a line of text from the workstation keyboard and adds it to the end of the log. The echo ': \c' command displays a colon prompt.

link Command

Purpose

Performs a **link** subroutine.

Syntax

link *File1 File2*

Description

The **link** command performs the **link** subroutine on a specified file. The **link** 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 **link** 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.

Note: If the `.` (dot) entry has been destroyed and the **fsck** command is unable to repair it (a rare occurrence), you can use the **link** command to restore the `.` (dot) entry of the damaged directory. Use the **link** *Dir Dir/*. command where the *Dir* parameter is the name of the damaged directory. However, use this only as a last resort when the directory is destroyed and the **fsck** command is unable to fix it.

Although the linked files and directories can be removed by the **unlink** command, it's safer to use the **rm** or **rmdir** command.

Examples

To create an additional link for an existing `file1`, enter:

```
link file1 file2
```

Files

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

lint Command

Purpose

Checks C and C++ language programs for potential problems.

Syntax

```
lint [ -a ] [ -b ] [ -c ] [ -C ] [ -h ] [ -lKey ] [ -n ] [ -oLibrary ] [ -qDBCS ] [ -p ] [ -t ] [ -u ] [ -v ] [ -w Class ] [ Class ... ] [ -x ] [ -MA ] [ -NdNumber ] [ -NlNumber ] [ -NnNumber ] [ -NtNumber ] [ -IDirectory ] [ -DName ] [ =Definition ] [ -UName ] File ...
```

Description

The **lint** command checks C and C++ language source code for coding and syntax errors and for inefficient or non-portable code. You can use this program to:

- Identify source code and library incompatibility.
- Enforce type-checking rules more strictly than does the compiler.
- Identify potential problems with variables.
- Identify potential problems with functions.
- Identify problems with flow control.
- Identify legal constructions that may produce errors or be inefficient.
- Identify unused variable and function declarations.
- Identify possibly non-portable code.

Note: Checking of C++ language files by the **lint** command requires the presence of the C Set++ Compiler package.

The inter-file usage of functions is checked to find functions that return values in some instances and not in others, functions called with varying numbers or types of arguments, and functions whose values are not used or whose values are used but not returned.

The **lint** command interprets file name extensions as follows:

- *File* names ending in `.c` are C language source files.

- File names ending in **.C** are C++ language source files.
- File names ending in **.ln** are non-ASCII files that the **lint** command produces when either the **-c** or the **-o** flag is used.

The **lint** command warns you about files with other suffixes and ignores them.

The **lint** command takes all the **.c**, **.C**, and **.ln** files and the libraries specified by **-l** flags and processes them in the order that they appear on the command line. By default, it adds the standard **llib-*lc*.ln** lint library to the end of the list of files. However, when you select the **-p** flag, the **lint** command uses the **llib-*port*.ln** portable library. By default, the second pass of **lint** checks this list of files for mutual compatibility; however, if you specify the **-c** flag, the **.ln** and **llib-*lx*.ln** files are ignored.

The **-c** and **-o** flags allow for incremental use of the **lint** command on a set of C and C++ language source files. Generally, use the **lint** command once for each source file with the **-c** flag. Each of these runs produces a **.ln** file that corresponds to the **.c** file and writes all messages concerning that source file. After you have run all source files separately through the **lint** command, run it once more, without the **-c** flag, listing all the **.ln** files with the needed **-l** flags. This writes all inter-file inconsistencies. This procedure works well with the **make** command, allowing it to run the **lint** command on only those source files modified since the last time that set of source files was checked.

The **lint** and **LINT** preprocessor symbols are defined to allow certain questionable code to be altered or removed for the **lint** command. Therefore, the **lint** and **LINT** symbols should be thought of as a reserved word for all code that is planned to be checked by **lint**.

The following comments in a C and C++ language source program change the way the **lint** command operates when checking the source program:

Item	Description
/*NOTREACHED*/	Suppresses comments about unreachable code.
/*VARARGS<i>Number</i>*/	Suppresses checking the following old style function declaration for varying numbers of arguments, but does check the data type of the first <i>Number</i> arguments. If you do not include a value for <i>Number</i> , the lint command checks no arguments (<i>Number</i> =0). The ANSI function prototypes should use the ellipsis to indicate unspecified parameters rather than this comment mechanism.
/*ARGSUSED*/	Suppresses warnings about function parameters not used within the function definition.
/*LINTLIBRARY*/	If you place this comment at the beginning of a file, the lint command does not identify unused functions and function parameters in the file. This is used when running the lint command on libraries.
/*NOTUSED*/	Suppresses warnings about unused external symbols, functions and function parameters in the file beginning at its point of occurrence. This is a superset of the /*LINTLIBRARY*/ comment directive, but applies also to external symbols. It is useful for suppressing warnings about unused function prototypes and other external object declarations.
/*NOTDEFINED*/	Suppresses warnings about used, but undefined external symbols and functions in the file beginning at its point of occurrence.
/*LINTSTDLIB*/	Permits a standard prototype-checking library to be formed from header files by making function prototype declarations appear as function definitions. This directive implicitly activates both the /*NOTUSED*/ and /*LINTLIBRARY*/ comment directives to reduce warning noise levels.

The **lint** command warning messages give file name and line number. As each file goes through the first pass, warnings for each file and each line number are reported.

If you have not specified the **-c** flag, the **lint** command collects information gathered from all input files and checks it for consistency. At this point, if it is not clear whether a message stems from a given source file or from one of its included files, the **lint** command displays the source file name followed by a question mark.

ANSI programs that include many standard header files may wish to set the **-wD** flag to reduce the quantity of warnings about prototypes not used, and the **-n** flag to disable checking against the ANSI standard library. For non-ANSI programs, it is advisable to specify the **-wk** flag to reduce the amount of warnings concerning the absence of function prototypes.

Flags

Item	Description
-a	Suppresses messages about assignments of long values to variables that are not long.
-b	Suppresses messages about unreachable break statements.
-c	Causes the lint command to produce an .ln file for every .c file on the command line. These .ln files are the product of the first pass of the lint command only and are not checked for inter-function compatibility.
-C	Specifies to use the C++ libraries (in the /usr/lpp/xlC/lib directory).
-h	Does not try to detect bugs, improper style, or reduce waste.
-lKey	Includes the additional llib-lKey.ln lint library. You can include a lint version of the llib-lm.ln math library by specifying -lm on the command line or llib-ldos.ln library by specifying the -ldos flag on the command line. Use this flag to include local lint libraries when checking files that are part of a project having a large number of files. This flag does not prevent the lint command from using the llib-lc.ln library. The lint library must be in the /usr/ccs/lib directory.
-n	Suppresses the check for compatibility with either the standard or the portable lint libraries. This applies for both the ANSI and extended mode libraries.
-oLibrary	Causes the lint command to create the llib-lLibrary.ln lint library. The -c flag nullifies any use of the -o flag. The lint library produced is the input that is given to the second pass of the lint command. The -o flag simply causes this file to be saved in the named lint library. To produce a llib-lLibrary.ln without extraneous messages, use the -x flag. The -v flag is useful if the source files for the lint library are just external interfaces (for example, the way the llib-lc file is written). These flag settings are also available through the use of lint command comment lines.
-p	Checks for portability to other C language dialects.
-t	Checks for problematic assignments when porting from 32 to 64 bit. Only the following cases are checked: <ul style="list-style-type: none">• all shift / mask operations are flagged because some operations that work well in 32-bit may cause problems in 64-bit.• warnings are given for the following type of assignments.
	<pre>int = long int = ptr</pre>
-u	Suppresses messages about functions and external variables that are either used and not defined or defined and not used. Use this flag to run the lint command on a subset of files of a larger program.
-v	Suppresses messages about function parameters that are not used.

Item	Description
-w <i>Class</i> [<i>Class...</i>]	<p>Controls the reporting of warning classes. All warning classes are active by default, but can be individually deactivated by including the appropriate option as part of the <i>Class</i> argument. The individual options are listed as:</p> <ul style="list-style-type: none"> a Non-ANSI features. c Comparisons with unsigned values. d Declaration consistency. h Heuristic complaints. k Use for K+R type source code. l Assignment of long values to variables that are not long. n Null-effect code. o Unknown order of evaluation. p Various portability concerns. r Return statement consistency. s Storage capacity checks. u Proper usage of variables and functions. A Deactivate all warnings. C Constants occurring in conditionals. D External declarations are never used. O Obsolescent features. P Function prototype presence. R Detection of unreachable code.
-x	Suppresses messages about variables that have external declarations but are never used.
-MA	Enforces the ANSI C language standard rules. The default mode is equal to the extended C mode. The ANSI mode prepends the standard ANSI library function prototypes in place of the default extended mode C lint library. The ANSI mode enforces a stricter inter-file object reference and provides definition linkage checks.
-Nd <i>Number</i>	Changes the dimension table size to <i>Number</i> . The default value of <i>Number</i> is 2000.
-NI <i>Number</i>	Changes the number of type nodes to <i>Number</i> . The default value of <i>Number</i> is 8000.

Item	Description
-Nn <i>Number</i>	Increases the size of the symbol table to <i>Number</i> . The default value of <i>Number</i> is 1500.
-Nt <i>Number</i>	Changes the number of tree nodes to <i>Number</i> . The default value of <i>Number</i> is 1000.

In addition, the **lint** command recognizes the following flags of the **cpp** command (macro preprocessor):

Item	Description
-ID <i>Directory</i>	Adds the <i>Directory</i> to the list of directories in which the lint command searches for the #include files.
-D <i>Name</i> [= <i>Definition</i>]	Defines the <i>Name</i> , as if by the #define file. The default of the <i>Definition</i> is the value of 1.
-q <i>DBCS</i>	Sets multibyte mode specified by the current locale.
-U <i>Name</i>	Removes any initial definition of the <i>Name</i> , where the <i>Name</i> is a reserved symbol that is predefined by the particular preprocessor.

Exit Status

This command returns the following exit values:

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

Examples

1. To check a C program for errors, enter:

```
lint command.c
```

2. To suppress some of the messages, enter:

```
lint -v -x program.c
```

This checks `program.c`, but does not display error messages about unused function parameters (**-v**) or unused externals (**-x**).

3. To check the program against an additional lint library, enter:

```
lint -lsubs program.c
```

This checks `program.c` against both the `/usr/ccs/lib/llib-lc.ln` standard lint library and `/usr/lib/llib-lsubs.ln` lint library.

4. To check against the portable library and an additional library, enter:

```
lint -lsubs -p program.c
```

This checks `program.c` against both the `/usr/ccs/lib/llib-port.ln` portable lint library and `/usr/lib/llib-lsubs.ln` lint library.

5. To check against a nonstandard library only, enter:

```
lint -lsubs -n program.c
```

This checks `program.c` against only `/usr/lib/llib-lsubs.ln`.

Files

Item	Description
<code>/usr/lib/lint{12}</code>	Programs
<code>/usr/ccs/lib/l-lib-lansi</code>	Declarations for standard ANSI functions (source)
<code>/usr/ccs/lib/l-lib-lansi.ln</code>	Declarations for standard ANSI functions (binary format)
<code>/usr/ccs/lib/l-lib-lc</code>	Declarations for standard functions (source)
<code>/usr/ccs/lib/l-lib-lc.ln</code>	Declarations for standard functions (binary format)
<code>/usr/ccs/lib/l-lib-lcrses</code>	Declarations for curses functions (source)
<code>/usr/ccs/lib/l-lib-lcrses.ln</code>	Declarations for curses functions (binary format)
<code>/usr/ccs/lib/l-lib-lm</code>	Declarations for standard math functions (source)
<code>/usr/ccs/lib/l-lib-lm.ln</code>	Declarations for standard math functions (binary format)
<code>/usr/ccs/lib/l-lib-port</code>	Declarations for portable functions (source)
<code>/usr/ccs/lib/l-lib-port.ln</code>	Declarations for portable functions (binary format)
<code>/usr/lpp//xlc/lib</code>	Directory containing C++ libraries
<code>/var/tmp/*lint*</code>	Temporary files

listdgrp Command

Purpose

Displays devices of a device class.

Syntax

`listdgrp` *DeviceClass*

Description

Lists information about devices where the *DeviceClass* parameter refers to a object class of Customized Devices in the Device Configuration database.

Parameters

Item	Description
<i>DeviceClass</i>	Specifies the device class whose members will be displayed.

Exit Status

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

Examples

- To list the devices in the **adapter** class, enter:

```
listdgrp adapter
```


The output looks similar to the following:

```
a0
sa1
siokma0
fda0
scsi0
scsi1
bl0
sioka0
siota0
```

Files

Item	Description
<code>/usr/bin/listdgrp</code>	Contains the System V <code>listdgrp</code> command.

listvgbackup Command

Purpose

Lists or restores the contents of a volume group backup on a specified media.

Syntax

```
listvgbackup [ -b blocks ] [ -f device ] [ -a ] [ -c ] [ -l ] [ -n ] [ -r ] [ -s ] [ -d path ] [ -B ] [ -D ] [ -L ] [ -V ] [ file_list ]
```

Description

The `listvgbackup` command lists the contents of a volume group backup from tape, file, CD-ROM, or other source and can be used to restore files from a valid backup source. The `listvgbackup` command also works for multi-volume backups such as multiple CDs, DVDs, USB disks, or tapes.

The `listvgbackup -r` and `restorevgfiles` commands perform identical operations and should be considered interchangeable.

Flags

Item	Description
-a	Verifies the physical block size of the tape backup, as specified by the -b <i>block</i> 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.
-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.
-B	Prints the volume group backup log to stdout . This flag will display the past 256 backups (roughly). The log is in alog format and is kept in <code>/var/adm/ras/vgbackuplog</code> . Each line of the log is a semicolon-separated list of the file or device name, the command used to make backup, date, shrink size, full size of the backup, and recommended maintenance or technology level (if any). Note: The shrink size is the size of the data on all filesystems. The full size is total size of each filesystem (unused + data).
-c	Produces colon-separated output. This flag only works with the -l and -L flags.

Item	Description
-d <i>path</i>	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.
-l	Displays useful information about a volume group backup. This flag requires the -f device flag. This flag causes listvgbackup to display information such as volume group, date and time backup was made, uname output from backed up system, oslevel, recommended maintenance or technology level, 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 " lsvg -l vgroupname ".
-L	Displays lpp fileset information about a mksysb backup only. This flag requires the -f device flag and displays the equivalent information to that produced by invoking " lspp -l " on the running backed up system. This flag does not produce output about any volume group backup other than that produced by mksysb .
-f device	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 .
-n	Does not restore ACLs, PCLs, or extended attributes.
-r	Specifies to restore the backup files, as defined by the <i>file-list</i> parameter. If the <i>file-list</i> parameter is not specified, then all files in the backup will be restored. If the -r flag is not used, then executing the listvgbackup command only lists the files in the specified backup.
-s	Specifies that the backup source is a user volume group and not rootvg.
-V	Verifies a tape backup. This flag requires the -f device flag and works for tape devices only. The -V flag causes listvgbackup to verify the readability of the header of each file on the volume group backup and print any errors that occur to stderr .

Parameters

Item	Description
<i>file_list</i>	Identifies the list of files to be restored. This parameter is used only when the -r flag is specified. 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 list the contents of the system backup located on the default device **/dev/rmt0**, enter:

```
listvgbackup
```

2. To list the contents of the system backup located on device **/dev/cd1**, enter:

```
listvgbackup -f /dev/cd1
```

3. To list the contents of the system backup located on device **/dev/cd1**, which is a user volume group that is not rootvg, enter:

```
listvgbackup -f /dev/cd1 -s
```

4. To restore **/etc/filesystems** from the system backup located on device **/dev/cd1**, enter:

```
listvgbackup -f /dev/cd1 -r ./etc/filesystems
```

5. To restore all files in the **/myfs/test** directory of the non-rootvg backup, which is located on device **/dev/cd1**, and write the restored files to **/data/myfiles**, enter:

```
listvgbackup -f /dev/cd1 -r -s -d /data/myfiles ./myfs/test
```

6. To display colon separated lpp information about a **mksysb** backup tape located on **/dev/rmt0**, enter the following:

```
lsmksysb -Lc -f /dev/rmt0
```

7. To display the volume group backup log to **stdout**, enter:

```
lssavevg -B
```

8. To list volume group and general backup data about a backup located at **/tmp/mybackup**, enter:

```
listvgbackup -l -f /tmp/mybackup
```

9. To verify the readability of each header on a volume group backup tape in **/dev/rmt0**, enter:

```
lsmksysb -V -f /dev/rmt0
```

10. To list the contents of the system backup located on device **/dev/usbms0**, use the following command:

```
listvgbackup -f /dev/usbms0
```

Files

Item	Description
/usr/bin/listvgbackup	Contains the listvgbackup command

listX11input Command

Purpose

Lists X11 input extension records entered into the Object Data Manager (ODM) database.

Syntax

listX11input

Description

The **listX11input** command lists all X11 input extension records entered in the ODM database.

Error Codes

Item	Description
ODM could not open class	The ODM database is not stored in the <code>/usr/lib/objrepos</code> directory.

livedumpstart Command

Purpose

Initiates a live dump.

Syntax

```
livedumpstart [ -e ] [ -h ] [ -p pseudo-component ] [ -q ] [ -r ] [ -u ] [ -c component_path ] [ -l logical_alias ] [ -t type ] [ -C component_path | -L logical_alias | -T type ] attribute [ ... ]
```

Description

The `livedumpstart` command is used to start a live dump. The dump can include one or more components. Only serialized dumps are used. It can be limited to one pass. The data acquired is dumped to the file system, and the dump is placed in a directory. The dump can be designated as informational or critical.

Components are dumped in the order that you specify. Specify the failing component with either the `-C`, `-L`, or `-T` flag. You cannot specify the name of a pseudo-component.

The data is dumped at the detail level that you set for that component, see the `dumpctrl` command for more information about managing system and live dumps.

If you do not specify the `-q` flag, the `livedumpstart` command displays a message containing the name of the dump.

Flags

Item	Description
<code>-c [+]</code> <i>component_path</i> [+] [: <i>parameter_list</i>]	Specifies a component by component path name. You can specify the <code>-c</code> flag more than once. If you precede a component name with a plus sign (+), the data from that component and its ancestors are dumped. If you follow a component name with a plus sign (+), the data from that component and its descendants are dumped. You can pass parameters to the component. Follow the component name and the optional "+": <i>parameter_list</i> . A <i>parameter_list</i> consists of parameters separated by commas. It can also be groups separated by blanks. If a component and its ancestors or descendants are specified, parameters are passed only to the component, not to the ancestors or descendants.
<code>-C [+]</code> <i>component_path</i> [+] [: <i>parameter_list</i>]	Specifies a failing component by component path name. At most one failing component can be specified. Thus, only one of the <code>-C</code> , <code>-L</code> , and <code>-T</code> flags is allowed, and that component specification must refer to a single component. If <code>-C basecomp+</code> is specified, and <code>basecomp</code> is not live dump aware, then only one component among <code>basecomp</code> and its descendants can be live dump aware. However, if <code>basecomp</code> is live dump aware, <code>basecomp</code> is the failing component, and it might have multiple live-dump-aware descendants. Tip: These rules also apply to a component and its ancestors. If a component is preceded with a plus sign, "+", then that component and its ancestors are dumped. If a component is followed with a plus sign, "+", then that component and its descendants are dumped. If parameters are passed to the component, the component and the optional "+" are followed with <i>parameter_list</i> . A <i>parameter_list</i> consists of parameters separated by commas, or <code>keyword=parm_list</code> pairs separated by blanks. See the section on specifying parameters from the command line for more information. Note that if a component and its ancestors and/or descendants are specified, parameters are passed only to the component, not to the ancestors or descendants.
<code>-e</code>	Displays an estimate for the size of the dump, which contains the specified components or pseudo-components. This flag obtains a size estimate for the dump without starting the dump. To get an accurate estimate, use the same components, parameters, and detail level that you intend to use for the dump. The estimate takes into account the compression factor.
<code>-h</code>	Shows help text. If the <code>-h</code> flag is specified with other components or pseudo-components, the help text for those components is shown.
<code>-l [+]</code> <i>logical_alias</i> [+] [: <i>parameter_list</i>]	Specifies a component by component logical alias. You can specify multiple <code>-c</code> , <code>-L</code> , and <code>-t</code> flags. If a logical alias is preceded with a plus sign "+", then that alias and its ancestors are dumped. If a logical alias is followed with a plus sign "+", then that alias and its descendants are dumped. If parameters are passed to the component, the component and the optional "+" are followed with <i>parameter_list</i> . A <i>parameter_list</i> consists of parameters separated by commas, or <code>keyword=parm_list</code> pairs separated by blanks. See the section on specifying parameters from the command line for more information. Note that if a component and its ancestors, descendants, or both are specified, parameters are passed only to the component, not to the ancestors or descendants.

Item	Description																																										
-L <i>[+]</i> <i>logical_alias[+]</i> <i>[:parameter_list]</i>	<p>Specifies a failing component by component logical alias. At most one failing component can be specified. Thus, only one of the -C, -L, and -T flags is allowed, and that component specification must refer to a single component. If -L basecomp+ is specified, and basecomp is not live dump aware, then only one component among basecomp and its descendants can be live dump aware.</p> <p>However, if basecomp is live dump aware, basecomp is the failing component, and it might have multiple live-dump-aware descendants.</p> <p>Tip: These rules also apply to a component and its ancestors.</p> <p>If a logical alias is preceded with a plus sign "+", then that alias and its ancestors are dumped. If a logical alias is followed with a plus sign "+", then that alias and its descendants are dumped.</p> <p>If parameters are passed to the component, the component and the optional "+" are followed with <i>:parameter_list</i>. A <i>parameter_list</i> consists of parameters separated by commas, or keyword=parm_list pairs separated by blanks. See the section on specifying parameters from the command line for more information. Note that if a component and its ancestors and/or descendants are specified, parameters are passed only to the component, not to the ancestors or descendants.</p>																																										
-p <i>pseudo-component</i> <i>[:parameter_list]</i>	<p>Specifies a pseudo-component.</p> <p>Note: A pseudo-component (-p) cannot be a failing component.</p> <p>If parameters are to be passed to the pseudo-component, the pseudo-component must be followed by a <i>:parameter_list</i>. A <i>parameter_list</i> consists of parameters separated by commas, or keyword=parm_list pairs separated by blanks. See the section on specifying parameters from the command line for more information.</p> <p>The following table is the description of pseudo-components.</p> <table border="1"> <thead> <tr> <th>Specification</th> <th>Parameters</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>eaddr:hex,hex</td> <td>address and length, hexadecimal values</td> <td>Dumps memory by kernel effective address.</td> </tr> <tr> <td>context:addr=hex-eaddr tid_t=hex-tid_t cpu=dec-lcpu bid=dec-bid</td> <td>hex-eaddr - context (MST) effective address, hex-tid_t - kernel thread id, dec-lcpu - logical cpu, dec-bid - cpu bind id</td> <td>Dumps a kernel context. This includes lightweight memory trace data, stack and thread state information.</td> </tr> <tr> <td>tid_t:hex-tid</td> <td>hexadecimal kernel thread id</td> <td>Dumps a kernel thread by kernel thread ID.</td> </tr> <tr> <td>tid:dec-tid</td> <td>decimal kernel thread id</td> <td>Dumps a kernel thread by kernel thread ID, and the ID is decimal.</td> </tr> <tr> <td>tslot:dec-slot</td> <td>decimal kernel thread slot number</td> <td>Dumps a kernel thread by kernel thread ID, and the thread is specified by decimal slot number.</td> </tr> <tr> <td>pid_t:hex-pid</td> <td>hexadecimal kernel process id</td> <td>Dumps a process by process ID.</td> </tr> <tr> <td>pid:dec-pid</td> <td>decimal process id</td> <td>Dumps a process by process ID, and the ID is decimal.</td> </tr> <tr> <td>pslot:dec-slot</td> <td>decimal process slot number</td> <td>Dumps a process by process ID, and the process is specified by decimal slot number.</td> </tr> <tr> <td>errbuf</td> <td>no parameters</td> <td>Dumps kernel error logging data.</td> </tr> <tr> <td>mtrc:common-size, rare-size</td> <td>common and rare decimal buffer sizes</td> <td>Dumps lightweight memory trace data.</td> </tr> <tr> <td>systrace:dec-size</td> <td>decimal buffer size</td> <td>Dump system trace data. If the buffer size is 0, the entire buffer is dumped.</td> </tr> <tr> <td>comptrace:component, dec-length</td> <td>component name and decimal amount of data. The component can be an alias, and the length can be zero to dump the entire buffer.</td> <td>Dumps component trace data.</td> </tr> <tr> <td>kernext:pathname</td> <td>extension's full path name</td> <td>Allows symbol resolution for this extension.</td> </tr> </tbody> </table>	Specification	Parameters	Description	eaddr:hex,hex	address and length, hexadecimal values	Dumps memory by kernel effective address.	context:addr=hex-eaddr tid_t=hex-tid_t cpu=dec-lcpu bid=dec-bid	hex-eaddr - context (MST) effective address, hex-tid_t - kernel thread id, dec-lcpu - logical cpu, dec-bid - cpu bind id	Dumps a kernel context. This includes lightweight memory trace data, stack and thread state information.	tid_t:hex-tid	hexadecimal kernel thread id	Dumps a kernel thread by kernel thread ID.	tid:dec-tid	decimal kernel thread id	Dumps a kernel thread by kernel thread ID, and the ID is decimal.	tslot:dec-slot	decimal kernel thread slot number	Dumps a kernel thread by kernel thread ID, and the thread is specified by decimal slot number.	pid_t:hex-pid	hexadecimal kernel process id	Dumps a process by process ID.	pid:dec-pid	decimal process id	Dumps a process by process ID, and the ID is decimal.	pslot:dec-slot	decimal process slot number	Dumps a process by process ID, and the process is specified by decimal slot number.	errbuf	no parameters	Dumps kernel error logging data.	mtrc:common-size, rare-size	common and rare decimal buffer sizes	Dumps lightweight memory trace data.	systrace:dec-size	decimal buffer size	Dump system trace data. If the buffer size is 0, the entire buffer is dumped.	comptrace:component, dec-length	component name and decimal amount of data. The component can be an alias, and the length can be zero to dump the entire buffer.	Dumps component trace data.	kernext:pathname	extension's full path name	Allows symbol resolution for this extension.
Specification	Parameters	Description																																									
eaddr:hex,hex	address and length, hexadecimal values	Dumps memory by kernel effective address.																																									
context:addr=hex-eaddr tid_t=hex-tid_t cpu=dec-lcpu bid=dec-bid	hex-eaddr - context (MST) effective address, hex-tid_t - kernel thread id, dec-lcpu - logical cpu, dec-bid - cpu bind id	Dumps a kernel context. This includes lightweight memory trace data, stack and thread state information.																																									
tid_t:hex-tid	hexadecimal kernel thread id	Dumps a kernel thread by kernel thread ID.																																									
tid:dec-tid	decimal kernel thread id	Dumps a kernel thread by kernel thread ID, and the ID is decimal.																																									
tslot:dec-slot	decimal kernel thread slot number	Dumps a kernel thread by kernel thread ID, and the thread is specified by decimal slot number.																																									
pid_t:hex-pid	hexadecimal kernel process id	Dumps a process by process ID.																																									
pid:dec-pid	decimal process id	Dumps a process by process ID, and the ID is decimal.																																									
pslot:dec-slot	decimal process slot number	Dumps a process by process ID, and the process is specified by decimal slot number.																																									
errbuf	no parameters	Dumps kernel error logging data.																																									
mtrc:common-size, rare-size	common and rare decimal buffer sizes	Dumps lightweight memory trace data.																																									
systrace:dec-size	decimal buffer size	Dump system trace data. If the buffer size is 0, the entire buffer is dumped.																																									
comptrace:component, dec-length	component name and decimal amount of data. The component can be an alias, and the length can be zero to dump the entire buffer.	Dumps component trace data.																																									
kernext:pathname	extension's full path name	Allows symbol resolution for this extension.																																									
-q	Specifies quiet mode. No messages are displayed.																																										
-r	Dumps data for any subcomponents of the specified components. Specifying this flag is equivalent to specifying every component followed by a "+".																																										
-t <i>[+]</i> <i>type[+]</i> <i>[:parameter_list]</i>	<p>Specifies a component by its type or subtype. You can specify multiple -c, -l, and -t flags.</p> <p>If a type or subtype is preceded with a plus sign (+), then that component and its ancestors are dumped. If a type or subtype is followed with a plus sign (+), then that component and its descendants are dumped.</p> <p>If parameters are passed to the component, the component and the optional "+" are followed with <i>:parameter_list</i>. A <i>parameter_list</i> consists of parameters separated by commas, or keyword=parm_list pairs separated by blanks. See the section on specifying parameters from the command line for more information. Note that if a component and its ancestors and/or descendants are specified, parameters are passed only to the component, not to the ancestors or descendants.</p>																																										
-T <i>[+]</i> <i>type[+]</i> <i>[:parameter_list]</i>	<p>Specifies a failing component by component type or subtype. At most one failing component can be specified. Thus, only one of the -C, -L, and -T flags is allowed, and that component specification must refer to a single component. If -T type+ is specified, and <i>type</i> is not live dump aware, then only one component among <i>type</i> and its descendants can be live dump aware.</p> <p>However, if a component of the type <i>type</i> is live dump aware, it is the failing component, and it might have multiple live-dump-aware descendants.</p> <p>Tip: These rules also apply to a component and its ancestors.</p> <p>If a type or subtype is preceded with a plus sign (+), then that component and its ancestors are dumped. If a type is followed with a plus sign (+), then that component and its descendants are dumped.</p> <p>If parameters are passed to the component, the component and the optional "+" are followed with <i>:parameter_list</i>. A <i>parameter_list</i> consists of parameters separated by commas, or keyword=parm_list pairs separated by blanks. See the section on specifying parameters from the command line for more information. Note that if a component and its ancestors, descendants, or both are specified, parameters are passed only to the component, not to the ancestors or descendants.</p>																																										
-u	Dumps the data for the components that are "above" the specified components in the component hierarchy. This is equivalent to specifying every component preceded by a "+".																																										

You can use wildcard when you specify component names and aliases. Remember that any parameters that you specify are passed to all matching components. You cannot use **all** or an asterisk (*).

Restriction: You can only specify one failing component, so **-C comp*** can resolve to only one component.

Attributes

The dump attributes are specified with keyword=value pairs. They are used to configure dump parameters, construct dump headers and edit symptom information. You change attributes by specifying an *Attribute=Value* parameter. If you have the proper authority you can set the following required attribute:

Item	Description
symptom=string	Provides symptom string details that must be supplied to further qualify the dump. The maximum length of this string is 2047 characters.

If you have the proper authority you can set the following optional group attributes:

Item	Description
errcode=code	Specifies the error code for the symptom string. If it begins with 0x, the value is in hex; if it begins with 0, the value is octal; otherwise it is decimal.
force=yes no	If yes, overrides duplicate checking, dumps the data regardless of whether it duplicates a previous dump. The default is yes , because any dump taken from the command line should not be treated as a duplicate.
log=yes no	Specifies whether a log entry should be written when the dump completes. If you specify yes , a message is written to the error log. The default is yes .
noforce	For software initiated live dumps. Specifies whether to initiate the dump if it duplicates a previous dump that was initiated within the last day. The noforce attribute makes the dump subject to duplicate elimination.
nolog	Specifies whether a message should be written to the error log when the dump is complete. If it is not specified, dump completion and errors are logged.
prefix=prefix	Specifies the file name prefix. The name can be no more than 63 characters.
priority=priority	Specifies the priority of the dump. You can specify info or critical . The default is critical . If you specify the value info , it indicates the dump is for informational purposes, while critical indicates the dump is necessary to debug a problem.
title=string	Specifies an optional dump title, which can be up to 127 characters.
type=type	Specifies whether data should be collected without freezing the system. <ul style="list-style-type: none"> serialized ser The dump data is gathered when the system is frozen. It might be necessary to use multiple freezes to dump all the data. This is the default. unserialized unser Data is gathered without freezing the system. It might be necessary to use multiple freezes to dump all the data. If you specify unserialized, the system is not frozen when gathering the data.
onepass	All data is gathered under one pass. The dump is truncated if all data does not fit in available memory. The default is to use multiple passes if required.

Exit Status

Item	Description
0 (zero)	The <code>livedumpstart</code> command completes successfully and produces a message containing the name of the dump.

Item	Description
nonzero	<p>The <code>livedumpstart</code> command fails and produces an error message. This command fails under the following conditions:</p> <ul style="list-style-type: none"> • One or more parameters are not valid. • One or more components are not valid. • None of these components can be specified for a live dump. • A component attempts to take a live dump from within a live dump. • Live dumps are disabled. • A dump already exists. This can occur when you specify the force=no attribute. • There is insufficient memory. • All the data cannot be buffered in this single-pass dump. • Too much time is spent while processors are disabled, and this dump is truncated.

Security

Only the root user can run this command.

Examples

1. To dump data for device `ent0`, and components above it in the component hierarchy, enter the following command:

```
livedumpstart -L +ent0 symptom=foo
```

The failing component is `ent0`. This creates a dump named `ent0.yymmddhhmm.00.DZ`. It is a serialized, critical dump.

Tip: According to the rules for specifying the failing component, if `ent0` is not live dump aware, but multiple ancestors are, then this command fails. If `ent0` is not live dump aware, and only one ancestor is, this ancestor is used as the failing component.

2. To create an informational dump of the process management data for processes 856 and 10272, enter the following command:

```
livedumpstart -p pid:856 -p pid:10272 \
  info prefix=mydump title="process dump" symptom="foo"
```

The dump is named `mydump.nocomp.yymmddhhmm.00.DZ`. Note there is no failing component.

3. To create a serialized, one-pass dump where `foo` is the failing component, enter the following command:

```
livedumpstart -C foo+:block=45ab8 -pcontext:tid_t=57B29 onepass symptom=bar
```

This command dumps `foo`, its descendents, and the context for kernel thread 57B29. The dump is named `foo.yymmddhhmm.00.DZ`.

4. A subsystem has the parent component with alias `subsys`. It has only one live-dump-aware component. To create a serialized live dump of this subsystem, you might use the following command:

```
livedumpstart -L subsyst+ title="Dump of subsystem subsyst" symptom=foo
```

- To specify that process 1234 is dumped along with 0x400 bytes starting at 0x45928, enter the following command:

```
livedumpstart -p tid:1234 -p eaddr:45928,400 symptom=foo
```

In this example, there is no failing component.

lkdev Command

Purpose

Locks a device. Any attempt to modify device characteristics fails.

Syntax

```
lkdev [ -l Name -a | -d [ -c Text ] ]
```

```
lkdev -h
```

Description

The **lkdev** command locks the specified device (the **-l** *Name* flag). Any attempt to modify device attributes by using the **chdev** or **chpath** command is denied. In addition, an attempt to delete the specified device or one of its paths from the Object Data Manager (ODM) by using either the **rmdev** or **rmpath** command is denied.

Flags

Item	Description
-h	Displays the command usage message.
-l <i>Name</i>	Specifies the logical device name of the target device for the paths affected by the change. This flag is required in all cases.
-a	Locks the specified device.
-d	Unlocks the specified device.
-c <i>Text</i>	Specifies a text string of up to 64 printable characters that contain no embedded spaces.

Security

Privilege control: Only the root user can execute this command.

Auditing events:

Event	Information
DEV_LOCK	The device command line

Examples

- To enable the lock for the *hdisk1* disk device, enter the following command:

```
lkdev -l hdisk1 -a
```

- To disable the lock for the *hdisk1* disk device, enter the following command:

```
lkdev -l hdisk1 -d
```


3. To enable the lock for the *hdisk1* disk device and create a text label, enter the following command:

```
lkdev -l hdisk1 -a -c test_string
```

4. To modify the text label for the *hdisk1* disk device, enter the following command:

```
lkdev -l hdisk1 -c new_test_string
```

Location

Item	Description
<code>/usr/sbin/lkdev</code>	Contains the lkdev command

In Command

Purpose

Links files.

Syntax

To Link a File to a File

```
ln [ -f | -n ] [ -s ] [ -P | -L ] SourceFile [ TargetFile ]
```

To Link a File or Files to a Directory

```
ln [ -f | -n ] [ -s ] [ -P | -L ] SourceFile ... TargetDirectory
```

Description

The **ln** command links the file designated in the *SourceFile* parameter to the file designated by the *TargetFile* parameter or to the same file name in another directory specified by the *TargetDirectory* parameter. By default, the **ln** command creates hard links. To use the **ln** command to create symbolic links, designate the **-s** flag.

A symbolic link is an indirect pointer to a file; its directory entry contains the name of the file to which it is linked. Symbolic links may span file systems and may refer to directories.

If you are linking a file to a new name, you can list only one file. If you are linking to a directory, you can list more than one file.

The *TargetFile* parameter is optional. If you do not designate a target file, the **ln** command creates a new file in your current directory. The new file inherits the name of the file designated in the *SourceFile* parameter. See example 5.

If you specify the **-P** flag and if the source file is a symbolic link, the **ln** command operates similar to the **linkat()** subroutine, where the *SourceFile* parameter is *Path1* argument and the target destination path is the *Path2* argument of the **linkat()** subroutine. Both *DirFileDescriptor1* and *DirFileDescriptor2* parameter values are set as **AT_FDCWD**, and the Flag parameter of the **linkat()** subroutine is set as 0.

If you specify the **-L** flag and if the source file is a symbolic link, the **ln** command operates similar to the **linkat()** subroutine, where the *SourceFile* parameter is *Path1* argument and the target destination path is the *Path2* argument of the **linkat()** subroutine. Both *DirFileDescriptor1* and *DirFileDescriptor2* parameters values are set as **AT_FDCWD**, and the Flag parameter of the **linkat()** subroutine is set as **AT_SYMLINK_FOLLOW**.

If you specify the **-s** flag to create an alias to alter the default behavior when you create hard links (for example, alias `ln='ln -L'`), the **-L** and **-P** flags are ignored because the *SourceFile* parameter value acts as a string that is used as the content for the created symbolic link and the source file is not required to exist as a file.

Notes:

1. You cannot link files across file systems without using the **-s** flag.
2. If *TargetDirectory* is already a symbolic link to a directory, then the **ln** command treats the existing target as a file. This means that a command such as **ln -fs somepath/lname symdir** will not follow the existing symbolic link of **symdir**, instead it will create a new symbolic link from **somepath/lname** to **symdir**.

Flags

Ite	Description
-----	-------------

- | | |
|-----------|---|
| -f | Causes the ln command to replace any destination paths that already exist. If a destination path already exists and the -f flag is not specified, the ln command writes a diagnostic message to standard error without creating a new link and continues to link the remaining <i>SourceFiles</i> . |
| -L | Creates a hard link to the file that is referenced by the symbolic link for each <i>SourceFile</i> parameter that names a file of symbolic link type. |
| -n | Specifies that if the link is an existing file, do not overwrite the contents of the file. The -f flag overrides this flag. This is the default behavior. |
| -P | Creates a hard link to the symbolic link for each <i>SourceFile</i> parameter that names a file of symbolic link type. |
| -s | Causes the ln command to create symbolic links. A symbolic link contains the name of the file to which it is linked. The referenced file is used when an open operation is performed on the link. A stat call on a symbolic link returns the linked-to file; an lstat call must be done to obtain information about the link. The readlink call may be used to read the contents of a symbolic link. Symbolic links can span file systems and refer to directories. |

Note: Absolute path names must be used when specifying the *SourceFile* parameter for the **-s** flag. If the absolute path name is not designated, unexpected results may occur when the *SourceFile* and the *TargetFile* parameters are located in different directories. The source file does not need to exist before creating the symbolic link.

Exit Status

This command returns the following exit values:

Ite	Description
-----	-------------

- | | |
|--------------|---|
| 0 | All specified files were successfully linked. |
| >0 | An error occurred. |

Examples

1. To create another link (alias) to a file, enter:

```
ln -f chap1 intro
```

This links `chap1` to the new name, `intro`. If `intro` does not already exist, the file name is created. If `intro` does exist, the file is replaced by a link to `chap1`. Then both the `chap1` and `intro` file names will refer to the same file. Any changes made to one also appear in the other. If one file name is deleted with the **rm** command, the file is not completely deleted since it remains under the other name.

2. To link a file to the same name in another directory, enter:

```
ln index manual
```

This links `index` to the new name, `manual/index`.

Note: `intro` in example 1 is the name of a file; `manual` in example 2 is a directory that already exists.

3. To link several files to names in another directory, enter:

```
ln chap2 jim/chap3 /home/manual
```

This links `chap2` to the new name `/home/manual/chap2` and `jim/chap3` to `/home/manual/chap3`.

4. To use the `ln` command with pattern-matching characters, enter:

```
ln manual/* .
```

This links all files in the `manual` directory into the current directory, `.` (dot), giving them the same names they have in the `manual` directory.

Note: You must type a space between the asterisk and the period.

5. To create a symbolic link, enter:

```
ln -s /tmp/toc toc
```

This creates the symbolic link, `toc`, in the current directory. The `toc` file points to the `/tmp/toc` file. If the `/tmp/toc` file exists, the `cat toc` command lists its contents.

To achieve identical results without designating the *TargetFile* parameter, enter:

```
ln -s /tmp/toc
```

6. To create a hard link to the symbolic link of the `toc` file, enter:

```
ln -P toc target
```

7. To create a hard link to the `/tmp/toc` file that is referenced by the symbolic link, enter:

```
ln -L toc target
```

Files

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

locale Command

Purpose

Writes information to standard output about either the current locale or all public locales.

Syntax

```
locale [ -O 64 ] [ -a | -m ] [ [ -c ] [ -k ] Name ... ]
```

Description

The `locale` command writes information to standard output about either the current locale or all public locales. A public locale is a locale available to any application.

To write the name and value of each current locale category, do not specify any flags or variables. To write the names of all available public locales, specify the **-a** flag. To write a list of the names of all available character-mapping (charmap) files, specify the **-m** flag. These charmap filenames are suitable values for the **-f** flag specified with the **localedef** command.

To write information about specified locale categories and keywords in the current locale, specify the *Name* parameter. The *Name* parameter can be one of the following:

- A locale category, such as **LC_CTYPE** or **LC_MESSAGES**
- A keyword, such as **yesexpr** or **decimal_point**
- The **charmap** reserved word to determine the current character mapping

You can specify more than one *Name* parameter with the **locale** command.

If you specify the **locale** command with a locale category name and no flags, the **locale** command writes the values of all keywords in the locale category specified by the *Name* parameter. If you specify the **locale** command with a locale keyword and no flags, the **locale** command writes the value of the keyword specified by the *Name* parameter.

If the *Name* parameter is a locale category name or keyword, the **-c** and **-k** flags can determine the information displayed by the **locale** command.

Flags

Item	Description
-a	Writes the names of all available public locales.
-c	Writes the names of selected locale categories. If the <i>Name</i> parameter is a keyword, the locale command writes the name of the locale category that contains the specified keyword, and the value of the specified keyword. If the <i>Name</i> parameter is a locale category, the locale command writes the name of the specified locale category and the values of all keywords in the specified locale category.
-k	Writes the names and values of selected keywords. If the <i>Name</i> parameter is a keyword, the locale command writes the name and value of the specified keyword. If the <i>Name</i> parameter is a locale category, the locale command writes the names and values of all keywords in the specified locale category.
-m	Writes the names of all available character-mapping (charmap) files.
-ck	Writes the name of the locale category, followed by the names and values of selected keywords. If the <i>Name</i> parameter is a keyword, the locale command writes the name of the locale category that contains the specified keyword, and the name and value of the specified keyword. If the <i>Name</i> parameter is a locale category, the locale command writes the name of the specified locale category and the names and values of all keywords in the specified locale category.
-O 64	Displays locale information as seen by a 64 bit executable. This should be identical to information as seen by a 32 bit executable.

Exit Status

This command returns the following exit values:

Ite	Description
m	
0	All the requested information was found and output successfully.
>0	An error occurred.

Examples

1. To retrieve the names and values of all the current locale environment variables, enter:

```
locale
```

If `locale_x` and `locale_y` are valid locales on the system, as determined with `locale -a`, and if the locale environment variables are set as follows:

```
LANG=locale_x  
LC_COLLATE=locale_y
```

The **locale** command produces the following output:

```
LANG=locale_x  
LC_CTYPE="locale_x"  
LC_COLLATE=locale_y  
LC_TIME="locale_x"  
LC_NUMERIC="locale_x"  
LC_MONETARY="locale_x"  
LC_MESSAGES="locale_x"  
LC_ALL=
```

Note: When setting the locale variables, some values imply values for other locale variables. For example, if the **LC_ALL** locale variable is set to the **En_US** locale, all locale environment variables are set to the **En_US** locale. In addition, implicit values are enclosed in double quotes ("). Explicitly set values are not enclosed in double quotes (").

2. To determine the current character mapping, enter:

```
locale charmap
```

If the **LC_ALL** locale variable is set to the C locale, the **locale** command produces the following output:

```
ISO8859-1
```

3. To retrieve the value of the `decimal_point` delimiter for the current locale, enter:

```
locale -ck decimal_point
```

If the **LC_ALL** locale variable is set to the C locale, the **locale** command produces the following output:

```
LC_NUMERIC  
decimal_point="."
```

localedef Command

Purpose

Converts locale and character set description (charmap) source files to produce a locale database.

Syntax

```
localedef [ -c ] [ -f Charmap ] [ -i SourceFile ] [ -L LinkOptions ] [ -m MethodFile ] LocaleName
```

Description

The **localedef** command converts source files that contain definitions of locale-dependent information (such as collation, date and time formats, and character properties) into a locale object file used at run-time. The locale object file created by the **localedef** command is then used by commands and subroutines that set the locale with the **setlocale** subroutine.

The **-i** *SourceFile* flag and variable specify the file that contains the source category definitions. If the **-i** flag is not specified, the file is read from standard input.

The **-f CharMap** flag and variable specify a file that maps character symbols to actual character encodings. Using the **-f** flag allows one locale source definition to be applicable to more than one code set. If the **-f** flag is not specified, the default value for the *CharMap* variable is ISO8859-1.

The *LocaleName* parameter specifies the locale name for the locale database generated by the **localedef** command from the specified source files. The *LocaleName* parameter can be either an absolute path name for the file location or a relative path name.

If a locale category source definition contains a copy statement and the statement names an existing locale installed in the system, the **localedef** command proceeds as though the source definition contained the valid category source definition for the named locale.

Notes:

1. The **localedef** command uses the C compiler to generate the locale database. Therefore, to use this command you must have the C compiler installed.
2. When replacing systemwide databases, it is advisable to do a soft reboot to ensure that the new locale is used throughout the system.

If an error is detected, no permanent output is created.

If warnings occur, permanent output is created when the **-c** flag is specified. The following conditions cause warning messages to be issued:

- A symbolic name not found in the file pointed to by the *Charmap* variable is used for the descriptions of the **LC_TYPE** or **LC_COLLATE** categories. This is an error condition for other categories.
- The number of operands to the **order_start** keyword exceeds the **COLL_WEIGHTS_MAX** limit.
- Optional keywords not supported by the implementation are present in the source file.

Flags

Item	Description
-c	Forces the creation of locale tables even if warning messages have been issued.
-f CharMap	Specifies the name of a file containing a mapping of character symbols and collating element symbols to actual character encodings. A locale is associated with one and only one code set. If this flag is not specified, the ISO 8859-1 code set is assumed. Note: The use of certain system-provided <i>CharMap</i> files is fully supported. However, while correctly defined user-provided <i>CharMap</i> files may work properly, the result of such use is not guaranteed.
-i SourceFile	Specifies the path name of a file containing the locale category source definitions. If this flag is not present, source definitions are read from standard input.
-L LinkOptions	Passes the specified link options to the ld command used to build the locale.
-m MethodFile	Specifies the name of a method file that describes the methods to override when constructing a locale. The method file specifies user-supplied subroutines that override existing definitions, as well as a path name for the library containing the specified subroutines. The localedef command reads the method file and uses entry points when constructing the locale objects. The code set methods specified are also used in parsing the file pointed to by the <i>CharMap</i> variable. Note: To create a 64-bit locale, the method file must specify the path of the library as a single archive that has two shared objects, one 32-bit and the other 64-bit, that contain the specified subroutines. Specifying separate paths to the 32-bit and 64-bit shared objects causes the localedef command to fail due to incompatible XCOFF format.

Item	Description
<i>LocaleName</i>	Specifies the name of the locale to be created. This is the name that can subsequently be used to access this locale information.

Exit Status

The **localedef** command returns the following exit values:

Item	Description
0	No errors occurred and the locales were successfully created.
1	Warnings occurred and the locales were successfully created.
2	The locale specification exceeded limits or the code set or sets used were not supported by the implementation, and no locale was created.
3	The capability to create new locales is not supported.
>3	Warnings or errors occurred and no locales were created.

Examples

1. To create a locale called `Austin` from standard input and disregard warnings, enter:

```
localedef -c Austin
```

2. To create a locale called `Austin` with `Austin.src` as source input, enter:

```
localedef -i Austin.src Austin
```

lock Command

Purpose

Reserves a terminal.

Syntax

lock [*-Timeout*]

Description

The **lock** command requests a password from the user, reads it, and requests the password a second time to verify it. In the interim, the command locks the terminal and does not relinquish it until the password is received the second time or one of the following occurs:

- The timeout interval is exceeded.
- The command is killed by a user with appropriate permission.

The timeout default value is 15 minutes, but this can be changed with the *-Timeout* flag.

Flags

Item	Description
<code>-Timeout</code>	Indicates the timeout interval in minutes, as specified by the <i>Timeout</i> parameter. The default value is 15 minutes.

Examples

1. To reserve a terminal under password control, enter:

```
lock
```

You are prompted for the password twice so the system can verify it. If the password is not repeated within 15 minutes, the command times out.

2. To reserve a terminal under password control, with a timeout interval of 10 minutes, enter:

```
lock -10
```

Files

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

lockd Daemon

Purpose

Processes lock requests.

Syntax

```
/usr/sbin/rpc.lockd [ -t TimeOut ] [ -g GracePeriod ] [ -d debug ] [ -x xnfs ] [ -T RetransmissionsTimeout ] [ number of server ]
```

Description

The **lockd** daemon processes lock requests that are either sent locally by the kernel or remotely by another lock daemon. The **lockd** daemon forwards lock requests for remote data to the server site lock daemon through the RPC package. The **lockd** daemon then asks the **statd** (status monitor) daemon for monitor service. The reply to the lock request is not sent to the kernel until both the **statd** daemon and the server site **lockd** daemon reply. The **statd** daemon should always be started before the **lockd** daemon.

If either the status monitor or the server site lock daemon is unavailable, the reply to a lock request for remote data is delayed until all daemons become available.

When a server recovers, it waits for a grace period for all client site **lockd** daemons to submit reclaim requests. The client site **lockd** daemons, on the other hand, are notified of the server recovery by the **statd** daemon. These daemons promptly resubmit previously granted lock requests. If a **lockd** daemon fails to secure a previously granted lock at the server site, the **lockd** daemon sends a SIGLOST signal to the process.

The **lockd** daemon is started and stopped with the following System Resource Controller (SRC) commands:

```
startsrc -s rpc.lockd
stopsrc -s rpc.lockd
```


To modify the arguments passed to the **lockd** daemon when it is started, use the following command:

```
chssys -s rpc.lockd Parameters...
```

The status monitor maintains information about the location of connections as well as the status in the **/var/statmon/sm** directory, the **/var/statmon/sm.bak** file, 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.

By default **rpc.lockd** establishes a dynamic socket port number for receiving requests. Entries may be added to the **/etc/services** file specifying the port that **rpc.lock** will listen for requests on. The service name is **lockd** and a unique port number should be specified. The following entries in **/etc/services** file would specify that port 16001 be used for both **tcp** and **udp**.

```
lockd 16001/tcp  
lockd 16001/udp
```

Flags

Item	Description
-d <i>debug</i>	Specifies the debug level of the rpc.statd daemon. By default, the debug level is disabled.
-g <i>GracePeriod</i>	Uses the <i>GracePeriod</i> variable to specify the amount of time, in seconds, that the lockd daemon should wait for reclaim requests for previously granted locks. The default value of the <i>GracePeriod</i> variable is 45 seconds.
-T <i>RetransmissionsTimeout</i>	Specifies the timeout for one-way RPC connections. One-way RPC connections stay valid for <i>RetransmissionsTimeout</i> number of seconds. If this variable is set to 0, then there is no client cache for one-way RPC calls. The default value for the <i>RetransmissionsTimeout</i> variable is 300 seconds.
-t <i>TimeOut</i>	Uses the <i>TimeOut</i> variable to specify the interval between retransmitting lock requests to the remote server. The default value for the <i>TimeOut</i> variable is 15 seconds.
-x <i>xnfs</i>	Specifies if the rpc.lockd daemon needs to follow the xnfs specification. By default, this flag is turned off.

Parameters

Item	Description
<i>number of server</i>	Specifies the number of daemons to start.

Examples

1. To specify a grace period, enter:

```
/usr/sbin/rpc.lockd -g 60
```

In this example, the grace period is set for 60 seconds.

2. To specify the amount of time the **lockd** daemon should wait before retransmitting a lock request, enter:

```
/usr/sbin/rpc.lockd -t 30
```

In this example, the retransmissions occur after 30 seconds.

Files

Item	Description
<code>/etc/services</code>	Contains lockd parameter information entries.

locktrace Command

Purpose

Controls kernel lock tracing.

Syntax

```
locktrace [ -r ClassName | -s ClassName | -S | -R | -l ]
```

Description

The **locktrace** command controls which kernel locks are being traced by the **trace** subsystem. The default is to **trace** none. If the machine has been rebooted after running the `bosboot -L` command, kernel lock tracing can be turned on or off for one or more individual lock classes, or for all lock classes. If `bosboot -L` was not run, lock tracing can only be turned on for all locks or none. The **trace** events collected in this case when locks are taken or missed (hook id 112), and released (hook id 113) do not have the lock class name available.

Flags

Item	Description
-r <i>classname</i>	Turns off lock tracing for all kernel locks belonging to the specified class. The option always fails if you did not run the <code>bosboot -L</code> command.
-s <i>classname</i>	Turns on lock tracing for all kernel locks belonging to the specified class. The option always fails if you did not run the <code>bosboot -L</code> command. To trace several specific classes at the same time, run the locktrace command multiple times, with a specific lock class each time. You can enter up to 32 class names.
-R	Turns off all lock tracing.
-S	Turns on lock tracing for all locks regardless of their class membership.
-l	Lists kernel lock tracing current status.

Examples

1. To start tracing the `SEM_LOCK_CLASS`, enter the following command:

```
locktrace -s SEM_LOCK_CLASS
```

2. To stop all lock tracing, enter the following command:

```
locktrace -R
```

3. To reset previous lock trace entries and then trace the `SEM_LOCK_CLASS` and `SHM_LOCK_CLASS` lock classes, enter the following commands:

```
locktrace -R  
locktrace -s SEM_LOCK_CLASS  
locktrace -s SHM_LOCK_CLASS
```

You can view current lock classes using the **-l** flag:

```
locktrace -l
```

The following output will be displayed:

```
lock tracing enabled for classes:
SHM_LOCK_CLASS
SEM_LOCK_CLASS
```

File

Item	Description
<code>/usr/bin/locktrace</code>	Contains the locktrace command.
<code>/usr/include/sys/lockname.h</code>	Contains the lock class names.

logevent Command

Purpose

Logs event information generated by the event response resource manager (ERRM) to a specified log file.

Syntax

```
logevent [-h] log_file
```

Description

The `logevent` captures event information that is posted by the event response resource manager (ERRM) in environment variables the ERRM generates when an 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 `logevent` script are returned depend on the locale settings.

Event information that is returned about the ERRM environment variables 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 uses the `alog` command to write event information to and read event information from the specified *log_file*.

Flags

-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

The script has run successfully.

1

A required *log_file* is not specified.

2

The *log_file* path is not valid.

Restrictions

- This script must be run on the node where the ERRM is running.
- The user who runs this script must have write permission for the *log_file* where the event information is logged.

Standard Output

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

Examples

1. To log information, specify `/tmp/event.log` as follows:

```
/opt/rsct/bin/logevent  
/tmp/event.log
```

The `/tmp/event.log` file does not need to exist when the command is run.

2. To see the contents of the `/tmp/event.log` file, run this command:

```
alog -f /tmp/event.log -o
```

The following sample output shows a warning event for the `/var` file system (a file system resource):

```
=====  
Event reported at Mon Mar 27 16:38:03 2007  
  
Condition Name:          /var space used  
Severity:                Warning  
Event Type:              Event  
Expression:              PercentTotUsed>90  
  
Resource Name:           /var  
Resource Class Name:     IBM.FileSystem  
Data Type:               CT_UINT32  
Data Value:              91
```

Location

`/opt/rsct/bin/logevent`

logform Command

Purpose

Initializes a logical volume for use as a Journaled File System (JFS) log. Initializes an Enhanced Journaled File System (JFS2) outline log. Reformats an inline log for an existing JFS2 file system using an inline log.

Syntax

logform [**-V** *vfstype*] *LogName*

Description

The **logform** command initializes a logical volume for use as a JFS or JFS2 log device. Running the **logform** command on any JFS log device or JFS2 outline or inline log device will destroy all log records on the log device. This may cause the file system to lose its recovery capability and therefore to lose the file system data.

When you run the **logform** command on an outline log for a JFS2 file system that is already using an outline log, the device type for the outline log must be **jfs2log**. Otherwise, the **logform** command will exit with an error.

To reuse an existing logical volume as an outline log device for a JFS2 file system, you must delete the logical volume and then recreate it as device type **jfs2log**.

For the outline log device of a JFS file system, the same rules are applied. That is, for a new logical volume, the type should be **jfslog**. For a reuse logical volume, you should delete the logical volume and recreate it as lv type **jfslog**. However, **logform** does not do type check for the log device of a JFS file system. The **logform** command does not report any error when input log device has a wrong lv type. The user should pay attention to the lv type.

When you run the **logform** command on a device with logical volume type **jfs2**, if the device has a file system with an inline log, then the inline log will be reformatted. If the device has a file system with an outline log, then an error will be reported.

When the **logform** command is used to format an inline log for an existing JFS2 file system, the file system data will not be affected; only the log records are destroyed. The logical volume type for an inline log is the same as for the file system. For a JFS2 file system, the inline log logical volume type is **jfs2**.

For a JFS2 file system, **logform** formats a maximum of 2047 MBytes of log. If the log size is greater than 2047 MBytes, only 2047 MBytes will be formatted and the rest will left untouched and will not be used.

Flags

Item	Description
-V <i>vfstype</i> [jfs jfs2]	If specified, <i>vfstype</i> indicates what type of file system the log should be formatted for. If this option is not specified, then the type is obtained from the logical volume's type. Note that for jfs2 log device this flag is always ignored. The logform command is unable to change the lv type according to the value of the -V flag. Therefore the user should create an lv with the correct lv type (jfslog or jfs2log) before calling the logform command. Use of this flag is strongly discouraged.

Parameters

Item	Description
<i>LogName</i>	The <i>LogName</i> parameter specifies the absolute path to the logical volume to be initialized (for example, <i>/dev/jfslog1</i>). When the logform command is run on an inline log, <i>LogName</i> is the device name of the file 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.

Examples

1. To create a JFS logging device on a newly created volume group, first create a logical volume of type **jfslog**:

```
mklv -t jfslog -y jfslog1 newvg 1
```

This command creates a `jfslog` logical volume named `jfslog1` in the volume group `newvg`. The size of the logical volume is 1 logical partition.

2. To format the `jfslog1` logical volume once it has been created, enter:

```
logform /dev/jfslog1
```

The `jfslog1` logical volume is now ready to be used as a JFS log device.

3. To format the inline log for an existing file system called `/j2` which is on the file system device `/dev/fslv00`, type:

```
logform /dev/fslv00
```

This formats the inline log for file system `/j2`, but does not touch the data in the file system.

Files

Item	Description
<code>/etc/filesystems</code>	Lists the known file systems and defines their characteristics, including the log device.

logger Command

Purpose

Makes entries in the system log.

Syntax

```
logger [ -f File ] [ -i ] [ -r [Count] ] [ -p Priority ] [ -t Tag ] [ Message ]
```

Description

The **logger** command provides an interface to the **syslog** subroutine, which writes entries to the system log. A *Message* variable can be specified on the command line, which is logged immediately, or a *File* variable is read and each line of the *File* variable is logged. If you specify no flags or variables, the **logger** command will wait for you to enter a message from standard input. The messages returned by the **LOG_KERN** facility cannot be logged by this command.

Flags

Item	Description
-f <i>File</i>	Logs the specified <i>File</i> variable. If the <i>Message</i> variable is specified, this flag is ignored.
-i	Logs the process ID of the logger process with each line.
-p <i>Priority</i>	Enters the message with the specified priority. The <i>Priority</i> parameter may be a number or a <i>facility.level</i> priority specifier.
-t <i>Tag</i>	Marks every line in the log with the specified <i>Tag</i> parameter.
<i>Message</i>	Indicates the message to log. If this variable is not specified, the logger command logs either standard input or the file specified with the -f <i>File</i> flag.

Item	Description
<code>-r Count</code>	If the buffer resource is not available, retries logging the message for the specified number of times. If the number is not specified, retries logging the message until the message is logged. The number must be a positive integer ranging from 1 - 1000.

Examples

1. To log a message indicating a system reboot, enter:

```
logger System rebooted
```

2. To log a message contained in the `/tmp/msg1` file, enter:

```
logger -f /tmp/msg1
```

3. To log the daemon facility critical level messages, enter:

```
logger -pdaemon.crit
```

Exit Status

This command returns the following exit values:

Item	Description
<code>0</code>	Successful completion.
<code>>0</code>	An error occurred.

Files

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

login Command

Purpose

Initiates a user session.

Syntax

```
login [ -h HostName ] [ -p ] [ -f User | -k ] [ -e Label ] [ -t Label ] [ User [ Environment ] ]
```

Description

The **login** command (part of the **tsm** command) initiates sessions on the system for the user that is specified by the *User* parameter. You can also specify environment variables to be added to the user's environment. These commands are strings of the form *Variable=Value*. The **login** command is not normally entered on the command line.

You can configure the **login** command to create your home directory at your login if you do not have a home directory already. The **login** 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`.

Note:

1. The **PATH**, **IFS**, **HOME**, and **SHELL** environment variables can not be initialized from the command line.
2. The **login** command supports multibyte user names. It is recommended that the system administrator must restrict the user names to characters within the portable character set, to avoid any ambiguity.
3. If the **/etc/nologin** file exists, the system prevents the user from logging in and displays the contents of the **/etc/nologin** file. The system does allow the root user to log in if this file exists. The **/etc/nologin** file is removed when you restart the system.
4. If the *domainlessgroups* attribute is set in the **/etc/secvars.cfg** file, all group IDs are fetched from the LDAP module, and from the files modules, if the user belongs to any one of these domains.

The **login** command can handle Distributed Computing Environment (DCE) user names of up to 1024 characters. DCE user names are stored in the **LOGIN** environment variable. Because DCE user names do not conform to standard operating system requirements, the first 8 characters of the DCE user name are stored in all standard operating system files and environments.

The **login** command performs the following functions:

Item	Description
Checks accounts	The login command validates the user's account, ensuring authentication, logins enabled properly, and correct capacity for the port that is used for the login.
Authenticates users	The login command verifies the user's identity by using the system defined authentication methods for each user. If a password has expired, the user must supply a new password. If secondary authentication methods are defined, these methods are invoked but need not be successful in logging in to the system.
Establishes credentials	The login command establishes the initial credentials for the user from the user database. These credentials define the user's access rights and accountability on the system.
Initiates a session	The login command initializes the user environment from the user database, from the command line, and from the /etc/environment configuration file; changes the current directory to the user's home directory (normally); and runs the user's initial program.

These functions are performed in the order given; if one fails, the functions that follow are not performed.

When a user logs in successfully, the **login** command makes entries in the **/etc/utmp** file that tracks current user logins and the **/var/adm/wtmp** file that is used for accounting purposes. The **login** command also sets the **LOGIN** and **LOGNAME** environment variables.

Information pertaining to each unsuccessful login is recorded in the **/etc/security/failedlogin** file. The information that is stored is the same as that in the **/etc/utmp** file, except that unrecognizable user names are logged as UNKNOWN_USER. This check ensures that a password accidentally entered as a user name, for example, is not allowed into the system unencrypted.

After a successful login, the **login** command displays the message of the day, the date and time of the last successful and unsuccessful login attempts for this account, and the total number of unsuccessful login attempts for this account since the last successful login. These messages are suppressed if there is a **.hushlogin** file in your home directory.

The **login** command also changes the ownership of the login port to the user. This includes any ports noted as synonyms in the **/etc/security/login.cfg** file.

To preserve the integrity of the system, only one session at a time is allowed to be logged in to a port. This check means that the **login** command entered from the shell prompt cannot succeed, as both the original session and the new login session would be on the same port. However, the **exec login** command

succeeds, because a new shell replaces the current one. The **login** command is typically a built-in shell command, causing the shell to replace itself.

If the label has spaces, specify it within quotation marks. The default login SL and TL are defined in the **/etc/security/user** file as user attributes. If no label attribute is specified in the file, the label attributes that are defined in the default stanza are used.

The labels that you supply must be dominated by your clearance and contained in the system accreditation range. You can specify the SL with the **-e** flag and the TL with the **-t** flag at login time. In a labeled network, unless the login is done by using the console, the network's label is assigned to you, regardless of the labels that you specified with the **-e** or **-t** flag.

Your SL clearance must be within the range that is defined for the TTY device in the **/etc/security/login.cfg** file. The effective TL of the user must be the same as the TL of the TTY. After successfully logging in, the clearance is assigned to the login port.

Tip:

Unless your terminal displays only uppercase letters, do not use only uppercase characters for your user name.

To log in with multibyte user names, you must first open a Japanese window (aixterm) and initiate a new login from the Japanese window.

Flags

Item	Description
-f <i>User</i>	Identifies a user who has already been authenticated. If the real ID of the login process is root (0), then the user is not authenticated.
-h <i>HostName</i>	Identifies the login as a remote login and specifies with the <i>HostName</i> variable the name of the system that is requesting the login. This form of the login is used only by the telnetd and rlogind daemons.
-k	Identifies the login as using Kerberos authentication and causes login to pass control to /usr/bin/k5dcelogin to handle authentication. This form of login is only used by the krshd daemon.
-p	Preserves the current terminal type by setting it the value of the \$TERM environment variable instead of the type that is contained in the CuAt/PdAt object classes database.

Security

The **login** command is a PAM-enabled application with a service name of **login**. 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 that are used when PAM is enabled depend on the configuration for the login service in **/etc/pam.conf**. The **login** command requires **/etc/pam.conf** entries for the **auth**, **account**, **password**, and **session** module types. The following is a recommended configuration in **/etc/pam.conf** for the login service:

```
#
# AIX login configuration
#
login auth required /usr/lib/security/pam_aix
login account required /usr/lib/security/pam_aix
login session required /usr/lib/security/pam_aix
login password required /usr/lib/security/pam_aix
```

Examples

1. To log in to the system as user `jamesd`, enter the following at the login prompt:

```
login: jamesd
```

If a password is defined, the password prompt appears. Enter your password at this prompt.

2. To log in with the effective SL of `SECRET`, and the effective TL of `TOP SECRET`, enter the following command:

```
login: james -e "TOP SECRET" -t "TOP SECRET"
```

3. On the command line the following can be used:

```
$ login -e "TOP SECRET" james
```

Files

Item	Description
<u>/usr/sbin/login</u>	Contains the login command.
<u>/etc/utmp</u>	Contains accounting information.
<u>/var/adm/wtmp</u>	Contains accounting information.
<u>/etc/motd</u>	Contains the message of the day.
<u>/etc/passwd</u>	Contains passwords.
<u>\$HOME/.hushlogin</u>	Suppresses login messages.
<u>/etc/environment</u>	Contains user environment configuration information.
<u>/etc/security/login.cfg</u>	Contains port synonyms.
<u>/etc/security/lastlog</u>	Contains information that pertains to the most recent successful and unsuccessful login attempts.
<u>/etc/security/failedlogin</u>	Contains information that pertains to each unsuccessful login.

logins Command

Note: **Logins** command displays system login information details only for the local users or groups which are defined in the `/etc/passwd` and `/etc/group` files.

Purpose

Displays user and system login information.

Syntax

```
logins [ -a ] [ -m ] [ -o ] [ -p ] [ -s ] [ -t ] [ -u ] [ -x ] [ -g Groups ] [ -l Logins ]
```

Description

The **logins** command displays information about user and system logins. By default, the **logins** command prints the following items:

- User ID
- Primary group name
- Primary group ID

- The `/etc/passwd` account field on user information.

The output is sorted by user ID, displaying system logins followed by user logins.

Depending on the options chosen, the following fields can also be displayed:

- user or system login
- user ID number
- multiple group names
- multiple group IDs
- home directory
- login shell
- four password aging parameters
- `/etc/passwd` account field value (user name or other information)
- primary group name
- primary group ID

Flags

Item	Description
-a	In addition to the default output, the -a flag adds two password expiration fields to the display. These fields show how many days a password can remain unused before it automatically becomes inactive and the date that the password will expire.
-g Groups	Displays all users belonging to group, sorted by user ID. Multiple groups can be specified as a comma separated list. <i>Groups</i> must specify valid group names on the system. Comma separate names when specifying more than one group.
-l Logins	Displays the requested login. Multiple logins can be specified as a comma-separated list. <i>Logins</i> must specify valid user names on the system.
-m	Displays multiple group membership information.
-o	Formats output into one line of colon separated fields.
-p	Displays users without passwords.
-s	Displays all system logins.
-t	Sorts output by user name instead of by user ID.
-u	Displays all user logins.
-x	Prints an extended set of information about each selected user. Information for each user is printed on a separate line containing the home directory, login shell, and password aging information. The extended information includes the following: <ul style="list-style-type: none"> • The password status • The date on which the password was last changed • The number of days required between changes • The number of days allowed before a change is needed • The number of days that the user will receive a password expiration warning message before the password expires <p>The password status is displayed in an abbreviated form as PS for logins with password, NP for no password or LK for locked.</p>

Exit Status

0

The command completed successfully.

>0

An error occurred.

Examples

1. To list all the logins with no passwords, enter:

```
logins -p
```

The output looks similar to the following:

```
pwdless    204    staff      1
nopwd      208    staff      1
```

The **-p** option ensures that only logins with no passwords are listed.

2. To list all the system logins sorted by alphabetical order enter:

```
logins -st
```

The output looks similar to the following:

```
adm         4      adm         4
bin         2      bin         2
daemon     1      staff       1
lp         11     lp          11
lpd         9      nobody     -2
root        0      system      0
sys         3      sys         3
uucp        5      uucp        5
```

The **-t** option prints out the logins sorted alphabetically and not by uid.

3. To list the login details of users "root" and "admin", enter:

```
logins -l root,adm
```

The output looks similar to the following:

```
root        0      system      0
adm         4      adm         4
```

4. To list the password aging details of users "root" and "admin" enter:

```
logins -xl root,adm
```

The output looks similar to the following:

```
root        0      system      0
            /
            /usr/bin/ksh
            PS 021102 0 0 0
adm         4      adm         4
            /var/adm
            /sbin/sh
            PS 000000 0 0 0
```

The **-x** option ensures that extended password information for these logins are retrieved and printed in the output.

5. To display the multiple group information of a particular user in a colon separated format enter:

```
logins -mol root,adm
```

The output looks similar to the following:

```
root:0:system:0::bin:2:sys:3:security:7:cron:8:audit:10:lp:11
adm:4:adm:4:
```

The **-m** option is used here to retrieve the multiple group information of a particular login (user). The **-o** option ensures that the output is displayed in colon separated format.

6. To display the users of the "staff" and "sys" groups in a colon separated format, sorted by user name, enter:

```
logins -tsog staff,sys
```

The output looks similar to the following:

```
bin:2:bin:2:
daemon:1:staff:1:
invscout:200:staff:1:
root:0:system:0:
sys:3:sys:3:
```

Files

Item	Description
<code>/usr/bin/logins</code>	Contains the logins command.
<code>/etc/passwd</code>	Contains the password file.
<code>/etc/group</code>	Contains the group file.

logname Command

Purpose

Displays login name.

Syntax

```
logname
```

Description

The **logname** command displays the login name of the current process. This is the name that the user logged in with and corresponds to the **LOGNAME** variable in the system-state environment. This variable is only set when the user logs into the system.

The **logname** command invokes the **getlogin** subroutine to get the information about the login name.

Security

Access Control: This program is installed as a normal user program in the Trusted Computing Base.

Exit Status

This command returns the following exit values:

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

Examples

To display your login name to standard output, enter:

```
logname
```

Files

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

logout Command

Purpose

Stops all processes on a port.

Syntax

```
logout
```

Description

The **logout** command terminates all processes either with the same controlling terminal as the present process or with all processes which have this terminal open. Processes that are not children of the present process are terminated upon access to the terminal. The present process is also terminated. If the **login** command user and the **logout** command user do not match, the **logout** command permission is denied, and the command stops.

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

From a shell started by the **ksh** or **bsh** command, enter:

```
logout
```

Files

Item	Description
<code>/usr/bin/logout</code>	Contains the logout command.
<code>/etc/utmp</code>	Contains a record of logged-in users.

look Command

Purpose

Finds lines in a sorted file.

Syntax

```
look [ -d ] [ -f ] String [ File ... ]
```

Description

The **look** command searches sorted files specified by the *File* parameter and prints all lines that begin with the string specified by the *String* parameter. The **look** command uses a binary search, therefore files specified by the *File* parameter must be sorted in the C locale collating sequence.

The **-d** and **-f** flags affect comparisons as in the **sort** command. This means a file must be sorted using the **-f** flag in the **sort** command before using the **look** command with the **-f** flag.

If the *File* parameter is not specified, the **/usr/share/dict/words** file is assumed with the collating sequence specified by the **-df** flags. The sort is completed using the current collating sequence. This should match the collating sequence used to produce the dictionary file. The **look** command limits the length of a word search to 256 characters.

Flags

Item	Description
------	-------------

- | | |
|-----------|---|
| -d | Specifies dictionary order. Only letters, digits, tabs, and spaces are considered in comparisons. |
| -f | Compares uppercase and lowercase letters as equivalent values. Case is not considered in the sorting so that initial-capital and all-capital words are not grouped together at the beginning of the output. |

Note: To use the **look -f** command, the input file must be sorted with the **sort -f** command.

Example

To search for all lines in the `sortfile` file that begin with the letter `a`, enter:

```
look a sortfile
```

File

Item	Description
/usr/share/dict/words	Contains the default dictionary.

lookbib Command

Purpose

Finds references in a bibliography.

Syntax

```
lookbib [ -n ] [ Database ... ]
```

Description

The **lookbib** command uses an inverted index made by the **indxbib** command to find sets of bibliographic references. The **lookbib** command reads keywords typed after the `>` prompt on the terminal, and

retrieves records containing all these keywords. If nothing matches, nothing is returned except another > prompt.

The **lookbib** command asks if you need instructions and prints some brief information if you type a user-defined affirmative answer.

The *Database* parameter specifies files that contain bibliographic references, indexes, or similar types of information. It is possible to search multiple databases as long as they have a common index made by the **indxbib** command. In that case, only the first database name given to the **indxbib** command is specified to the **lookbib** command.

If the **lookbib** command does not find the index files (the **.i[abc]** files), it looks for a reference file with the same name as the first database, but without the suffixes. It creates a file with a **.ig** suffix, suitable for use with the **fgrep** command. It then uses this **fgrep** command file to find references. Using the **.ig** file is simpler but slower than using the **.i[abc]** files, and does not allow the use of multiple reference files.

Flags

Item	Description
------	-------------

-n	Turns off the prompt for instructions.
-----------	--

Files

Item	Description
<i>Database.ia</i>	Contains the entry file.
<i>Database.ib</i>	Contains the posting file.
<i>Database.ic</i>	Contains the tag file.
<i>Database.ig</i>	Contains the output file.

loopmount Command

Purpose

Associate an image file to a loopback device. Optionally, make an image file available as a file system via the loopback device.

Syntax

```
loopmount { -i imagefile | -l device } [-o mount options -m mountpoint ]
```

Description

This command is similar to **mount** except that it creates a loopback device if not specified, binds the specified file to it, and optionally mounts it. If the command implicitly creates a new loopback device, it sets the temporary attribute in CuAt to **yes** so that it will be deleted by subsequent loopmount or reboot. All the restrictions and features of the **mount** command apply to **loopmount** also.

Flags

Item	Description
-i	Specify an image file name such as an ISO image. This must be specified if -l is not specified.

Item	Description
-l	ODM name of a loopback device such as loop0, loop1, etc. This must be specified if -i is not specified.
-o	Options for the mount command.
-m	Mount point such as /mnt .

If **-l** and **-i** are both specified, the *imagefile* is associated with the device before mounting it.

Security

Privilege Control: Only the root user and members of the system group should have execute (x) access to 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. The following mount an ISO image on /mnt.

```
$ loopmount -i cdrom.iso -o "-V cdrfs -o ro" -m /mnt
```

2. The following mounts a disk image on /mydisk with loop2. The image file was bound to loop2 earlier with **chdev** command.

```
$ loopmount -l loop2 -o "-V jfs2 -o rw,log=NULL" -m /mydisk
```

In case the filesystem was created with an INLINE log, this INLINE log can be used.

```
$ loopmount -l loop2 -o "-V jfs2 -o rw,log=INLINE" -m /mydisk
```

3. The following mounts an image file bound to loop0 on /mnt.

```
$ loopmount -i mycd.iso -l loop0 -o "-V cdrfs -o ro" -m /mnt
```

Files

Item	Description
/usr/sbin/loopmount	Contains the loopmount command.

loopumount Command

Purpose

Unmounts a previously mounted image file on a loopback device and then removes the device.

Syntax

```
loopumount { -i imagefile | -l device } [-o umount options -m mountpoint ]
```

Description

This command is similar to **umount** except that it unmount the file and deletes the loopback device associated with the mount point if the temporary attribute for the loopback device in CuAt is set to **yes**, and then unmounts it. All the restrictions and features of the **umount** command apply to **loopumount** also.

Flags

Item	Description
------	-------------

- | | |
|-----------|---|
| -i | Specify an image file name such as an ISO image. This must be specified if -l is not specified. |
| -l | ODM name of a loopback device such as loop0, loop1, etc. This must be specified if -i is not specified. |
| -o | Options for the umount command. |
| -m | Mount point such as /mnt. |

If **-i** and **-l** both are specified, **-i** is ignored.

Security

Privilege Control: Only the root user and members of the system group should have execute (x) access to 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. The following unmount /mnt and deletes the underlying device that was created by a previous invocation of **loopmount**.

```
$ loopumount -i cdrom.iso -o "/mnt"
```

2. The following command unmount /dev/loop2 from /mydisk but does not delete the device as loop2 was created by the user with **mkdev**.

```
$ loopumount -l loop2 -o "/mydisk"
```

Files

Item	Description
/usr/sbin/ loopumount	Contains the loopumount command.

lorder Command

Purpose

Finds the best order for member files in an object library.

Syntax

```
lorder [ -X {32|64|32_64}] File ...
```

Description

The **lorder** command reads one or more object or library archive files, looking for external references and writing a list of paired file names to standard output. The first pair of files contains references to identifiers that are defined in the second file.

If object files do not end with **.o**, the **lorder** command overlooks them and attributes their global symbols and references to some other file.

Flags

Item	Description
-X mode	Specifies the type of object file lorder 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 lorder to process any 64-bit objects and ignore 32-bit objects. The -X flag overrides the OBJECT_MODE variable.

Files

Item	Description
/tmp/sym*	Contains temporary files.

lp Command

The **lp** command includes information for the AIX Print Subsystem **lp** and the System V Print Subsystem **lp**.

AIX Print Subsystem lp Command

Purpose

Sends requests to a line printer.

Syntax

```
lp [ -c ] [ -dQueue ] [ -m ] [ -nNumber ] [ -oOption ] [ -s ] [ -tTitle ] [ -w ] [ Files ]
```

Description

The **lp** command arranges for the files specified by the *Files* parameter and their associated information (called a request) to be printed by a line printer. If you do not specify a value for the *Files* parameter, the **lp** command accepts standard input. The file name **-** (dash) represents standard input and can be specified on the command line in addition to files. The **lp** command sends the requests in the order specified. If the job is submitted to a local print queue, the **lp** command displays the following to standard output:

```
Job number is: nnn
```

where nnn is the assigned job number. To suppress the job number use the **-s** flag.

Flags

Item	Description
-c	Copies the files to be printed immediately when the lp command is run. The lp command copies files only when requested. No links are created. If you specify the -c flag, be careful not to remove any of the files before they are printed. If you do not specify the -c flag, changes made to the files after the request is made appear in the printed output.
-dQueue	Specifies the print queue to which a job is sent.
-m	Sends mail (see the mail command) after the files are printed. By default, no mail is sent upon normal completion of the print request.
-nNumber	Prints the number of copies of printed output. The default number of copies is 1.
-oOptions	Specifies that flags specific to the backend be passed to the backend. Thus, for each queue, other flags not described in this article can be included with the lp command. See the piobe command for a list of these flags. Specifying this flag is the same as specifying the -o flag for the enq command.
-s	Suppresses the automatic return of job numbers. The lp command reports the job number as the default, the -s flag overrides the default.
-tTitle	Specifies printing the title of the file on the banner page of the output.
-w	Writes a message on the print requesters terminal after the files are printed. If the requester is not logged in, the mail command sends the message. If the user is logged in on multiple windows or terminals, the message may not be sent to the LFT where the command was issued. The message is sent to the first terminal on which the writesrv daemon sees the user to be logged in.

Note: If the **-w** flag is used in conjunction with the **-m** flag, the print requester will only receive mail and will not get a message on the terminal.

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 the **/etc/motd** file on printer lp0 attached to device dlp0, enter:

```
lp /etc/motd
```

2. To print 30 copies of the **/etc/motd** file using a copy of the file, and to notify the user that the job is completed using mail, enter:

```
lp -c -m -n30 -dlp0:lpd0 /etc/motd
```

3. To print the **/etc/motd** file using backend flags **-f** and **-a**, with a job title of blah, enter:

```
lp -t"blah" -o -f -o -a /etc/motd
```

4. To queue the MyFile file and return the job number, enter:

```
lp myfile
```

5. To queue the MyFile file and suppress the job number, enter:

```
lp -s myfile
```

Exit Status

This command returns the following exit values:

Item Description

m

0 All input files processed successfully.

>0 No output device is available, or an error occurred.

Files

Item	Description
<u>/usr/sbin/qdaemon</u>	Contains the queuing daemon.
<u>/var/spool/lpd/qdir/*</u>	Contains the queue requests.
<u>/var/spool/lpd/stat/*</u>	Contains information on the status of the devices.
<u>/var/spool/qdaemon/*</u>	Contains temporary copies of enqueued files.
<u>/etc/qconfig</u>	Contains the queue configuration file.
<u>/etc/qconfig.bin</u>	Contains digested, binary version of the <u>/etc/qconfig</u> file.

System V Print Subsystem lp Command

Purpose (System V)

Sends print requests

Syntax (System V)

```
lp [print-options] [files]
```

```
lp -i request-ID print-options
```

Description (System V)

The first form of the **lp** command arranges for the named *files* and associated information (collectively called a request) to be printed. If filenames are not specified on the command line, the standard input is assumed. The standard input may be specified along with named *files* on the command line by listing the filenames and specifying **-** for the standard input. The *files* will be printed in the order in which they appear on the command line.

The LP print service associates a unique *request-ID* with each request and displays it on the standard output. This *request-ID* can be used later when canceling or changing a request, or when determining its status. See the **cancel** command for details about canceling a request, and **lpstat** for information about checking the status of a print request.

The second form of **lp** is used to change the options for a request submitted previously. The print request identified by the *request-ID* is changed according to the *print-options* specified with this command. The *print-options* available are the same as those with the first form of the **lp** command. If the request has finished printing, the change is rejected. If the request is already printing, it will be stopped and restarted from the beginning (unless the **-P** flag has been given).

If you enter `lp -?`, the system displays the command usage message and returns 0.

Sending a print request (System V)

The first form of the **lp** command is used to send a print request either to a particular printer or to any printer capable of meeting all requirements of the print request.

Flags must always precede filenames, but may be specified in any order.

Printers for which requests are not being accepted will not be considered when the destination is **any**. (Use the **lpstat -a** command to see which printers are accepting requests.) However, if a request is destined for a class of printers and the class itself is accepting requests, then all printers in the class will be considered, regardless of their acceptance status.

For printers that take mountable print wheels or font cartridges, if you do not specify a particular print wheel or font with the **-S** flag, whichever one happens to be mounted at the time your request is printed will be used. The **lpstat -p printer -l** command is used to see which print wheels are available on a particular printer. The **lpstat -S -l** command is used to see what print wheels are available and on which printers. Without the **-S** flag, the standard character set is used for printers that have selectable character sets.

If you experience problems with jobs that usually print but on occasion do not print, check the physical connections between the printer and your computer. If you are using an automatic data switch or an A/B switch, try removing it and see if the problem clears.

Flags (System V)

-c

Make copies of the *files* to be printed immediately when **lp** is invoked. Normally *files* will not be copied, but will be linked whenever possible. If the **-c** flag is not specified, the user should be careful not to remove any of the *files* before the request has been printed in its entirety. It should also be noted that if the **-c** flag is not specified, any changes made to the named *files* after the request is made but before it is printed will be reflected in the printed output.

-d dest

Choose *dest* as the printer or class of printers that is to do the printing. If *dest* is a printer, then the request will be printed only on that specific printer. If *dest* is a class of printers, then the request will be printed on the first available printer that is a member of the class. If *dest* is **any**, then the request will be printed on any printer that can handle it. Under certain conditions (unavailability of printers, file space limitations, and so on) requests for specific destinations may not be accepted (see **lpstat**). By default, *dest* is taken from the environment variable **LPDEST**. If **LPDEST** is not set, then *dest* is taken from the environment variable **PRINTER**. If **PRINTER** is not set, a default destination (if one exists) for the computer system is used. If no system default is set and **-T** is used, *dest* will be selected on the basis of *content-type* specified with the **-T** flag [see the description of **-T**]. Destination names vary between systems (see **lpstat**).

-f form-name [-d any]

Print the request on the form *form-name*. The LP print service ensures that the form is mounted on the printer. If *form-name* is requested with a printer destination that cannot support the form, the request is rejected. If *form-name* has not been defined for the system, or if the user is not allowed to use the form, the request is rejected. (see **lpforms**). When the **-d any** flag is given, the request is printed on any printer that has the requested form mounted and can handle all other needs of the print request.

-H special-handling

Print the request according to the value of *special-handling*. Acceptable values for *special-handling* are defined below:

hold

Do not print the request until notified. If printing has already begun, stop it. Other print requests will go ahead of a held request until it is resumed.

resume

Resume a held request. If it had been printing when held, it will be the next request printed, unless subsequently bumped by an **immediate** request. The **-i** flag (followed by a *request-ID*) must be used whenever this argument is specified.

immediate

(Available only to LP administrators) Print the request next. If more than one request is assigned **immediate**, the most recent request will be printed first. If another request is currently printing, it must be put on hold to allow this immediate request to print.

-L locale-name

Specify *locale-name* as the locale to use with this print request. By default, *locale-name* is set to the value of **LC_CTYPE**. If **LC_CTYPE** is not set, *locale-name* defaults to the C locale.

-m

Send mail after the files have been printed. By default, mail is not sent upon normal completion of the print request.

-n number

Print *number* copies of the output. The default is one copy.

-o options

Specify printer-dependent *options*. Several such *options* may be collected by specifying the **-o** keyletter more than once (that is, **-o option[1] -o option[2] ... -o option[n]**), or by specifying a list of options with one **-o** keyletter enclosed in double quotes and separated by spaces (that is, **-o "option[1] option[2] . . . option[n]"**).

nobanner

Do not print a banner page with this request. The administrator can disallow this option at any time.

nofilebreak

Do not insert a form feed between the files given, if submitting a job to print more than one file. This option is not supported by printers configured to use the PS (PostScript) interface.

length=scaled-decimal-number

Print this request with pages *scaled-decimal-number* long. A *scaled-decimal-number* is an optionally scaled decimal number that gives a size in lines, characters, inches, or centimeters, as appropriate. The scale is indicated by appending the letter **i** for inches, or the letter **c** for centimeters. For length or width settings, an unscaled number indicates lines or characters; for line pitch or character pitch settings, an unscaled number indicates lines per inch or characters per inch (the same as a number scaled with **i**). For example, **length=66** indicates a page length of 66 lines, **length=11i** indicates a page length of 11 inches, and **length=27.94c** indicates a page length of 27.94 centimeters. This option may not be used with the **-f** option and is not supported by the PS (PostScript).

width=scaled-decimal-number

Print this request with pages *scaled-decimal-number* wide. (See the explanation of *scaled-decimal-numbers* in the discussion of **length**, above.) This option may not be used with the **-f** option and is not supported by the PS (PostScript).

lpi=scaled-decimal-number

Print this request with the line pitch set to *scaled-decimal-number*. (See the explanation of *scaled-decimal-numbers* in the discussion of **length**, above.) This option may not be used with the **-f** flag and is not supported by the PS (PostScript).

cpi=pica|elite|compressed

Print this request with the character pitch set to **pica** (representing 10 characters per inch), **elite** (representing 12 characters per inch), or **compressed** (representing as many characters per inch as a printer can handle). There is not a standard number of characters per inch for all printers; see the **terminfo** database for the default character pitch for your printer. This option may not be used with the **-f** flag and is not supported by the PS (PostScript).

stty=stty-option-list

A list of options valid for the **stty** command; enclose the list with single quotes if it contains blanks.

-P page-list

Print the pages specified in *page-list*. This flag can be used only if there is a filter available to handle it; otherwise, the print request will be rejected. The *page-list* may consist of ranges of numbers, single page numbers, or a combination of both. The pages will be printed in ascending order.

-q priority-level

Assign this request *priority-level* in the printing queue. The values of *priority-level* range from 0 (highest priority) to 39 (lowest priority). If a priority is not specified, the default for the print service is used, as assigned by the system administrator. A priority limit may be assigned to individual users by the system administrator.

-R

Remove file(s) after submitting the print request. Use this flag with caution.

-r

See **-T content-type [-r]** below.

-s

Suppress the ``request id is ..." message.

-S character-set [-d any]**-S print-wheel [-d any]**

Print this request using the specified *character-set* or *print-wheel*. If a form was requested and it requires a character set or print wheel other than the one specified with the **-S** flag, the request is rejected.

For printers that take print wheels: if the print wheel specified is not one listed by the administrator as acceptable for the printer specified in this request, the request is rejected unless the print wheel is already mounted on the printer.

For printers that use selectable or programmable character sets: if the *character-set* specified is not one defined in the Terminfo database for the printer (see [terminfo](#)), or is not an alias defined by the administrator, the request is rejected.

When the **-d any** flag is used, the request is printed on any printer that has the print wheel mounted or any printer that can select the character set, and that can handle all other needs of the request.

-t title

Print *title* on the banner page of the output. The default is no title. Enclose *title* in quotes if it contains blanks.

-T content-type [-r]

Print the request on a printer that can support the specified *content-type*. If no printer accepts this type directly, a filter will be used to convert the content into an acceptable type. If the **-r** flag is specified, a filter will not be used. If **-r** is specified but no printer accepts the *content-type* directly, the request is rejected. If the *content-type* is not acceptable to any printer, either directly or with a filter, the request is rejected.

In addition to ensuring that no filters will be used, the **-r** flag will force the equivalent of the **-o 'stty=-opost'** flag.

-w

Write a message on the user's terminal after the *files* have been printed. If the user is not logged in, or if the printer resides on a remote system, then mail will be sent instead. Be aware that messages may be sent to a window other than the one in which the command was originally entered.

-y mode-list

Print this request according to the printing modes listed in *mode-list*. The allowed values for *mode-list* are locally defined. This option may be used only if there is a filter available to handle it; otherwise, the print request will be rejected.

The following list describes the *mode-list* options:

-y reverse

Reverse the order in which pages are printed. This filter option is not supported by the LP Print Service.

-y landscape

Change the orientation of a physical page from portrait to landscape.

-y x=*number*,y=*number*

Change the default position of a logical page on a physical page by moving the origin.

-y group=*number*

Group multiple logical pages on a single physical page.

-y magnify=*number*

Change the logical size of each page in a document.

-o length=*number*

Select the number of lines in each page of the document.

-P *number*

Select, by page numbers, a subset of a document to be printed.

-n *number*

Print multiple copies of a document.

Security (System V)

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 (System V)

`/var/spool/lp/*`

lp.cat, lp.set, lp.tell Command

Note: This is a System V Print Subsystem command.

Purpose

Default printer filter used by interface programs.

Syntax

`/usr/lib/lp/bin/lp.cat [-r] [Rate]`

`/usr/lib/lp/bin/lp.set H_pitch V_pitch Width Length Charset`

`/usr/lib/lp/bin/lp.tell Printer`

Description

The **lp.cat** command is the default printer filter called by the interface programs. **lp.cat** reads the file to be printed on its standard input and writes it to the device to be printed on.

lp.cat handles the following signals:

- normal termination (**SIGTERM**)
- serial line hangup (**SIGHUP** due to DCD, Data Carrier Detect, dropping)

- interrupts (**SIGINT** and **SIGQUIT**)
- early pipe termination (**SIGPIPE**)

The **lp.cat** command aborts a printing job if it has to wait too long on output. The default maximum time to wait is calculated as twice the output buffer size (2 * 1024 bytes) divided by the smallest of the values of the transmission rate, print rate, or the specified *Rate* (all rates are in characters per second, CPS). A new maximum delay time may be defined by the *Rate* parameter provided that this increases the delay time. If *Rate* is 0 then the delay allowed is effectively infinite.

When the **-r** flag is specified, **lp.cat** reports the effective throughput in characters per second. This statistic is printed on the standard error after the transmission of every 1024 bytes.

Interface programs may call the **lp.set** command to set the character pitch (*H_pitch*), line pitch (*V_pitch*), page *Width*, page *Length*, and character set (*Charset*) on the printer.

The default units are characters per inch for the character and line pitch, number of columns for width, and number of lines for length. Units may be stated explicitly by appending the values with **c** for centimeters, and **i** for inches.

If it cannot set a particular characteristic, **lp.set** exits with a value of 1 and prints a one letter character code followed by a newline character to the standard error. The character codes are shown in the following table:

Character code	Printer characteristic not set
H	character pitch
L	page length
S	character set
V	line pitch
W	page width

Interface scripts use **lp.tell** to forward descriptions of printer faults to the print service. **lp.tell** sends everything that it reads on its standard input to the print service. The print service forwards the message as an alert to the print administrator.

Flags

Item	Description
-r	Specifies reports the effective throughput in characters per second for the lp.cat command. This statistic is printed on the standard error after the transmission of every 1024 bytes.

Exit Status

The following table shows the possible exit values for **lp.cat**:

Item	Description
0	Normal exit; no error
1	Standard input not defined
2	Standard output not defined
3	Printer type (TERM) not defined or not in terminfo database
4	Standard input and output are identical
5	Write failed; printer may be off-line
6	Excessive delay caused timeout; printer may be off-line

Item	Description
7	Hangup (SIGHUP) detected; loss of carrier
8	Termination (SIGINT or SIGQUIT), or pipe closed early (SIGPIPE)

lp.set returns 0 if successful; otherwise it returns 1 on error.

lp.tell returns:

Item	Description
0	Normal exit; no error
1	Cannot open message queue to the print service
90	Cannot get printer name or key from print service
91	Cannot send message to print service
92	Cannot receive acknowledgment from print service
93	Corrupted acknowledgment received from print service
94	Print service reports message corrupted in transmission

Files

Item	Description
/usr/lib/lp/bin/lp.cat	Full pathname of lp.cat
/usr/lib/lp/bin/lp.set	Full pathname of lp.set
/usr/lib/lp/bin/lp.tell	Full pathname of lp.tell
/etc/lp/model	Printer interface programs directory
/etc/lp/interfaces	Printer interface programs directory

lpac Information

Purpose

Provides general information about protecting the least-privilege (LP) commands resource class and its resources by using access controls that are provided by the resource monitoring and control (RMC) subsystem.

Description

RMC controls access to all of its resources and resource classes through access control lists (ACLs), using two different ACL implementations. The implementation that RMC uses depends on which class is involved. The two major differences between the implementations are in: 1) the mechanisms with which ACLs are viewed and modified and 2) whether ACLs are associated with individual resources.

RMC implements access controls for its resources and resource classes in the following ways:

1. Through ACLs that are defined by resource class stanzas in the `ctrmc.acls` file.

You can view these ACLs by examining the `ctrmc.acls` file. You can modify these ACLs using the `chrmcac1` command. Use a stanza to define an ACL that applies to a class or to all of the resources in a class.

RMC uses this method for all of its resources and resource classes, except for the `IBM.LPCommands` resource class and its resources.

2. Through ACLs that are associated with resources and a resource class within the RMC subsystem.

You can view and modify these ACLs using LP commands. You can define an ACL that applies to a class or an ACL that applies to an individual resource of a class.

RMC uses this method for the IBM.LPCommands resource class and its resources.

This section provides information about ACLs that are specific to the IBM.LPCommands resource class and its resources.

The LP resource manager uses the IBM.LPCommands resource class to define LP resources. These resources represent commands or scripts that require `root` authority to run, but typically the users who need to run these commands do not have `root` authority. By using the LP resource manager commands, users can run commands that require `root` authority. The LP resource manager commands are:

chlpcmd

Changes the read or write attribute values of an LP resource

lphistory

Lists or clears a certain number of LP commands that were previously issued during the current RMC session.

lslpcmd

Lists information about the LP resources on one or more nodes in a domain.

mklpcmd

Defines a new LP resource to RMC and specifies user permissions.

rmlpcmd

Removes one or more LP resources from the RMC subsystem.

runlpcmd

Runs an LP resource.

For descriptions of these commands, see Least-privilege (LP) resource manager commands in *Technical Reference: RSCT for AIX* for AIX and Least-privilege (LP) resource manager commands in *Technical Reference: RSCT for Multiplatforms* for other operating systems. For information about how to use these commands, see the *Administering RSCT* guide.

Because each LP resource can define a unique command, RMC implements ACLs for the IBM.LPCommands class that allows access to be controlled at the individual resource level and at the class level. RSCT provides a set of commands that you can use to list and modify the ACLs for the IBM.LPCommands class and its resources. The LP ACL commands are:

chlpclacl

Changes the Class ACL

chlpracl

Changes the Resource ACL

chlpriacl

Changes the Resource Initial ACL

chlprsacl

Changes the Resource Shared ACL

lslpclacl

Lists the Class ACL

lslptracl

Lists the Resource ACL

lslpriacl

Lists the Resource Initial ACL

lslptracl

Lists the Resource Shared ACL

mklpcmd

Defines a new LP resource to RMC and specifies user permissions

Security

- To use the LP commands that change the Class ACL, the Resource Initial ACL, and the Resource Shared ACL, you must have query and administrator permission for the IBM.LPCCommands class.
- To use the LP command that changes a Resource ACL for an LP resource, you must have query and administrator permission for the LP resource.
- To use the LP commands that list the Class ACL, the Resource Initial ACL, and the Resource Shared ACL, you must have query permission for the IBM.LPCCommands class.
- To use the LP command that lists a Resource ACL for an LP resource, you must have query permission for the LP resource.

The Security section of each LP command description indicates which permissions are required for the command to run properly.

Implementation specifics

This information is part of the Reliable Scalable Cluster Technology (RSCT) filesset.

Location

`/opt/rsct/man/lpacl.7`

Examples

Some examples of how to modify the LP ACLs follow. In these examples, the commands are run on a management server for a group of nodes in a management domain. The management server is named `ms_node` and the managed nodes are called `mc_node1`, `mc_node2`, and so on. In a management domain, it is most likely that the LP resources are defined on the management server and the LP commands themselves are targeted to the managed nodes. In these examples, the Resource Shared ACL is not used because separate permissions are required for the individual LP resources. These examples assume that the LP resources are not yet defined by using the `mklpcmd` command.

1. You want to define the `lpadmin` ID to be the administrator for the LP commands. This ID has the authority to modify the LP ACLs. You also want to give this ID read and write permission to be able to create, delete, and modify the LP resources. To configure this setting, use the `root` mapped identity to run these commands on the management server:

```
chlpclacl lpadmin@LOCALHOST rwa
chlpriacl lpadmin@LOCALHOST rwa
```

These commands define the `lpadmin` ID on the management server as having administrator, read, and write permission for the IBM.LPCCommands class and for the Resource Initial ACL. The Resource Initial ACL is used to initialize a Resource ACL when an LP resource is created. Therefore, when an LP resource is created, the `lpadmin` ID has administrator, read, and write permission to it.

2. The `lpadmin` ID can now create LP resources that define the LP commands that are needed. Access to the LP resources can be defined using the `mklpcmd` command or the `chlpac1` command. When the resource is created, the Resource Initial ACL is copied to the Resource ACL. To modify the Resource ACL using the `chlpac1` command so that `joe` is able to use the `runlpcmd` command for the resource named `SysCmd1`, the `lpadmin` ID runs this command on the management server:

```
chlpac1 SysCmd1 joe@LOCALHOST x
```

This command gives `joe` run permission on the management server to the `SysCmd1` resource so he can use the `runlpcmd` command.

3. In this example, only the `lpadmin` ID has permission to create, delete, and modify LP resources. Use the `chlpclacl` command so that other users can create and delete LP resources. In this case, they need to have write access to the class. To be able to list the resources in the IBM.LPCCommands class, read permission is required. Read permission on a Resource ACL allows a user to view that LP resource. Write permission on a Resource ACL allows a user to modify that LP resource. To allow `joe`

to view the LP resource named SysCmd1, the lpadmin ID runs this command on the management server:

```
ch1prac1 SysCmd1 joe@LOCALHOST r
```

4. There are several nodes in a peer domain. There is an LP resource called SysCmdB1 on nodeB for which joe needs run permission. In addition, joe needs to have run permission from nodes nodeA, nodeB, and nodeD. If you run the ch1prac1 command on nodeB, you can use joe@LOCALHOST for nodeB, but you need to determine the node IDs for nodeA and nodeD. To obtain the node IDs, enter:

```
lsrpnode -i
```

The following output is displayed:

Name	OpState	RSCTVersion	NodeNum	NodeID
nodeA	Online	3.1.0.0	2	48ce221932ae0062
nodeB	Online	3.1.0.0	1	7283cb8de374d123
nodeC	Online	3.1.0.0	4	b3eda8374bc839de
nodeD	Online	3.1.0.0	5	374bdcbe384ed38a
nodeE	Online	3.1.0.0	2	ba74503cea374110
nodeF	Online	3.1.0.0	1	4859dfbd44023e13
nodeG	Online	3.1.0.0	4	68463748bcc7e773

Then, to give joe the permissions as stated earlier, run on nodeB:

```
ch1prac1 SysCmd1 -l joe@LOCALHOST joe@0x48ce221932ae0062 \  
joe@0x374bdcbe384ed38a x
```

Setting or Changing the Printerlpadmin Command

Note: This is a System V Print Subsystem command.

Purpose

Configures the LP print service.

Syntax

Adding or Changing the Configuration of a Local Printer

```
lpadmin -p Printer -v Device [ -D Comment ] [ -A AlertType ] [ -W Minutes ] [ -c Class ] [ -e Printer1 ]  
[ -F FaultRecovery ] [ -f allow:FormList | -f denyFormList ] [ -h ] [ -I Content-Type-List ] [ -i Interface ]  
[ -l ] [ -M -f Form-Name [ -o File-break ] ] [ -M -S Print-Wheel ] [ -m Model ] [ -O Copy-Options ] [ -o  
Print-Options ] [ -o nobanner | -o banner ] [ -r Class ] [ -S List ] [ -s Server-Name [!ServerPrinterName] ]  
[ -T Printer-Type-List ] [ -u allow:Login-Id-List | -u deny:Login-Id-List ]
```

Adding or Changing the Configuration of a Remote Printer

```
lpadmin -p Printer -s ServerName [!ServerPrinterName] -v Device [ -D Comment ] [ -A AlertType ] [ -W  
Minutes ] [ -c Class ] [ -e Printer1 ] [ -F FaultRecovery ] [ -f allow:FormList | -f denyFormList ] [ -h ] [ -I  
Content-Type-List ] [ -i Interface ] [ -l ] [ -M -f Form-Name [ -o Filebreak ] ] [ -M -S Print-Wheel ] [ -m Model ]  
[ -O CopyOptions ] [ -o PrintOptions ] [ -o nobanner | -o banner ] [ -r Class ] [ -S List ] [ -T PrinterTypeList ]  
[ -u allow:LoginIdList | -u deny:LoginIdList ] [ -v Device ]
```

Removing a Printer Destination

```
lpadmin -x Destination
```

Setting or Changing the System Default Destination

```
lpadmin -d [ Destination ]
```

Setting an Alert for a Print Wheel

```
lpadmin -S Print-Wheel -A AlertType [ -W Minutes ] [ -Q Requests ]
```

Description

The **lpadmin** command configures the LP print service by defining printers and devices. It is used to:

- Add and change printers
- Remove printers from the service
- Set or change the system default destination
- Define alerts for printer faults
- Mount print wheels
- Define printers for remote printing services

Printer and class names may be no longer than the maximum length filename allowed for the file system type you are using, and may consist of all printable characters except the space, slash, backslash, colon, semicolon, comma, asterisk, question mark, and tilde. The dash can be used in any position except the first position in a printer name.

If you enter `lpadmin -?`, the system displays the command usage message and returns 0.

Adding or changing a printer

The **-p** *Printer* flag is used to configure a new printer or to change the configuration of an existing printer. When you use this form of the **lpadmin** command, you must select one of the following:

- **inulag -v** *Device*, required to configure a local printer
- **inulag -s** *ServerName* [*!ServerPrinterName*], required to configure a remote printer

Removing a printer destination

The **-x** *dest* flag removes the destination *dest* (a printer or a class), from the LP print service. If *dest* is a printer and is the only member of a class, then the class is deleted. If *dest* is **all**, all printers and classes are removed. No other parameters are allowed with **-x**.

Setting/changing the system default destination

The **-d** [*dest*] flag makes *dest*, an existing printer or class, the new system default destination. If *dest* is not supplied, then there is no system default destination. No other parameters are allowed with **-d**. To unset the system default printer, the user can enter the keyword **none**.

Setting an alert for a print wheel

The **-S** *Print-Wheel* flag is used with the **-A** *Alert-Type* flag to define an alert to mount the print wheel when there are jobs queued for it. If this command is not used to arrange alerting for a print wheel, no alert will be sent for the print wheel. See the other use of **-A** flag, with the **-p**.

The *Alert-Types* are the same as those available with the **-A** flag: **mail**, **write**, **quiet**, **none**, *shell-command*, and **list**. See the description of **-A**, for details about each.

The message sent appears as follows:

```
The print wheel Print-Wheel needs to be mounted
on the printer(s):
printer (integer1 requests)
integer2 print requests await this print wheel.
```

The printers listed are those that the administrator had earlier specified were candidates for this print wheel. The number *integer1* listed next to each printer is the number of requests eligible for the printer. The number *integer2* shown after the printer list is the total number of requests awaiting the print wheel. It will be less than the sum of the other numbers if some requests can be handled by more than one printer.

If the *Print-Wheel* is **all**, the alerting defined in this command applies to all print wheels already defined to have an alert.

If the **-W** flag is not given, the default procedure is that only one message will be sent per need to mount the print wheel. Not specifying the **-W** flag is equivalent to specifying **-W once** or **-W 0**. If *Minutes* is a number greater than zero, an alert is sent at intervals specified by *minutes*.

If the **-Q** flag is also given, the alert is sent when a certain number (specified by the argument *requests*) of print requests that need the print wheel are waiting. If the **-Q** flag is not given, or *requests* is 1 or the word **any** (which are both the default), a message is sent as soon as anyone submits a print request for the print wheel when it is not mounted.

Flags

Item	Description
-A <i>AlertType</i> [-W <i>minutes</i>]	<p>The -A flag defines an alert to inform the administrator when a printer fault is detected, and periodically thereafter, until the printer fault is cleared by the administrator. If an alert is not defined for a particular printer, mail is sent to user lp by default. The <i>AlertTypes</i> are:</p> <p>mail Send the alert message using mail (see mail) to the administrator.</p> <p>write Write the message to the terminal on which the administrator is logged in. If the administrator is logged in on several terminals, one is chosen arbitrarily.</p> <p>quiet Do not send messages for the current condition. An administrator can use this option to temporarily stop receiving further messages about a known problem. Once the fault has been cleared and printing resumes, messages will again be sent when another fault occurs with the printer.</p> <p>none Do not send messages; any existing alert definition for the printer is removed. No alert is sent when the printer faults until a different alert-type (except quiet) is used.</p> <p>shell-command Run the <i>shell-command</i> each time the alert needs to be sent. The shell command should expect the message in standard input. If there are blanks embedded in the command, enclose the command in quotes. The mail and write values for this option are equivalent to the values mail login-ID and write login-ID respectively, where <i>login-ID</i> is the current name for the administrator. This will be the login ID of the person submitting this command unless he or she has used the su command to change to another login ID. If the su command has been used to change the login ID, then the <i>login-ID</i> for the new login is used.</p> <p>list Display the type of the alert for the printer fault. No change is made to the alert.</p>

Item**Description**

The message sent appears as follows:

```
The printer Printer has stopped printing for the reason given
below. Fix the problem and bring the printer back on line.
Printing has stopped, but will be restarted in a few minutes;
issue an enable command if you want to restart sooner.
Unless someone issues a change request

lp -i request-id -P . . .

to change the page list to print, the current request will be
reprinted from the beginning.

The reason(s) it stopped (multiple reasons indicate reprinted
attempts):

reason
```

The LP print service can detect printer faults only through an adequate fast filter and only when the standard interface program or a suitable customized interface program is used. Furthermore, the level of recovery after a fault depends on the capabilities of the filter.

If the *Printer* is **all**, the alerting defined in this command applies to all existing printers.

If the **-W** flag is not used to arrange fault alerting for *Printer*, the default procedure is to mail one message to the administrator of *Printer* per fault. This is equivalent to specifying **-W once** or **-W 0**. If *minutes* is a number greater than zero, an alert is sent at intervals specified by *minutes*.

- c** *Class* Insert *Printer* into the specified *Class*. *Class* is created if it does not already exist.
- d** [*Dest*] Makes *dest*, an existing printer or class, the new system default destination.
- D** *Comment* Saves the *Comment* for display whenever a user asks for a full description of *Printer* (see [lpstat](#)). The LP print service does not interpret this comment.
- e** *Printer1* Copies the interface program of an existing *Printer1* to be the interface program for *Printer*.
 - Note:** Do not specify the **-i** and **-m** flags may not be specified with the **-e** flag.
- f allow:** *FormList* Allows or denies the forms in *FormList* to be printed on *Printer*. By default no forms are allowed on a new printer.
- f deny:** *FormList*

For each printer, the LP print service keeps two lists of forms: an *allow-list* of forms that may be used with the printer, and a *deny-list* of forms that may not be used with the printer. With the **-f allow** flag, the forms listed are added to the allow-list and removed from the deny-list. With the **-f deny** flag, the forms listed are added to the deny-list and removed from the allow-list.

If the allow-list is not empty, only the forms in the list may be used on the printer, regardless of the contents of the deny-list. If the allow-list is empty, but the deny-list is not, the forms in the deny-list may not be used with the printer. All forms can be excluded from a printer by specifying **-f deny:all**. All forms can be used on a printer (provided the printer can handle all the characteristics of each form) by specifying **-f allow:all**.

Item	Description
	<p>The LP print service uses this information as a set of guidelines for determining where a form can be mounted. Administrators, however, are not restricted from mounting a form on any printer. If mounting a form on a particular printer is in disagreement with the information in the allow-list or deny-list, the administrator is warned but the mount is accepted. Nonetheless, if a user attempts to issue a print or change request for a form and printer combination that is in disagreement with the information, the request is accepted only if the form is currently mounted on the printer. If the form is later unmounted before the request can print, the request is canceled and the user is notified by mail.</p> <p>If the administrator tries to specify a form as acceptable for use on a printer that doesn't have the capabilities needed by the form, the command is rejected.</p> <p>The lpadmin command issues a warning when an invalid (nonexistent) form name is submitted with the -f deny: flag.</p> <p>See the other use of -f, with the -M flag.</p>
-F <i>FaultRecovery</i>	<p>Specifies the recovery to be used for any print request that is stopped because of a printer fault, according to the value of <i>FaultRecovery</i>:</p> <p>continue Continue printing on the top of the page where printing stopped. This requires a filter to wait for the fault to clear before automatically continuing.</p> <p>beginning Start printing the request again from the beginning.</p> <p>wait Disable printing on <i>Printer</i> and wait for the administrator or a user to enable printing again.</p> <p>During the wait the administrator or the user who submitted the stopped print request can issue a change request that specifies where printing should resume. (See the -i flag of the lp command.) If no change request is made before printing is enabled, printing will resume at the top of the page where stopped, if the filter allows; otherwise, the request will be printed from the beginning.</p> <p>The default value of <i>FaultRecovery</i> is beginning.</p>
-h	<p>Indicates that the device associated with the printer is hardwired. If neither of the mutually exclusive flags, -h and -l, is specified, this flag is assumed.</p>
-i <i>Interface</i>	<p>Establish a new interface program for <i>Printer</i>. The <i>Interface</i> is the path name of the new program. Do not specify the -e and -m flags with this flag.</p>

Item**Description****-I** *Content-Type-List*

Allow *Printer* to handle print requests with the content types listed in a *Content-Type-List*. If the list includes names of more than one type, the names must be separated by commas or blank spaces. If they are separated by blank spaces, the entire list must be enclosed in double quotes.

The type **simple** is recognized as the default content type. A **simple** type of file is a data stream containing only printable ASCII characters and the following control characters.

Control Character	Octal Value	Meaning
backspace	10	move back one character,
except at beginning of line		
tab	11	move to next tab stop
linefeed (newline)	12	move to beginning of next
line		
form feed	14	move to beginning of next
page		
carriage return	15	move to beginning of
current line		

To prevent the print service from considering **simple** a valid type for the printer, specify either an explicit value (such as the printer type) in the *content-type-list*, or an empty list. If you do want **simple** included along with other types, you must include **simple** in the *content-type-list*.

Except for **simple**, each *content-type* name is freely determined by the administrator. If the printer type is specified by the **-T** option, then the printer type is implicitly considered to be also a valid content type.

-l

Indicates that the device associated with *Printer* is a login terminal. The LP scheduler (**lpsched**) disables all login terminals automatically each time it is started. The **-h** flag may not be specified with this flag.

-M -f *Form-Name*
[-a [-o filebreak]]

Mounts the form *Form-Name* on *Printer*. Print requests that need the pre-printed form *Form-Name* are printed on *Printer*. If more than one printer has the form mounted and the user has specified **any** with the **-d** flag of the **lp** command as the printer destination, then the print request is printed on the one printer that also meets the other needs of the request.

The page length and width, and character and line pitches needed by the form are compared with those allowed for the printer, by checking the capabilities in the *terminfo* database for the type of printer. If the form requires attributes that are not available with the printer, the administrator is warned but the mount is accepted. If the form lists a print wheel as mandatory, but the print wheel mounted on the printer is different, the administrator is also warned but the mount is accepted.

If the **-a** flag is given, an alignment pattern is printed, preceded by the same initialization of the physical printer that precedes a normal print request. Printing is assumed to start at the top of the first page of the form. After the pattern is printed, the administrator can adjust the mounted form in the printer and press return for another alignment pattern (no initialization this time), and can continue printing as many alignment patterns as desired. The administrator can quit the printing of alignment patterns by typing q.

If the **-o filebreak** flag is given, a form feed is inserted between each copy of the alignment pattern. By default, the alignment pattern is assumed to correctly fill a form, so no form feed is added.

A form is unmounted either by mounting a new form in its place or by using the **-f none** flag. By default, a new printer has no form mounted.

Item	Description
	See the other use of -f without the -M .
-M -S Print-Wheel	<p>Mount the <i>Print-Wheel</i> on <i>Printer</i>. Print requests that need the <i>Print-Wheel</i> will be printed on <i>Printer</i>. If more than one printer has <i>Print-Wheel</i> mounted and the user has specified any with the -d flag of the lp command as the printer destination, then the print request is printed on the one printer that also meets the other needs of the request.</p> <p>If the <i>Print-Wheel</i> is not listed as acceptable for the printer, the administrator is warned but the mount is accepted. If the printer does not take print wheels, the command is rejected.</p> <p>A print wheel is unmounted either by mounting a new print wheel in its place or by using the -S none flag. By default, a new printer has no print wheel mounted.</p> <p>See the other uses of the -S flag without the -M.</p>
-m Model	<p>Select <i>Model</i> interface program, provided with the LP print service, for the printer. DO not use the -e and -i flags with this flag. The following interface programs are available:</p> <p>standard generic printer interface</p> <p>PS interface for PostScript printers only</p> <p>By default, the standard interface is used.</p> <p>-O Copy-Option The -O controls whether or not lp makes a copy of the user's file(s) when a print job is submitted. The <i>copy-option</i> can be either copy or nocopy. If -O copy is specified, the LP system always copies the user's source files to the spool area when a print job is submitted. If -O nocopy is specified, the files are copied only if the user specifies the -c flag of lp when submitting the job.</p> <p>This flag sets the value of the copy-files parameter in the /etc/default/lp file. The value, which can be either on or off, is checked every time a print job is submitted.</p>
-o Printing-Option	<p>Specifies the, in the list below the default given to an interface program if the option is not taken from a preprinted form description or is not explicitly given by the user submitting a request (see lp). The only -o options that can have defaults defined are listed below.</p>

```
length=scaled-decimal-number
width=scaled-decimal-number
cpi=scaled-decimal-number
lpi=scaled-decimal-number
stty='stty-option-list'
```

scaled-decimal-number refers to a non-negative number used to indicate a unit of size. The type of unit is shown by a trailing letter attached to the number. Three types of scaled decimal numbers can be used with the LP print service:

- numbers that show sizes in centimeters, marked with a trailing **c**
- numbers that show sizes in inches, marked with a trailing **i**
- numbers that show sizes in units appropriate to use, without a trailing letter

that is, lines, characters, lines per inch, or characters per inch.

Item**Description**

The first four default option values must agree with the capabilities of the type of physical printer, as defined in the *terminfo* database for the printer type. If they do not, the command is rejected.

The *stty-option-list* is not checked for allowed values, but is passed directly to the **stty** program by the standard interface program. Any error messages produced by **stty** when a request is processed (by the standard interface program) are mailed to the user submitting the request.

For each printing option not specified, the defaults for the following attributes are defined in the *terminfo* entry for the specified printer type.

```
length
width
cpi
lpi
```

The default for **stty** is

```
stty='9600 cs8 -cstopb -parenb ixon
-ixany opost -olcuc onlcr -ocrnl -onocr
-onlret -ofill nl0 cr0 tab0 bs0 vt0 ff0'
```

You can set any of the **-o** flags to the default values (which vary for different types of printers), by typing them without assigned values, as follows:

```
length=
width=
cpi=
lpi=
stty=
```

- o nobanner** Allows a user to submit a print request specifying that no banner page be printed.
- o banner** Forces a banner page to be printed with every print request, even when a user asks for no banner page. This is the default; you must specify **-o nobanner** if you want to allow users to be able to specify **-o nobanner** with the **lp** command.
- p Printer** Configures a new printer changes the configuration of an existing printer.
- Q Requests** Specifies that an alert be sent when a certain number of print *Requests* that need the print wheel are waiting.
- r Class** Remove *Printer* from the specified *Class*. If *Printer* is the last member of *Class*, then *Class* is removed.
- s Server-Name** Specifies that you are configuring a remote printer. It makes a server printer accessible to users on your system. *Server-Name* is the name of the system on which the printer is located. It must be listed in the LP systems table. *Server-Printer-Name* is the name used on the server system for that printer. For example, if you want to access *Printer1* on *Server1* and you want it called *Printer2* on your system, enter **-p Printer2 -s Server1!Printer1**.

If *Server-Name* is a Netware server, defined as **-t nuc** using the **lpssystem** command, then *Server-Printer-Name* can be the name of a Netware queue or Netware printer.

Item**Description****-S List**

Allows either the print wheels or aliases for character sets named in *List* to be used on the printer. The **-S** flag does not let you add items to a *List* specified with an earlier invocation of **-S**; instead, it replaces an existing *List* with a new one. Thus **-S** differs from the **-f**, **-u**, **allow**, and **deny** options, which allow you to modify existing lists of available forms and authorized users. Once you've run the **-S** flag, the print wheels and character sets specified, in *List*, on the current command line are the only ones available.

If the printer is a type that takes print wheels, then *List* is a comma or space separated list of print wheel names. Enclose the list with quotes if it contains blanks. These are the only print wheels considered mountable on the printer. You can always force a different print wheel to be mounted, however. Until the flag is used to specify a list, no print wheels is considered mountable on the printer, and print requests that ask for a particular print wheel with this printer is rejected.

If the printer is a type that has selectable character sets, then *List* is a comma or blank separated list of character set name mappings or aliases. Enclose the list with quotes if it contains blanks. Each mapping is of the form:

```
known-name=alias
```

The *known-name* is a character set number preceded by **cs**, such as **cs3** for character set three, or a character set name from the *Terminfo* database entry **csnm**. See [terminfo](#). If this flag is not used to specify a list, only the names already known from the *Terminfo* database or numbers with a prefix of **cs** are acceptable for the printer.

If *List* is the word *none*, any existing print wheel lists or character set aliases is removed.

See the other uses of the **-S** with the **-M** flag.

-T Printer-Type-List

Identify the printer as being of one or more *Printer-Types*. Each *Printer-Type* is used to extract data from the **terminfo** database; this information is used to initialize the printer before printing each user's request. Some filters may also use a *Printer-Type* to convert content for the printer. If this flag is not used, the default *Printer-Type* is unknown; no information will be extracted from **terminfo** so each user request is printed without first initializing the printer. Also, this flag must be used if the following are to work: **-o cpi**, **-o lpi**, **-o width**, and **-o length** flags of the **lpadm** and **lp** commands, and the **-S** and **-f** flags of the **lpadm** command.

If the *Printer-Type-List* contains more than one type, then the *content-type-list* of the **-I** option must either be specified as **simple**, as empty (**-I ""**), or not specified at all.

-u allow:Login-ID-List

Item	Description
-u deny: <i>Login-ID-List</i>	<p>Allows or denies the users in <i>Login-ID-List</i> access to the printer. By default all users on the local system are allowed on a new printer. The <i>Login-ID-List</i> parameter may include any or all of the following constructs:</p> <p>login-ID a user on the local system</p> <p>system-name!login-ID a user on system <i>system-name</i></p> <p>system-name!all all users on system <i>system-name</i></p> <p>all!login-ID a user on all systems</p> <p>all all users on the local system</p> <p>all!all all users on all systems</p> <p>For each printer the LP print service keeps two lists of users:</p> <ul style="list-style-type: none"> • An <i>allow-list</i> of people allowed to use the printer • A <i>deny-list</i> of people denied access to the printer. <p>With the -u allow flag, the users listed are added to the allow-list and removed from the deny-list. With the -u deny flag, the users listed are added to the deny-list and removed from the allow-list.</p> <p>If the allow-list is not empty, only the users in the list may use the printer, regardless of the contents of the deny-list. If the allow-list is empty, but the deny-list is not, the users in the deny-list may not use the printer. All users can be denied access to the printer by specifying -u deny:all. All users may use the printer by specifying -u allow:all.</p>
-v Device	<p>Specifies you are configuring a local printer. It associates a <i>Device</i> with <i>Printer</i>. <i>Device</i> is the path name of a file that is writable by lp. The same <i>Device</i> can be associated with more than one printer.</p>
-x Dest	<p>Removes the destination <i>dest</i> (a printer or a class), from the LP print service.</p>

Notes:

- When creating a new printer, you must specify the **-v**, or **-s** flag. In addition, only one of the following can be supplied: **-e**, **-i**, or **-m**; if none of these three flags is supplied, the model standard is used.
- If you specify the **-s** or **-R** flags, the following flags are not valid: **-A**, **-e**, **-F**, **-h**, **-i**, **-l**, **-M**, **-m**, **-o**, **-v**, and **-W**.
- If you specify the **-J** or **-L** flag, you must specify both flags. The **-p** flag is the only other flag that you can specify with these two 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 that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Files

`/var/spool/lp/*`

`/etc/lp`

lpar_netboot Command

Purpose

Retrieves the media access control (MAC) address and physical location code from network adapters for a partition, or instructs a partition to start the network.

Syntax

To retrieve the MAC address and physical location code:

```
lpar_netboot -M -n [ -v ] [ -x ] [ -f ] [ -i ] [ -E environment [ -E ... ] ] [ -A ] -t ent [ -T { on / off } ] [ -D -s Speed -d Duplex -S Server -G Gateway -C Client [ -K subnetmask ] [ -V vlan_tag ] [ -Y vlan_priority ] ] partition_name partition_profile managed_system
```

To perform a network boot:

```
lpar_netboot [ -v ] [ -x ] [ -f ] [ -i ] [ -E environment [ -E ... ] ] [ -g args ] [ { -A -D | [ -D ] -l physical_location_code | [ -D ] -m MAC-address } ] -t ent [ -T { on / off } ] -s Speed -d Duplex -S Server -G Gateway -C Client [ -K subnetmask ] [ -V vlan_tag ] [ -Y vlan_priority ] [ [ -a -B tftp_image_filename ] -B bootp_image_filename ] partition_name partition_profile managed_system
```

To retrieve the MAC address and physical location code on a system supporting a full system partition:

```
lpar_netboot -M -n [ -v ] [ -x ] [ -f ] [ -i ] [ -E environment [ -E ... ] ] [ -A ] -t ent [ -T { on / off } ] [ -D -s Speed -d Duplex -S Server -G Gateway -C Client [ -K subnetmask ] [ -V vlan_tag ] [ -Y vlan_priority ] ] managed_system managed_system
```

To perform network boot on a system supporting a full system partition:

```
lpar_netboot [ -v ] [ -x ] [ -f ] [ -i ] [ -E environment [ -E ... ] ] [ -g args ] [ { -A -D | [ -D ] -l physical_location_code | [ -D ] -m MAC-address } ] -t ent [ -T { on / off } ] -s Speed -d Duplex -S Server -G Gateway -C Client [ -K subnetmask ] [ -V vlan_tag ] [ -Y vlan_priority ] [ [ -a -B tftp_image_filename ] -B bootp_image_filename ] managed_system managed_system
```

Description

The `lpar_netboot` command instructs a logical partition to network boot, by having the partition send out a bootp request to a server specified with the `-S` flag. The server can be a NIM server that serves SPOT resources, or it can be another server that serves network boot images.

If the `-M` and `-n` flags are specified, the `lpar_netboot` command returns the Media Access Control (MAC) address and the physical location code for a particular type of network adapter that is specified with the `-t` flag. When the `-m` flag is specified, `lpar_netboot` boots a partition, by using a specific network adapter that matches the specified MAC address. When the `-l` flag is specified, `lpar_netboot` boots a partition, by using a specific physical location code for the network adapter that matches the specified physical location code. The matching MAC address or physical location code is dependent upon the hardware resource allocation in the profile in which the partition was booted. The `lpar_netboot` command also requires arguments for partition name, partition profile (which contains the allocated hardware resources), and the name of the managed system in which the partition was defined.

Flags

Item	Description
-A	Returns all adapters of the particular type that are specified with the -t flag.
-a	Specifies the network IP addresses when the server, client, and gateway are IPv6 addresses.
-B <i>Image_filename</i>	Specifies the file name of the network boot image. The -B flag is a required flag for IPv6 addresses.
-C <i>Client</i>	Specifies the IP address of the partition to start the network.
-D	Performs a ping test to identify and use the adapter that can successfully ping the server that is specified with the -S flag.
-d <i>Duplex</i>	Specifies the duplex setting of the partition that is specified with the -C flag. The valid values for the -d flag are <i>full</i> , <i>half</i> , and <i>auto</i> .
-E	Specifies the setting for the environment variable. The following commands return the same output: <pre>-E LPAR_NETBOOT_DEBUG=1</pre> <pre>export LPAR_NETBOOT_DEBUG=1</pre>
-f	Force closes a virtual terminal session for the partition.
-G <i>Gateway</i>	Specifies the gateway IP address of the partition that is specified with the -C flag.
-g <i>args</i>	Specifies generic arguments for starting the partition. You can specify additional arguments with the firmware boot command, by using the -g flag. The -g flag is added for starting the preboot execution environment (PXE). An example for the -g argument follows: <pre>-g autoyast= nfs://9.184.115.219// csminstall/csm/SLES10/09B873DC dhcptimeout=150 install=nfs://9.184.115.219// csminstall/Linux/SLES/10/ppc64/GA/CD1</pre>
-i	Forces an immediate shutdown of the partition. If this option is not specified, a delayed shutdown is performed.
-K <i>subnetmask</i>	Specifies the mask that the gateway uses in determining the appropriate subnetwork for routing. The subnet mask is a set of 4 bytes, as in the IP address. The subnet mask consists of high bits (1s) corresponding to the bit positions of the network and subnetwork address, and low bits (0s) corresponding to the bit positions of the host address.

Item	Description
-l <i>phys_loc</i>	Specifies the physical location code of the network adapter to use for network boot.
-M	Displays the network adapter MAC address and physical location code.
-m <i>address</i>	Specifies the MAC address of the network adapter to use for network boot.
-n	Instructs the partition to not network boot.
-S <i>Server</i>	Specifies the IP address of the partition, from which to retrieve the network boot image during network boot.
-s <i>Speed</i>	Specifies the speed setting of the partition that is specified with the -C flag.
-T	Enables or disables the display of the firmware-spanning tree. The valid values for the -d flag are <i>on</i> and <i>off</i> .
-t <i>ent</i>	Specifies the type of adapter for displaying the MAC address or physical location code discovery, or for network boot. The only valid value for the -t flag is <i>ent</i> for Ethernet.
-V <i>vlan_tag</i>	Specifies the VLAN tag identifier for tagging Ethernet frames during network installation for virtual network communication. The valid values for the -V flag are 0 - 4094.
-v	Displays additional information while the command is running.
-x	Displays debug output while the command is running.
-Y <i>vlan_priority</i>	Specifies the VLAN tag priority for tagging Ethernet frames during network installation for virtual network communication. The valid values for the -Y flag are 0 - 7.

Parameters

Item	Description
<i>partition_name</i>	Specifies the name of the partition.
<i>partition_profile</i>	Specifies the name of the partition profile to use.
<i>managed_system</i>	Specifies the name of the managed system on which the partition is defined.

Environment variables

Item	Description
<i>INSTALLIOS_DEBUG</i>	Prints the lpar_netboot debug output, when specified with the <i>installios</i> command.
<i>LPAR_NETBOOT_DEBUG</i>	Prints the lpar_netboot debug output. Hence, it is similar to the -x flag.

Item	Description
<code>LPAR_NETBOOT_DEBUG_BOOT</code>	Initiates the firmware boot command, when specified with the -s flag.
<code>LPAR_NETBOOT_ADD_TIMEOUT</code>	Extends the timeout value by 5 seconds, as shown in the following example: <pre>LPAR_NETBOOT_ADD_TIMEOUT=5</pre>
<code>LPAR_NETBOOT_SUB_TIMEOUT</code>	Lowers the timeout value by 8 seconds, as shown in the following example: <pre>LPAR_NETBOOT_SUB_TIMEOUT=8</pre>
<code>LPAR_NETBOOT_SPANNING_TREE</code>	Enables or disables the display of the firmware-spanning tree. The valid values for the -d flag are <i>on</i> and <i>off</i> . Hence, it is similar to the -T flag
<code>OPEN_DEV_DEBUG</code>	Displays the firmware open_dev debug output, when the value of the <code>OPEN_DEV_DEBUG</code> variable is set to <i>yes</i> .
<code>FIRMWARE_DUMP</code>	Displays the firmware dump for firmware debugging, when the value of the <code>FIRMWARE_DUMP</code> variable is set to <i>yes</i> .

Exit Status

Item	Description
0	Successful completion

Security

Access Control: You must have root authority to run the `lpar_netboot` command.

Examples

1. To retrieve MAC address and physical location code for partition `machA` with a partition profile `machA_prof` on a managed system `test_sys`, enter the following command:

```
lpar_netboot -M -n -t ent "machA" "machA_prof" "test_sys"
```

2. To network boot partition `machA` with a partition profile `machA_prof` on a managed system `test_sys`, enter the following command:

```
lpar_netboot -t ent -s auto -d auto -S 9.3.6.49 -G 9.3.6.1 -C 9.3.6.234
"machA" "machA_prof" "test_sys"
```

3. To network boot partition `machA` with a specific MAC address of `00:09:6b:dd:02:e8` and a partition profile `machA_prof` on a managed system `test_sys`, enter the following command:

```
lpar_netboot -t ent -m 00096bdd02e8 -s auto -d auto -S 9.3.6.49 -G 9.3.6.1
-C 9.3.6.234 "machA" "machA_prof" "test_sys"
```

4. To network boot partition `machA` with a specific physical location code of `U1234.121.A123456-P1-T6` and a partition profile `machA_prof` on a managed system `test_sys`, enter the following command:

```
lpar_netboot -t ent -l U1234.121.A123456-P1-T6 -s auto -d auto -S 9.3.6.49
-G 9.3.6.1 -C 9.3.6.234 "machA" "machA_prof" "test_sys"
```

- To perform a ping test and a network boot of partition machA with a partition profile machA_prof on a managed system test_sys, enter the following command:

```
lpar_netboot -t ent -D -s auto -d auto -S 9.3.6.49 -G 9.3.6.1 -C 9.3.6.234  
"machA" "machA_prof" "test_sys"
```

- To perform a ping test and a network boot of partition machA with a partition profile machA_prof on a managed system test_sys and disable firmware-spanning tree discovery, enter the following command:

```
lpar_netboot -t ent -T off -D -s auto -d auto -S 9.3.6.49 -G 9.3.6.1  
-C 9.3.6.234 "machA" "machA_prof" "test_sys"
```

Location

/opt/ibm/sysmgmt/dsm/dsmbin/lpar_netboot

lparstat Command

Purpose

Reports logical partition (LPAR) related information and statistics.

Syntax

```
lparstat { -i [ -W | -x | -s | -P | >|-N|< ] | -W | -s | -P | >|-u|< | >|-N|< | -d | -m [ -e [ r | R ]  
[ -p[w] ] ] | [ -H | -h ] | [-X [-o filename ] ] [ -c ] | [-E [w ] ] = [ Interval [ Count ] ] [-L] }
```

Description

The `lparstat` command provides a report of LPAR related information and utilization statistics. This command provides a display of current LPAR related parameters and Hypervisor information, as well as utilization statistics for the LPAR. An interval mechanism retrieves numbers of reports at a certain interval.

The various options of `lparstat` command are exclusive of each other. The `lparstat` command with no options will generate a single report containing utilization statistics related to the LPAR since boot time. If the `-h` option is specified, the report will include summary statistics related to the Hypervisor. If an *interval* and *count* are specified, the above report display repeats for every *interval* seconds and for *count* iterations. *interval* and *count* cannot be used with the `-i` option. Only root users can run the `-h` and `-H` flags.

The *interval* parameter specifies the amount of time in seconds between each report. If you do not specify the *interval* parameter, the **lparstat** 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, its 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.

When the `lparstat` command is invoked without the `-i` flag, two rows of statistics are displayed. The first row displays the System Configuration, which is displayed once when the command starts and again whenever there is a change in the system configuration. The second row contains the Utilization Statistics which will be displayed in intervals and again any time the values of these statistics are deltas from the previous interval.

If you specify the `-X` option, the `lparstat` command creates an XML file.

The following information is displayed in the system configuration row:

type

Indicates the partition type. The value can be either dedicated or shared.

mode

Indicates whether the partition processor capacity is capped or uncapped allowing it to consume idle cycles from the shared pool. Dedicated LPAR is capped or donating.

smt

Indicates whether simultaneous multithreading is enabled or disabled in the partition. If there are two SMT threads, the row is displayed as "on." However, if there are more than two SMT threads, the number of SMT threads is displayed.

lcpu

Indicates the number of online logical processors.

mem

Indicates online memory capacity.

Note: If Active Memory Expansion is enabled, **mem** specifies the expanded memory size configured for this LPAR. However, if the environment variable *AME_MEMVIEW* is set to **TRUE**, the **mem** value specifies the true memory size.

psize

Indicates the number of online physical processors in the pool.

ent

Indicates the entitled processing capacity in processor units. This information is displayed only if the partition type is shared.

If you specify the **-m** flag, the following information is displayed in the system configuration row:

lcpu

Indicates the number of online logical processors.

ent

Indicates the entitled processing capacity in processor units.

mem

Indicates online memory capacity.

Note: If Active Memory Expansion is enabled, **mem** specifies the expanded memory size configured for this LPAR. However, if the environment variable *AME_MEMVIEW* is set to **TRUE**, the **mem** value specifies the true memory size.

mpsz

Indicates the memory pool size of the pool that the partition belongs to (in GB).

iome

Indicates the I/O memory entitlement of the partition (in MB).

iomp

Indicates the number of I/O memory entitlement pools in the LPAR.

If you specify the **-c** flag, the following additional information is displayed in the system configuration row:

mmode

Indicates the system's memory mode. The values for **mmode** are:

Item	Description
Ded	Neither Active Memory Sharing nor Active Memory Expansion is enabled
Shar	Active Memory Sharing is enabled
Ded-E	Active Memory Expansion is enabled
Shar-E	Both Active Memory Sharing and Active Memory Expansion are enabled

mem

Indicates the expanded memory size of the LPAR.

tmem

Indicates the true memory size of the LPAR.

The following information is displayed in the utilization row:

%user

Indicates the percentage of the entitled processing capacity used while executing at the user level (application).

For dedicated partitions, the entitled processing capacity is the number of physical processors.

For uncapped partitions with a current physical processor consumption above their entitled capacity, the percentage becomes relative to the number of physical processor consumed (phpsc).

%sys

Indicates the percentage of the entitled processing capacity used while executing at the system level (kernel).

For dedicated partitions, the entitled processing capacity is the number of physical processors.

For uncapped partitions with a current physical processor consumption above their entitled capacity, the percentage becomes relative to the number of physical processor consumed (phpsc).

%idle

Indicates the percentage of the entitled processing capacity unused while the partition was idle and did not have any outstanding disk I/O request.

For dedicated partitions, the entitled processing capacity is the number of physical processors.

For uncapped partitions with a current physical processor consumption above their entitled capacity, the percentage becomes relative to the number of physical processor consumed (phpsc).

%wait

Indicates the percentage of the entitled processing capacity unused while the partition was idle and had outstanding disk I/O request(s).

For dedicated partitions, the entitled processing capacity is the number of physical processors.

For uncapped partitions with a current physical processor consumption above their entitled capacity, the percentage becomes relative to the number of physical processor consumed (phpsc).

The following statistics are displayed when the partition type is shared or dedicated-donating:

phpsc

Indicates the number of physical processors consumed.

vcswh

Indicates the number of virtual context switches that are virtual-processor hardware preemptions.

The following statistics are displayed only when the partition type is shared:

%entc

Indicates the percentage of the entitled capacity consumed. 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.

lbusy

Indicates the percentage of logical processor(s) utilization that occurred while executing at the user and system level.

app

Indicates the available physical processors in the shared pool.

phint

Indicates the number of phantom (targeted to another shared partition in this pool) interruptions received.

The following statistics are displayed only when the -h flag is specified:

%hypv

Indicates the percentage of physical processor consumption spent making hypervisor calls.

hcalls

Indicates the average number of hypervisor calls that were started.

The following statistic is displayed only if the hardware can use the SPURR, and the processor is not running at nominal speed:

%nsp

Indicates the current average processor speed as a percentage of nominal speed.

The following statistic is displayed only if the turbo-mode accounting is disabled:

%utcyc

Indicates the total percentage of unaccounted turbo cycles.

The following statistics are displayed only when the **-d** flag is specified.

%utuser

Indicates the percentage of unaccounted turbo cycles in the user mode execution (application).

%utsys

Indicates the percentage of unaccounted turbo cycles in the kernel mode execution (kernel).

%utidle

Indicates the percentage of the unaccounted turbo cycles when the partition is idle and does not have any outstanding disk I/O requests.

%utwait

Indicates the percentage of the unaccounted turbo cycles when the partition is idle and has outstanding disk I/O requests.

If you specify the **-m** flag, the following metrics are displayed:

physb

Indicates that the physical processor is busy.

%entc

Indicates the percentage of the entitled capacity consumed. Because the time base over which this data is computed might vary, the entitled capacity percentage can sometimes exceed 100%. This excess is noticeable only with small sampling intervals.

vcsw

Indicates the number of virtual context switches that are virtual-processor hardware preemptions.

hpi

Indicates the number of hypervisor page-ins occurred.

hpit

Indicates the time that is spent waiting for hypervisor page-ins in milliseconds.

pmem

Indicates the physical memory that is allocated to the LPAR by hypervisor in GB.

iomin

Indicates the minimum entitlement of the I/O memory pool in MB.

iomu

Indicates the I/O memory entitlement of the LPAR in use in MB.

iomf

Indicates the free I/O memory entitlement in MB.

iohwm

Indicates the high water mark of I/O memory entitlement usage in GB.

iomaf

Indicates the total number of times that allocation requests for I/O memory entitlement pools have failed since system startup.

If you specify the **-e** flag with the **-m** flag, the following information about I/O memory entitlement pools is displayed:

iompn

Indicates the name of the I/O memory entitlement pool in MB.

iomn

Indicates the minimum entitlement of the I/O memory pool in MB.

iodes

Indicates the desired entitlement of the I/O memory pool in MB.

ioinu

Indicates the entitlement of the I/O memory pool in use in MB.

iores

Indicates the reserved entitlement of the I/O memory pool in MB.

iohwm

Indicates the high water mark of entitlement usage of the I/O memory pool in MB.

ioafl

Indicates the total number of times that allocation requests for this I/O memory entitlement pool have failed since system startup.

The following statistics are displayed only when the `-c` flag is specified:

%xcpu

Indicates the percentage of utilization (relative to the overall CPU consumption by the logical partition) for the Active Memory Expansion (AME) activity.

xphysc

Indicates the number of physical processors used for the Active Memory Expansion activity.

dxm

Indicates the size of the expanded memory deficit for the LPAR in MB.

pgcol

Indicates the logical real memory pages of the calling partition in megabytes that are coalesced during the active memory sharing activity.

mpgcol

Indicates the number of megabytes of the memory pages that are called by the memory pool of the coalesced partition during the Active Memory sharing activity. If the partition is not authorized to access the poolwide statistics, the metric shows zero.

ccol

Indicates the fraction of the CPU consumed in coalescing pages during the Active Memory sharing activity. If the partition is not authorized to access the poolwide statistics, the metric shows zero.

Note: Memory page coalescing is a transparent operation wherein the hypervisor detects duplicate pages, directs all the user read pages to a single copy, and reclaims the other duplicate physical memory pages.

Flags

Item	Description
<code>-c</code>	Adds the memory compression statistics of the LPAR to the default <code>lparstat</code> output. Note: This option is available only when Active Memory Expansion is enabled.
<code>-d</code>	Shows the detailed CPU utilization statistics. When the turbo-mode accounting is disabled, the <code>lparstat</code> command shows the breakdown by category of the unaccounted turbo cycles along with the dedicated, donating or shared utilization columns: %user, %sys, %idle, %wait, %entc, %idon, %bdon, %istol and %bstol.
<code>-e</code>	Displays information about the I/O memory entitlement pools of the LPAR. You can specify the <code>-e</code> flag only with the <code>-m</code> flag. See the metrics that are displayed when you specify the <code>-m</code> flag.
<code>-E</code>	Reports Scaled Processor Utilization Resource Register (SPURR) based utilization metrics if run on a SPURR-capable processor.
<code>-h</code>	Adds summary hypervisor statistics to the default <code>lparstat</code> output.

Item	Description
-H	<p>Provides detailed Hypervisor information. This option basically displays the statistics for each of the Hypervisor calls. The various Hypervisor statistics displayed by this option, for each of the Hypervisor calls, are as below:</p> <p>Statistic</p> <p style="padding-left: 20px;">Description</p> <p>Number of calls Number of Hypervisor calls made.</p> <p>Total Time Spent Percentage of total time spent in this type of call.</p> <p>Hypervisor Time Spent Percentage of Hypervisor time spent in this type of call.</p> <p>Average Call Time Average call time for this type of call in nano-seconds.</p> <p>Maximum Call Time Maximum call time for this type of call in nano-seconds.</p>
-i	<p>Lists details on the LPAR configuration. The various details displayed by the -i option are listed below:</p> <p>Name</p> <p style="padding-left: 20px;">Description</p> <p>Partition Name Logical partition name as assigned at the HMC.</p> <p>Partition Number Number of this logical partition.</p> <p>Power[®] Save Mode Power saving mode of this logical partition.</p> <p>Online Virtual CPUs Number of CPUs (virtual engines) currently online.</p> <p>Maximum Virtual CPUs Maximum possible number of CPUs (virtual engines).</p> <p>Online Memory Amount of memory currently online.</p> <p>Maximum Memory Maximum possible amount of Memory.</p> <p>Type Indicates whether the LPAR is using dedicated or shared CPU resource and if the SMT is turned ON. The Type is displayed in the format [Shared Dedicated] [-SMT] [-#] The following list explains the different Type formats:</p> <ul style="list-style-type: none"> • Shared - Indicates that the LPAR is running in the Shared processor mode. • Dedicated - Indicates that the LPAR is running in the dedicated processor mode. • SMT[-#] - Indicates that the LPAR has SMT mode turned ON and the number of SMT threads is 2. If the number of threads is greater than 2, then the number of threads is also displayed. <p>Mode Indicates whether the LPAR processor capacity is capped or uncapped allowing it to consume idle cycles from the shared pool. Dedicated LPAR is capped or donating.</p> <p>Entitled Capacity The number of processing units this LPAR is entitled to receive.</p> <p>Variable Capacity Weight The priority weight assigned to this LPAR which controls how extra (idle) capacity is allocated to it. A weight of -1 indicates a soft cap is in place.</p> <p>Minimum Capacity The minimum number of processing units this LPAR was defined to ever have. Entitled capacity can be reduced down to this value.</p> <p>Maximum Capacity The maximum number of processing units this LPAR was defined to ever have. Entitled capacity can be increased up to this value.</p> <p>Capacity Increment The granule at which changes to Entitled Capacity can be made. A value in whole multiples indicates a Dedicated LPAR.</p> <p>Maximum Physical CPUs in System The maximum possible number of physical CPUs in the system containing this LPAR.</p>

Item**Description**

(Details displayed by the **-i** flag, are as follows):

Active Physical CPUs in System

The current number of active physical CPUs in the system containing this LPAR.

Active CPUs in Pool

The maximum number of CPUs available to this LPAR's shared processor pool.

Shared Physical CPUs in system

The number of physical CPUs available for use by shared processor LPARs.

Maximum Capacity of Pool

The maximum number of processing units available to this LPAR's shared processor pool.

Entitled Capacity of Pool

The number of processing units that this LPAR's shared processor pool is entitled to receive.

Unallocated Capacity

The sum of the number of processor units unallocated from shared LPARs in an LPAR group. This sum does not include the processor units unallocated from a dedicated LPAR, which can also belong to the group. The unallocated processor units can be allocated to any dedicated LPAR (if it is greater than or equal to 1.0) or shared LPAR of the group.

Physical CPU Percentage

Fractional representation relative to whole physical CPUs that these LPARs virtual CPUs equate to. This is a function of Entitled Capacity / Online CPUs. Dedicated LPARs would have 100% Physical CPU Percentage. A 4-way virtual with Entitled Capacity of 2 processor units would have a 50% physical CPU Percentage.

Minimum Memory

Minimum memory this LPAR was defined to ever have.

Minimum Virtual CPUs

Minimum number of virtual CPUs this LPAR was defined to ever have.

Unallocated Weight

Number of variable processor capacity weight units currently unallocated within the LPAR group.

Partition Group ID

LPAR group that this LPAR is a member of.

Shared Pool ID

Identifier of Shared Pool of Physical processors that this LPAR is a member.

(Details displayed by the **-i** flag, are as follows):

Memory Mode

Indicates whether the memory mode is shared or dedicated. If Active Memory Expansion is enabled, the memory mode also includes a new mode called **Expanded**.

Total I/O memory entitlement

The I/O memory entitlement of the LPAR.

Variable memory capacity weight

The variable memory capacity weight of the LPAR.

Memory Pool ID

The memory pool ID of the pool that the LPAR belongs to.

Physical Memory in the Pool

The physical memory present in the pool that the LPAR belongs to.

Hypervisor Page Size

The page size that hypervisor uses for the page-in and page-out of LPAR logical-memory pages.

Unallocated Variable Memory Capacity Weight

The unallocated variable memory-capacity weight of the LPAR.

Unallocated I/O memory entitlement

The unallocated I/O memory entitlement of the LPAR.

Memory Group ID of LPAR

The memory group ID of the Workload Manager group that the LPAR belongs to.

Target Memory Expansion Factor

The target memory expansion factor configured for the LPAR.

Note: The target memory expansion factor is displayed when Active Memory Expansion is enabled.

Target Memory Expansion Size

The target expanded memory size for the LPAR. The target expanded memory size is the true memory size multiplied by the target memory expansion factor.

Note: The target memory expansion size is displayed when Active Memory Expansion is enabled.

Power Save Mode

The power saving mode for the LPAR.

Subprocessor Mode

The subprocessor mode for the LPAR.

You can specify the **-i** flag alone or with the **-P**, **-W**, **-s**, and **-N** flags.

-L

Displays whether the Live Partition Mobility (LPM) operation has been disabled for the server. The Hardware Management Console (HMC) should be consulted to determine if LPM has been disabled for the logical partition.

Item	Description
-m	<p>Displays the statistics that are related to the following aspects:</p> <ul style="list-style-type: none"> • The logical memory • The physical memory backing the logical memory of the LPAR • The I/O memory entitlement of the LPAR • The memory pool information on the pool that the LPAR belongs to <p>For more information about the metrics that are displayed when you specify the -m flag, see the metrics section.</p>
>> -N	<p>Displays information about the EnergyScale modes of the system. You can specify only the -N flag or you can specify this flag along with the -i, -P, -W, and -s flags.</p> <p>Note: The details that are listed when you run the lparstat -N command might change based on the hardware configuration of the system and new firmware level of the system.</p> <p><<<</p>
-o	Specifies the file name for the XML output.
-p	Displays information about the page coalescing statistics of the LPAR. You can specify the -p flag only with the -m flag. When you run the lparstat command with the -w and -p flags, the result displays all the metrics that are displayed by the -e flag in a single line.
-P	Displays information about the energy management tuning parameters. You can specify the -P flag alone or with the -i , -W , -s , and -N flags.
-r	Resets the high water mark of I/O memory entitlement once at the beginning of the command. You can use this flag only with the -m and -e flags.
-R	Resets the high water mark at the beginning of each monitoring interval. If you specify both the -r and -R flags, the -R flag takes effect.
-s	<p>Displays LPAR information. The -s flag displays the following details:</p> <p>Service partition ID The service partition ID as assigned by the Hardware Management Console (HMC).</p> <p>Number of configured LPARs The number of LPARs that are configured on the HMC.</p> <p>You can specify the -s flag alone or with the -P, -i, -W and -N flags.</p>
-t	Displays the time in the HH:MM:SS format when the command is run with intervals.
-W	<p>Lists details of the workload partition (WPAR) configuration. If the command is run from the global environment, the WPAR Key value is 0. The -W flag displays the following details:</p> <p>WPAR Key WPAR static identifier.</p> <p>WPAR Configured ID WPAR dynamic identifier.</p> <p>WPAR Maximum CPUs Number of processors in a resource set. It displays the value of 0 if it is not restricted.</p> <p>WPAR Effective CPUs Number of processors in an effective resource set. It displays the value of 0 if it is not restricted.</p> <p>WPAR CPU Percentage WPAR processor-limit percentage.</p> <p>You can specify the -W flag alone or with -P, -i, -s, and -N flags.</p>
-x	Lists the security mode settings for the LPAR.
-X	Generates the XML output. The default file name is lparstat_DDMMYYHHMM.xml , unless the user specifies a different file name with the -o option.
>> -u	<p>Displays the expiration date of AIX Update Access Key (UAK) of the system, the expiration date of the Firmware Update Access Key of the system, and the image date of the AIX operating system that is running.</p> <p><<<</p>

Note: If Pool Utilization Authority (PUA) is not available, the app column is not displayed.

Examples

1. To get the default LPAR statistics, enter the following command:

```
lparstat 1 1
```

2. To get default LPAR statistics with summary statistics on Hypervisor, enter the following command:

```
lparstat -h 1 1
```

3. To get the information about the partition, enter the following command:

```
lparstat -i
```

4. To get detailed Hypervisor statistics, enter the following command:

```
lparstat -H 1 1
```

5. To get statistics about the shared memory pool and the I/O memory entitlement of the partition, enter the following command:

```
lparstat -m
```

6. To get statistics about I/O memory pools inside the LPAR, enter the following command:

```
lparstat -me
```

7. If the LPAR is running in shared mode and with 4 SMT threads the type would be in the following format:

```
Type - Shared-SMT-4
```

8. If the LPAR is running in dedicated mode and with 2 SMT threads the type would be in the following format:

```
Type - Dedicated-SMT
```

9. To calculate the memory compression statistics in an LPAR when Active Memory Expansion is enabled, enter the following command:

```
lparstat -c 1 1
```

10. To get statistics about page coalescing inside an LPAR, enter the following command:

```
lparstat -mp
```

11. To check whether you can perform the Live Partition Mobility operations on an LPAR, enter the following command:

```
lparstat -L
```

An output similar to the following example is displayed based the LPM capability of the LPAR:

```
Live Partition Mobility: Enabled
```

Files

Item	Description
/usr/bin/lparstat	Contains the lparstat command.

lpc Command

Note: This is a System V Print Subsystem command.

Purpose

Provides (BSD) line printer control.

Syntax

`/usr/ucb/lpc` [*Command* [*Parameter* . . .]]

Description

The **lpc** command controls the operation of the printer or of multiple printers. The **lpc** command can be used to start or stop a printer, disable or enable a printer's spooling queue, rearrange the order of jobs in a queue, or display the status of each printer, along with its spooling queue and printer daemon.

If you enter `lpc -?`, the system displays the command usage message and returns 0.

With no parameters, the **lpc** command runs interactively, prompting with `lpc>`. If parameters are supplied, the **lpc** command interprets the first as a *Command* to execute; each subsequent parameter is taken as a *Parameter* for that command. The standard input can be redirected so that the **lpc** command reads *Commands* from a file.

Commands may be abbreviated to an unambiguous substring.

Note: The *printer* parameter is specified just by the name of the printer (as **lw**), not as you would specify it to **lpr** or **lpq** (not as **-Plw**).

Item	Description
<code>? [<i>Command</i> . . .]</code>	
<code>help [<i>Command</i> . . .]</code>	Displays a short description of each command specified in the parameter list or, if no parameters are given, a list of the recognized commands.
<code>abort [all [<i>Printer</i> . . .]]</code>	Terminates an active spooling daemon on the local host immediately and then disables printing (preventing new daemons from being started by lpr) for the specified printers. The abort command can only be used by a privileged user.
<code>clean [all [<i>Printer</i> . . .]]</code>	Removes all files created in the spool directory by the daemon from the specified printer queues on the local machine. The clean command can only be used by a privileged user.
<code>disable [all [<i>Printer</i> . . .]]</code>	Turns the specified printer queues off. This prevents new printer jobs from being entered into the queue by lpr . The disable command can only be used by a privileged user.
<code>down [all [<i>Printer</i> . . .]] [<i>Message</i>]</code>	Turns the specified printer queue off, disables printing, and puts <i>Message</i> in the printer status file. The message does not need to be quoted. The remaining parameters are treated like echo . This is normally used to take a printer down and let others know why (lpq indicates that the printer is down, as does the status command).
<code>enable [all [<i>Printer</i> . . .]]</code>	Enables spooling on the local queue for the listed printers so that lpr can put new jobs in the spool queue. The enable command can only be used by a privileged user.
<code>exit</code>	Exits from lpc .
<code>quit</code>	Quits from lpc .

Item	Description
restart [all [Printer....]]	Attempts to start a new printer daemon. This is useful when some abnormal condition causes the daemon to die unexpectedly leaving jobs in the queue. This command can be run by any user.
start [all [Printer...]]	Enables printing and starts a spooling daemon for the listed printers. The start command can only be used by a privileged user.
status [all [Printer...].]	Displays the status of daemons and queues on the local machine. This command can be run by any user.
stop [all [Printer...]]	Stops a spooling daemon after the current job completes and disable printing. The stop command can only be used by a privileged user.
topq Printer [Job#...] [User...]	Moves the print jobs specified by <i>Job#</i> or those jobs belonging to <i>User</i> to the top (head) of the printer queue. The topq command can only be used by a privileged user.
up [all [Printer...]]	Enables everything and starts a new printer daemon. Undoes the effects of down .

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
/var/spool/lp/*	
/var/spool/lp/system/pstatus	

Error Codes

Item	Description
?Ambiguous command	The abbreviation matches more than one command.
?Invalid command	A command or abbreviation is not recognized.
?Privileged command	The command can be executed only by the privileged user.
lpc: printer: unknown printer to the print service	The <i>printer</i> was not found in the System V LP database. Usually this is a typing mistake; however, it may indicate that the printer does not exist on the system. Use lptstat -p to find the reason.

Item	Description
lpc: error on opening queue to spooler	The connection to lpssched on the local machine failed. This usually means the printer server started at boot time has died or is hung. Check if the printer spooler daemon /usr/lib/lp/lpsched is running.
lpc: Can't send message to LP print service	
lpc: Can't receive message from LP print service	Indicates that the LP print service has been stopped. Get help from the system administrator.
lpc: Received unexpected message from LP print service	It is likely there is an error in this software. Get help from system administrator.

lpd Command

Purpose

Provides the remote print server on a network.

Syntax

```
lpd [ -d ] [ -l ] [ -D DebugOutputFile]
```

Description

The **lpd** daemon is the remote print server. It monitors port 515 for print requests. Each request is placed in a directory named **/var/spool/lpd**.

A computer on a network (host) that can create a Transmission Control Protocol/Internet Protocol (TCP/IP) data stream and use the **lpd** protocol can print remotely or act as a print server. As a security feature, the **lpd** daemon accepts print requests only from remote hosts that are listed in the local **/etc/hosts.equiv** or **/etc/hosts.lpd** file.

The **lpd** daemon can run on any host in the network; its function is to accept print requests from foreign hosts (on port 515). The **lpd** daemon handles each request by forking a child process. Remote requests are first checked against the **/etc/hosts.equiv** and **/etc/hosts.lpd** files for permission to print on the local host.

Changes can be made to the **/etc/hosts.equiv** and **/etc/hosts.lpd** files without restarting the system. To put changes to these files into effect without restarting the system, use the System Resource Controller (SRC) **refresh** command. This command causes the **/etc/hosts.equiv** and **/etc/hosts.lpd** database files to be reloaded and the changes implemented.

Note: The queuing system does not support multibyte host names.

The **/etc/locks/lpd** file contains the process ID of the currently running instance of the **lpd** daemon. If the current machine becomes inoperable, you may need to remove the ID for the **lpd** daemon when the system starts up again. The error message displayed is **lpd: lock file or duplicate daemon**.

Manipulating the lpd Daemon with the System Resource Controller

The **lpd** daemon is a subsystem controlled by the System Resource Controller (SRC). The **lpd** daemon is a member of the TCP/IP system group.

Use the following SRC commands to manipulate the **lpd** daemon:

Item	Description
startsrc	Starts a subsystem, group of subsystems, or a subserver.

Item	Description
stopsrc	Stops a subsystem, group of subsystems, or a subserver.
refresh	Causes the subsystem or group of subsystems to reread the appropriate configuration file.
traceson	Enables tracing of a subsystem, group of subsystems, or a subserver.
tracesoff	Disables tracing of a subsystem, group of subsystems, or a subserver.
lssrc	Gets the status of a subsystem, group of subsystems, or a subserver.

Flags

Item	Description
-d	Sends a status of Inactive to be logged with the SRC controller and sends error messages during socket communication setup failures to the user display.
-l	Sends a status of Active to be logged with the SRC controller and sends valid or invalid job request messages to the user display.
-D <i>DebugOutputFile</i>	Sends extensive debugging output used for problem determination to the file specified by <i>DebugOutputFile</i> . This should only be used during problem determination as the <i>DebugOutputFile</i> can grow large rapidly. If the output file specified already exists, new debugging output is appended to the end of it. If there are any problems creating or writing to the output file, the debugging option is ignored.

Examples

1. To start the **lpd** server daemon, enter:

```
startsrc -s lpd
```

2. To start the **lpd** server daemon while enabling the display of certain error messages, enter:

```
startsrc -s lpd -a " -d"
```

3. To send logging information to the **stderr** daemon, enter:

```
startsrc -s lpd -a " -l"
```

4. To start the **lpd** server daemon in debugging mode with output going to **/tmp/dbglpd.out**, enter:

```
startsrc -s lpd -a " -D /tmp/dbglpd.out"
```

Files

Item	Description
/usr/sbin/lpd	Specifies the path to the lpd daemon.
/dev/lp*	Contains the names of print devices.
/etc/hosts.equiv	Contains the names of hosts allowed to execute commands and print.
/etc/hosts.lpd	Contains the names of hosts allowed to print only.
/var/spool/lpd	Contains the spool directory for control, status, and data files.

Item	Description
<code>/etc/locks/lpd</code>	Contains the PID of the currently running lpd daemon. After a system crash, this PID may need to be deleted. The following error message indicates the problem:
	<pre>lpd: lock file or duplicate daemon</pre>

lpfilter Command

Note: This is a System V Print Subsystem command.

Purpose

Administers filters used with the LP print service.

Syntax

lpfilter **-f** *FilterName* **-F** *PathName*

lpfilter **-f** *FilterName* **-**

lpfilter **-f** *FilterName* **-i**

lpfilter **-f** *FilterName* **-x**

lpfilter **-f** *FilterName* **-l**

Description

The **lpfilter** command is used to add, change, delete, and list a filter used with the LP print service. These filters are used to convert the content type of a file to a content type acceptable to a printer.

If you enter `lpfilter -?`, the system displays the command usage message and returns 0.

Flags

Item	Description
- (hyphen)	Adds or changes a filter as specified from standard input.
-f <i>FilterName</i>	Specifies the name of the filter to be added, changed, deleted, or listed.
-F <i>PathName</i>	Add or changes a filter as specified by the contents of the file pathname.
-i	Resets an original filter to its original settings.
-l	Lists a filter description.
-x	Deletes a filter.

The parameter **all** can be used instead of a *FilterName* with any of these flags. When **all** is specified with the **-F** or **-** flag, the requested change is made to all filters. Using **all** with the **-i** flag has the effect of restoring to their original settings all filters for which predefined settings were initially available. Using the **all** parameter with the **-x** flag results in all filters being deleted, and using it with the **-l** flag produces a list of all filters.

Adding or changing a filter

The filter named in the **-f** flag is added to the filter table. If the filter already exists, its description is changed to reflect the new information in the input.

The filter description is taken from the *PathName* if the **-F** flag is given or from the standard input if the **-** flag is specified. One of the two must be given to define or change a filter. If the filter named is one originally delivered with the LPprint service, the **-i** flag restores the original filter description.

When an existing filter is changed with the **-F** flag or the **-** flag, items that are not specified in the new information are left as they were. When a new filter is added with this command, unspecified items are given default values.

Filters are used to convert the content of a request into a data stream acceptable to a printer. For a given print request, the LP print service knows the following:

- Content in the request
- Name of the printer
- Type of the printer
- Types of content acceptable to the printer
- Modes of printing asked for by the originator of the request

It uses this information to find a filter or a pipeline of filters that converts the content into a type acceptable to the printer.

A list of items that provide input to this command and a description of each item follows. All lists are comma or space separated.

- Input types: *content-type-list*
- Output types: *content-type-list*
- Printer types: *printer-type-list*
- Printers: *printer-list*
- Filter type: *filter-type*
- Command: *shell-command*
- Flags: *template-list*

Item	Description
Input types	Gives the types of content that can be accepted by the filter. (The default is any.)
Output types	Gives the types of content that the filter can produce from any of the input content types. (The default is any.)
Printer types	Gives the type of printers for which the filter can be used. The LP print service restricts the use of the filter to these types of printers. (The default is any.)
Printers	Gives the names of the printers for which the filter can be used. The LP print service restricts the use of the filter to just the printers named. (The default is any .)
Filter type	Marks the filter as a slow filter or a fast filter. Slow filters are generally those that take a long time to convert their input. They are run unconnected to a printer to keep the printers from being tied up while the filter is running. If a listed printer is on a remote system, the filter type for it must have the value slow . Fast filters are generally those that convert their input quickly or those that must be connected to the printer when run. These are given to the interface program IP to run connected to the physical printer.
Command	Specifies the program to run to invoke the filter. The full program pathname as well as fixed flags must be included in the <i>shell-command</i> ; additional flags are constructed, based on the characteristics of each print request and on the "flags" field. A command must be given for each filter. The command must accept a data stream as standard input and produce the converted data stream on its standard output. This allows filter pipelines to be constructed to convert data not handled by a single filter.

Item **Description**

Flags

Specifies the comma-separated list of templates used by the LP print service to construct flags to the filter from the characteristics of each print request listed in the table later.

In general, each template is of the following form:

keyword-pattern=replacement

The *keyword* names the characteristic that the template attempts to map into a filter-specific flag; each valid *keyword* is listed in the table below. A *pattern* is one of the following: a literal pattern of one of the forms listed in the table, a single asterisk (*), or a regular expression. If *pattern* matches the value of the characteristic, the template fits and is used to generate a filter-specific flag. The *replacement* is what is used as the flag.

Regular expressions are the same as those found in the **ed** or **vi** commands. This includes the `\(. . . \)` and `\n` constructions, which can be used to extract portions of the *pattern* for copying into the *replacement*, and the `` ` &'`, which can be used to copy the entire *pattern* into the *replacement*.

The *replacement* can also contain a `` ` *'`. It too, is replaced with the entire *pattern*, just like the `` ` &'` of the **ed** command.

lp flag	Properties
-T	<p>Characteristic Content type (input)</p> <p>Keyword INPUT</p> <p>Possible patterns content-type</p>
N/A	<p>Characteristic Content type (output)</p> <p>Keyword OUTPUT</p> <p>Possible patterns content-type</p>
N/A	<p>Characteristic Printer type</p> <p>Keyword TERM</p> <p>Possible patterns printer-type</p>
-d	<p>Characteristic Printer name</p> <p>Keyword PRINTER</p> <p>Possible patterns printer-name</p>

lp flag	Properties
-f, -o cpi=	<p>Characteristic Character pitch</p> <p>Keyword CPI</p> <p>Possible patterns integer</p>
-f, -o lpi=	<p>Characteristic Line pitch</p> <p>Keyword LPI</p> <p>Possible patterns integer</p>
-f, -o length=	<p>Characteristic Page length</p> <p>Keyword LENGTH</p> <p>Possible patterns integer</p>
-f, -o width=	<p>Characteristic Page width</p> <p>Keyword WIDTH</p> <p>Possible patterns integer</p>
-P	<p>Characteristic Pages to print</p> <p>Keyword PAGES</p> <p>Possible patterns page-list</p>
-S	<p>Characteristic Character set Print wheel</p> <p>Keyword CHARSET CHARSET</p> <p>Possible patterns character-set-name print-wheel-name</p>
-f	<p>Characteristic Form name</p> <p>Keyword FORM</p> <p>Possible patterns form-name</p>

lp flag	Properties
-y	<p>Characteristic Modes</p> <p>Keyword MODES</p> <p>Possible patterns mode</p>
-n	<p>Characteristic Number of copies</p> <p>Keyword COPIES</p> <p>Possible patterns integer</p>

For example, the template `MODES landscape = -1` shows that if a print request is submitted with the **-y landscape** flag, the filter is given the flag **-l**. As another example, the template `TERM * = -T *` shows that the filter is given the flag **-T printer-type** for whichever *printer-type* is associated with a print request using the filter.

As a last example, consider the template `MODES prwidth\=(.*) = -w\1`. Suppose a user gives the command **lp -y prwidth=10**

From the table above, the LP print service determines that the **-y** flag is handled by a **MODES** template. The **MODES** template here works because the pattern `prwidth\=(.*)` matches the **prwidth=10** given by the user. The *replacement* `-w\1` causes the LP print service to generate the filter flag **-w10**.

If necessary, the LP print service constructs a filter pipeline by concatenating several filters to handle the user's file and all the print flags. If the print service constructs a filter pipeline, the **INPUT** and **OUTPUT** values used for each filter in the pipeline are the types of the input and output for that filter, not for the entire pipeline.

Deleting a filter

The **-x** flag is used to delete the filter specified in *FilterName* from the LP filter table.

Listing a filter description

The **-l** flag is used to list the description of the filter named in *FilterName*. If the command is successful, the following message is sent to standard output:

```
Input types: content-type-list
Output types: content-type-list
Printer types: printer-type-list
Printers: printer-list
Filter type: filter-type
Command: shell-command
flags: template-list
```

If the command fails, an error message is sent to standard 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.

lpforms Command

Note: This is a System V Print Subsystem command.

Purpose

Administer forms used with the LP print service.

Syntax

lpforms -f *FormName* *Options*

lpforms -f *FormName* **-A** *AlertType* [**-Q** *minutes*] [**-W** *requests*]

Description

The **lpforms** command is used to administer the use of preprinted forms, such as company letterhead paper, with the LP print service. A form is specified by its *FormName*. Users may specify a form when submitting a print request. The parameter **all** can be used instead of *FormName* with either of the command lines shown above. The first command line allows the administrator to add, change, and delete forms, to list the attributes of an existing form, and to allow and deny users access to particular forms. The second command line is used to establish the method by which the administrator is alerted that the form *FormName* must be mounted on a printer.

If you enter `lpforms -?`, the system displays the command usage message and returns 0.

With the first **lpforms** command line, one of the following flags must be used:

Flags

Item	Description
- (hyphen)	Adds or changes form <i>FormName</i> , as specified by the information from standard input.
-F <i>pathname</i>	Adds or changes form <i>FormName</i> , as specified by the information in <i>pathname</i> .
-l	Lists the attributes of form <i>FormName</i> .
-x	Deletes form <i>FormName</i> (this flag must be used separately; it may not be used with any other flag).

Adding or changing a form

The **-F** *pathname* flag is used to add a new form, *FormName*, to the LP print service, or to change the attributes of an existing form. The form description is taken from *pathname* if the **-F** flag is given, or from the standard input if the **-** flag is used. One of these two flags must be used to define or change a form. *pathname* is the pathname of a file that contains all or any subset of the following information about the form:

```
Page length: scaled-decimal-number1
Page width: scaled-decimal-number2
Number of pages: integer
Line pitch: scaled-decimal-number3
Character pitch: scaled-decimal-number4
Character set choice: character-set/print-wheel [mandatory]
Ribbon color: ribbon-color
Comment:
comment
Alignment pattern: [content-type]
content
```

The term "scaled-decimal-number" refers to a non-negative number used to indicate a unit of size. The type of unit is shown by a "trailing" letter attached to the number. Three types of scaled decimal numbers can be used with the LP print service: numbers that show sizes in centimeters (marked with a trailing "c"); numbers that show sizes in inches (marked with a trailing "i"); and numbers that show sizes in units appropriate to use (without a trailing letter), that is, lines, characters, lines per inch, or characters per inch.

Except for the last two lines, the above lines may appear in any order. The `Comment` and `comment` items must appear in consecutive order but may appear before the other items, and the "Alignment pattern" and the `content` items must appear in consecutive order at the end of the file. Also, the `comment` item may not contain a line that begins with any of the key phrases above, unless the key phrase is preceded with a ">". Any leading ">" sign found in the `comment` are removed when the comment is displayed. Case distinctions in the key phrases are ignored.

When this command is issued, the form specified by `FormName` is added to the list of forms. If the form already exists, its description is changed to reflect the new information. Once added, a form is available for use in a print request, except where access to the form has been restricted, as described under the `-u` flag. A form may also be allowed to be used on certain printers only.

A description of each form attribute is below:

Item	Description
Page length and Page width	<p>Before printing the content of a print request needing this form, the generic interface program provided with the LP print service initializes the physical printer to handle pages <i>scaled-decimal-number1</i> long, and <i>scaled-decimal-number2</i> wide using the printer type as a key into the <i>terminfo</i> database.</p> <p>The page length and page width are also passed, if possible, to each filter used in a request needing this form.</p>
Number of pages	<p>Each time the alignment pattern is printed, the LP print service attempts to truncate the <i>content</i> to a single form by, if possible, passing to each filter the page subset of <i>1-integer</i>.</p>
Line pitch and Character pitch	<p>Before printing the content of a print request needing this form, the interface programs provided with the LP print service initializes the physical printer to handle these pitches, using the printer type as a key into the <i>terminfo</i> database. Also, the pitches are passed, if possible, to each filter used in a request needing this form. <i>scaled-decimal-number3</i> is in lines per centimeter if a "c" is appended, and lines per inch otherwise; similarly, <i>scaled-decimal-number4</i> is in characters per centimeter if a "c" is appended, and characters per inch otherwise. The character pitch can also be given as elite (12 characters per inch), pica (10 characters per inch), or compressed (as many characters per inch as possible).</p>

Item	Description
Character set choice	When the LP print service alerts an administrator to mount this form, it also mentions that the print wheel <i>print-wheel</i> should be used on those printers that take print wheels. If printing with this form is to be done on a printer that has selectable or loadable character sets instead of print wheels, the interface programs provided with the LP print service automatically selects or loads the correct character set. If mandatory is appended, a user is not allowed to select a different character set for use with the form; otherwise, the character set or print wheel named is a suggestion and a default only.
Ribbon color	When the LP print service alerts an administrator to mount this form, it also mentions that the color of the ribbon should be <i>ribbon-color</i> .
Comment	The LP print service displays the <i>comment</i> unaltered when a user asks about this form.
Alignment pattern	When mounting this form, an administrator can ask for the <i>content</i> to be printed repeatedly, as an aid in correctly positioning the preprinted form. The optional <i>content-type</i> defines the type of printer for which <i>content</i> had been generated. If <i>content-type</i> is not given, simple is assumed.

Note: The content is stored as given and is readable only by the user *lp*.

When an existing form is changed with this command, items missing in the new information are left as they were. When a new form is added with this command, missing items gets the following defaults:

```
Page Length: 66
Page Width: 80
Number of Pages: 1
Line Pitch: 6
Character Pitch: 10
Character Set Choice: any
Ribbon Color: any
```

Deleting a form

The **-x** flag is used to delete the form *FormName* from the LP print service.

Listing form attributes

The **-l** flag is used to list the attributes of the existing form *FormName*. Because of the potentially sensitive nature of the alignment pattern, only the administrator can examine the form with this command. Other people may use the **lpstat** command to examine the non-sensitive part of the form description.

Allowing and denying access to a form

The **-u** flag, followed by the parameter **allow:login-ID-list** or **-u deny:login-ID-list** lets you determine which users are allowed to specify a particular form with a print request. This flag can be used with the **-F** or **-f** flag.

The *login-ID-list* parameter may include any or all of the following constructs:

Item	Description
<i>login-ID</i>	A user on the local system

Item	Description
<i>system-name</i> ! <i>login-ID</i>	A user on system <i>system-name</i>
<i>system-name</i> ! all	All users on system <i>system-name</i>
all ! <i>login-ID</i>	A user on all systems
all	All users on the local system
all ! all	All users on all systems

The default value of *login-ID-list* is **all**.

The LP print service keeps two lists of users for each form: an "allow-list" of people allowed to use the form, and a "deny-list" of people that may not use the form.

- If allow-list is present and *login-ID* is in it, access is allowed.
- If only deny-list is present and *login-ID* is not in it, access is allowed.
- If *login-ID* is in deny-list, access is denied.
- If neither allow-list or deny-list are present, access is denied.
- If both lists are present, and *login-ID* is in neither, access is denied.
- If only allow-list is present and *login-ID* is not in it, access is denied.

If the allow-list is not empty, only the users in the list are allowed access to the form, regardless of the contents of the deny-list. If the allow-list is empty but the deny-list is not, the users in the deny-list may not use the form (but all others may use it).

All users can be denied access to a form by specifying **-f deny:all**. All users can be allowed access to a form by specifying **-f allow:all**. (This is the default.)

Setting an alert to mount a form

The **-f FormName** flag is used with the **-A AlertType** flag to define an alert to mount the form when there are queued jobs which need it. If this flag is not used to arrange alerting for a form, no alert is sent for that form.

The method by which the alert is sent depends on the value of the *AlertType* parameter specified with the **-A** flag. The alert types are the same as those available with the **-A** flag to **lpadmin**: **mail**, **write**, **quiet**, **none**, *shell-command*, and **list**.

The message sent appears as follows:

```
The form FormName needs to be mounted
on the printer(s):
printer (integer1 requests).
integer2 print requests await this form.
Use the ribbon-color ribbon.
Use the print-wheel print wheel, if appropriate.
```

The printers listed are those that the administrator had earlier specified were candidates for this form. The number *integer1* listed next to each printer is the number of requests eligible for the printer. The number *integer2* shown after the list of printers is the total number of requests awaiting the form. It is less than the sum of the other numbers if some requests can be handled by more than one printer. The *ribbon-color* and *print-wheel* are those specified in the form description. The last line in the message is always sent, even if none of the printers listed use print wheels, because the administrator may choose to mount the form on a printer that does use a print wheel.

Where any color ribbon or any print wheel can be used, the statements above read:

```
Use any ribbon.
Use any print-wheel.
```

If *FormName* is **any**, the alerting defined in this command applies to any form for which an alert has not yet been defined. If *FormName* is **all**, the alerting defined in this command applies to all forms.

If the **-W** flag is not given, the default procedure is that only one message is sent per need to mount the form. Not specifying the **-W** flag is equivalent to specifying **-W once** or **-W 0**. If *minutes* is a number greater than 0, an alert is sent at intervals specified by *minutes*.

If the **-Q** flag is also given, the alert is sent when a certain number (specified by the parameter *requests*) of print requests that need the form are waiting. If the **-Q** flag is not given, or the value of *requests* is 1 or **any** (which are both the default), a message is sent as soon as anyone submits a print request for the form when it is not mounted.

Listing the current alert

The **-f** flag, followed by the **-A** flag and the parameter **list** is used to list the type of alert that has been defined for the specified form *FormName*. No change is made to the alert. If *FormName* is recognized by the LP print service, one of the following lines is sent to the standard output, depending on the type of alert for the form.

```
When requests requests are queued:
alert with shell-command every minutes minutes

When requests requests are queued:
write to user-name every minutes minutes

When requests requests are queued:
mail to user-name every minutes minutes

No alert
```

The phrase "every *minutes* minutes" is replaced with "once" if *minutes* (**-W minutes**) is 0.

Terminating an active alert

The **-A quiet** flag is used to stop messages for the current condition. An administrator can use this flag to temporarily stop receiving further messages about a known problem. Once the form has been mounted and then unmounted, messages are again sent when the number of print requests reaches the threshold *requests*.

Removing an alert definition

No messages are sent after the **-A none** flag is used until the **-A**flag is given again with a different *AlertType*. This can be used to permanently stop further messages from being sent as any existing alert definition for the form is 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 that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

lphistory Command

Purpose

Displays or clears the history list of least-privilege (LP) commands that have been run during the current resource monitoring and control (RMC) session.

Syntax

- To list a particular number of previously-issued commands:
 - On the local node:

```
lphistory [ -u user_ID ] [ -m mapped_ID ] [ -C command_name ] [ -S command_path ] [ -B MMddhhmmyyyy ] [ -E MMddhhmmyyyy ] [ -L a | c | e | m | n | t | u | x ] [ -h ] [ -TV ] [ num_records ]
```

- On all nodes in a domain:

```
lphistory -a [ -u user_ID ] [ -m mapped_ID ] [ -C command_name ] [ -S command_path ] [ -B MMddhhmmyyyy ] [ -E MMddhhmmyyyy ] [ -L a | c | e | m | n | t | u | x ] [ -h ] [ -TV ] [ num_records ]
```

- On a subset of nodes in a domain:

```
lphistory -n host1[,host2...] [ -u user_ID ] [ -m mapped_ID ] [ -C command_name ] [ -S command_path ] [ -B MMddhhmmyyyy ] [ -E MMddhhmmyyyy ] [ -L a | c | e | m | n | t | u | x ] [ -h ] [ -TV ] [ num_records ]
```

- To clear the history list:

- On the local node:

```
lphistory -c [ -u user_ID ] [ -m mapped_ID ] [ -C command_name ] [ -S command_path ] [ -B MMddhhmmyyyy ] [ -E MMddhhmmyyyy ] [ -h ] [ -TV ]
```

- On all nodes in a domain:

```
lphistory -c -a [ -u user_ID ] [ -m mapped_ID ] [ -C command_name ] [ -S command_path ] [ -B MMddhhmmyyyy ] [ -E MMddhhmmyyyy ] [ -h ] [ -TV ]
```

- On a subset of nodes in a domain:

```
lphistory -c -n host1[,host2...] [ -u user_ID ] [ -m mapped_ID ] [ -C command_name ] [ -S command_path ] [ -B MMddhhmmyyyy ] [ -E MMddhhmmyyyy ] [ -h ] [ -TV ]
```

Description

The **lphistory** command lists the history of LP commands that have been run by the least-privilege resource manager. The command history is maintained as records in the RSCT audit log. By default, only the command string (the path name plus arguments) from each audit log record is listed. The **-L** flag controls the output format of **lphistory**; use it to display specific fields as needed. The selection flags (**-B**, **-C**, **-E**, **-m**, **-S**, or **-u**) control the selection string that is passed to **lsaudrec**.

The **lphistory** command takes one optional parameter: the number of records to list. The default value of *num_records* is 10. If none of the selection flags is used, the latest number of records in the audit log (specified by *num_records*) are listed. Otherwise, the latest number of records (specified by *num_records*) from those selected by one or more of the selection flags are listed. This selection process applies to the audit records on each node specified by the **-a** flag or the **-n** flag. If neither **-a** nor **-n** is specified, the selection process applies to the audit records on the local node.

The **-B** and **-E** flags take time stamps as arguments. Time stamps are in the form *MMddhhmmyyyy*, where *MM* is the two-digit month (01-12), *dd* is the two-digit day of the month (01-31), *hh* is the two-digit hour (00-23), *mm* is the two-digit minute (00-59), and *yyyy* is the four-digit year.

You can use the wild card character (%) with identity-related arguments (*user_ID*, *mapped_ID*) and command names. The % can be placed at the beginning or end of the string, or anywhere within it. You cannot use any wild card characters when specifying *command_path*.

You can remove audit log records using the **-c** flag. If none of the selection flags is specified, all audit log records for the least-privilege resource manager are removed. Otherwise, the records selected by one or more of the selection flags are removed. The **-c** flag cannot be used with the **-L** flag or the *num_records* parameter.

Flags

-a

Displays previously-issued LP commands for all nodes in the domain.

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.

You cannot specify this flag with the **-n** flag.

-B MMddhhmmyyyy

Specifies a beginning time stamp in the form *MMddhhmmyyyy*, where *MM* is the two-digit month (01-12), *dd* is the two-digit day (01-31), *hh* is the two-digit hour (00-23), *mm* is the two-digit minute (00-59), and *yyyy* is the four-digit year. The time can be truncated from right to left, except for *MM*. If not all digits are specified, the year defaults to the current year, minutes to 0, hour to 0, and day to 01. At a minimum, the month must be specified. The command lists or removes only those records that were created at or after this time.

-c

Clears the history of LP commands. You cannot specify this flag with the *number_of_commands* parameter or the **-n** flag.

-C command_name

Specifies a command name. **lphistory -C** lists or removes only those records that contain *command_name*, which is the name of a command without a fully-qualified path (**mkrsrc**, for example). You can use wild card characters in *command_name*.

-E MMddhhmmyyyy

Specifies an ending time stamp in the form *MMddhhmmyyyy*, where *MM* is the two-digit month (01-12), *dd* is the two-digit day (01-31), *hh* is the two-digit hour (00-23), *mm* is the two-digit minute (00-59), and *yyyy* is the four-digit year. The time can be truncated from right to left, except for *MM*. If not all digits are specified, the year defaults to the current year, minutes to 0, hour to 0, and day to 01. At a minimum, the month must be specified. The command lists or removes only those records that were created at or before this time.

-L a | c | e | m | n | t | u | x

By default, only the command string (path name plus arguments) from each audit log record is listed. If this flag is specified, the argument is one or more of the following letters; the fields are displayed in the same order as the letters in the flag argument.

a

Displays all fields from the audit log in the following order: **t, u, m, n, x, c** (specifying **-L a** is the same as specifying **-L tumnxc**)

c

Displays the command string (the default)

e

Displays the standard error output

m

Displays the mapped identity

n

Displays the name of the node where the command ran

t

Displays the time field

u

Displays the authenticated user identity

x

Displays the LP command exit status

You cannot specify this flag with the **-c** flag.

-m mapped_ID

Specifies a mapped identity. **lphistory -m** lists or removes only those records that contain *mapped_ID*. You can use wild card characters in *mapped_ID*.

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

Specifies one or more nodes in the cluster on which the LP command history list is to be retrieved or cleared. (By default, the history list for the local node is retrieved or cleared.)

This flag is valid only in a management domain or a peer domain. If the **CT_MANAGEMENT_SCOPE** environment variable is not set, management domain scope is chosen first (if a management domain exists) and then peer domain scope is chosen, until the scope is valid for the command. The command runs once for the first valid scope it finds.

You cannot specify this flag with the **-a** flag.

-S *command_path*

Specifies a command path name. **lphistory -S** lists or removes only those records that contain *command_path*, which is identical to the value of the **CommandPath** in the LPCommands class (**/opt/rsct/bin/mkrsrc**, for example). You cannot use wild card characters in *command_path*.

-u *user_ID*

Specifies an authenticated user identity. **lphistory -u** lists or removes only those records that contain *user_ID*. You can use wild card characters in *user_ID*.

-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

num_records

Specifies the number of commands to be displayed from the history list. You can list a minimum of one command and a maximum of 100 commands. The default value is 10. You cannot specify this parameter with the **-c** flag.

Security

To run the **lphistory** command, you need 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 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.

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 list 20 LP commands that were previously issued on the local node, enter:

```
lphistory 20
```

2. Suppose nodeA is in a management domain and CT_MANAGEMENT_SCOPE is set to 3. To list the LP command history on nodeA, enter:

```
lphistory -c -n nodeA
```

3. To display the last 15 LP commands invoked with time, user ID, mapped ID, mechanism, return code, standard error, command name, and command string, enter:

```
lphistory -L a 15
```

4. To display the LP command names that end with `rsrc`, enter:

```
lphistory -C %rsrc
```

5. To display the LP commands that were invoked after 11:30 PM on April 18, 2006, enter:

```
lphistory -B 041823302006
```

Location

/opt/rsct/bin/lphistory

Contains the `lphistory` command.

lpmove Command

Note: This is a System V Print Subsystem command.

Purpose

Moves print requests.

Syntax

lpmove *Requests Destination*

lpmove *Destination1 Destination2*

Description

The **lpmove** command moves requests that were queued by **lp** between LP destinations. This command moves a specific *Request* to the specified *Destination*. *Requests* are request-IDs returned by **lp**. You can also attempt to move all requests for *Destination1* to *Destination2*. This form of the **lpmove** command causes **lp** to reject any new requests for *Destination1*.

Note: When moving requests, **lpmove** never checks the acceptance status of the new destination. Also, the request-IDs of the moved requests are not changed, so you can still find their requests. The **lpmove** command does not move requests that have options (such as content type and form required) that cannot be handled by the new destination.

If a request was originally queued for a class or the special destination **any** and the first form of **lpmove** was used, the destination of the request is changed to *New-Destination*. A request thus affected is printable only on *New-Destination* and not on other members of the class or other acceptable printers if the original destination was **any**.

If you enter `lpmove -?`, the system displays the command usage message and returns 0.

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>/var/spool/lp/*</code>	

lppchk Command

Purpose

Verifies files of an installable software product.

Syntax

lppchk [**-R** { **Path** | **ALL** }] { **-c** [**u**] | **-f** | **-l** [**u**] | **-v** } [**-m** [**1** | **2** | **3**]] [**-O** { [**r**] [**s**] [**u**] }] [*ProductName* [*FileList ...*]]

Description

The **lppchk** command verifies that files for an installable software product (fileset) match the Software Vital Product Data (SWVPD) database information for file sizes, checksum values, or symbolic links. A fileset is a separately installable option of a software package.

Flags

Item	Description
-c	Performs a checksum operation on the <i>FileList</i> items and verifies that the checksum and the file size are consistent with the SWVPD database.
-f	Checks that the <i>FileList</i> items are present and the file size matches the SWVPD database.
-l	Verifies symbolic links for files as specified in the SWVPD database.
-m [1 2 3]	Displays three levels of information. The levels are as follows: 1 Error messages only (default). 2 Error messages and warnings. 3 Error messages, warnings and informational messages.
-O {[r][s][u]}	Verifies the specified parts of the program. This flag is not needed with standalone systems because without this option all parts are verified by default. The flags specify the following parts: r Indicates the / (root) part is to be verified. s Indicates the /usr/share part is to be verified. u Indicates the /usr part is to be verified.
-R { Path ALL }	Indicates a user-specified installation location.
-u	Updates the SWVPD with new checksum or size information from the system when the system information does not match the SWVPD database. This flag sets symbolic links that are found to be missing. This flag is valid with only the -c or -l flag.
-v	Verifies that the / (root), /usr and /usr/share parts of the system are valid with each other. In other words, this flag verifies that all software products installed on the / (root) file system are also installed on the /usr file system and, conversely, all the software products installed in the /usr file system are also installed on the / (root) file system. You cannot specify <i>FileList</i> items with this flag. This flag also verifies requisites. Note: Only one of the -c , -f , -l , and -v flags can be specified with each use of the lppchk command.

Parameters

Item	Description
<i>FileList</i>	Specifies the file or files to check. This parameter is a list of file names separated by spaces. The file names can be a single name or a pair of names separated by a colon. The first form specifies a simple file and the second form specifies a member of an archive file, where the first name specifies the member and the second name specifies the archive file that contains the member. The full path name of the file or files must be specified. To specify multiple files you can use the pattern-matching characters * (asterisk) and ? (question mark), but they should be enclosed in a pair of 's (single quotes). Single quotes are recommended to prevent the korn shell wildcard expansion. If this parameter is omitted, all files of a software product are checked. If this parameter is specified, it must be preceded by a software product name.
<i>ProductName</i>	Specifies the name of the software product whose files are to be checked. If this parameter is omitted, all software products in the SWVPD are checked. To specify multiple software products you can use the pattern-matching characters * (asterisk) and ? (question mark), but they must be enclosed in a pair of 's (single quotes) to prevent the shell from expanding them.

Exit Status

Item	Description
0 (zero)	The command completed successfully.
nonzero	An error was found.

The **lppchk** command returns zero if no errors were found. Any other return value indicates an error was found.

Note: If **lppchk -f** (size) or **lppchk -c** (checksum) detects a mismatch in the respective size or checksum for a file, it does not report an error for the file if the file has been changed by an interim fix within **/usr/emgrdata/DBS/files.db**.

Examples

1. To verify all files that comprise the **X11.fnt** package, type:

```
lppchk -c X11.fnt
```

2. To verify the symbolic links of all software products whose names begin with **X11**, type:

```
lppchk -l 'X11*'
```

3. To verify that all filesets have all required requisites and are completely installed, type:

```
lppchk -v
```

Files

Item	Description
/etc/objrepos/lpp	Specifies installation information of all software products on the root.
/usr/lib/objrepos/lpp	Specifies installation information of all software products on the /usr file system.

Item	Description
/usr/share/lib/objrepos/lpp	Specifies installation information of all software products on the /usr/share file system.
/etc/objrepos/product	Specifies installation and update information of all software products on the root.
/usr/lib/objrepos/product	Specifies installation and update information of all software products on the /usr file system.
/usr/share/lib/objrepos/product	Specifies installation and update information of all the software products on the /usr/share file system.
/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.

lppmgr Command

Purpose

Manages an existing installp image source.

Syntax

```
lppmgr -d DirectoryOrDevice [ -r | -m MoveDirectory | -a APAR number ] { [ -x ] [ -X ] [ -l ] [ -u ] [ -b ] [ -k ] [ LANG ] } [ -p ] [ -t ] [ -s ] [ -v ] [ -D ]
```

Description

lppmgr is designed to perform the following functions on an existing installp image source (also known as an **lpp_source** in the NIM environment):

1. Remove duplicate updates (**-u** Flag).
2. Remove duplicate base levels (**-b** Flag).
3. Eliminating updates that are the same level as bases of the same file set. Such updates can create conflicts that lead to installation failure (**-u** Flag).
4. Remove message and locale file sets other than the language you specify (**-k** Flag).
5. Remove superseded file sets (**-x** Flag).
6. Remove non-system images from a NIM **lpp_source** resource (**-X** Flag).

By default, **lppmgr** lists all images that are filtered by the preceding routines. The **-r** flag can be used to remove the filtered images and the **-m** flag can be used to move the images to another location.

Note: **lppmgr** is not intended to replace **bffcreate**, install anything, or work with installed file sets. It is also not intended to address any issues other than those mentioned earlier. Before you use the **-X** flag, you must have a good understanding of NIM, system images (known as SIMAGES in NIM), and the workings of a NIM **lpp_source** resource.

Flags

Item	Description
-a <i>APAR number</i>	Displays file sets associated with an APAR number or keyword. If there are more than one APAR numbers, they must be contained in quotation marks and separated by spaces.
-b	Causes lppmgr to filter for base level duplicates.
-D	Specifies debug mode. This flag is for debugging the lppmgr script. Note: Debug. This produces a large quantity of output and greatly reduces lppmgr performance. It is not useful for normal operations.
-d <i>DeviceOrDirectory</i>	Specifies the device or directory where the installp images reside. Currently it can be any directory, NFS mount point, or cdrom device. If the directory is not writable, you must use the -t flag. If the target of your operation is a NIM lpp_source resource, you must specify the lpp_source location (see the lsnim command). This flag is required for all operations.
-k <i>LANG</i>	Keeps only the message and locale images for the language specified by <i>LANG</i> . All other languages are filtered.
-l	Lists filtered images only. By default, lppmgr will only list all filtered image files unless the "-r" or "-m" flag is specified. The "-l" flag will override the "-r" or "-m" flag.
-m <i>Directory</i>	Moves filtered files to <i>Directory</i> . The location that is specified by <i>Directory</i> can be any writable directory path. This flag cannot be used with the "-r" flag.
-p	Specifies prompt mode. Prompt when moving or removing files.
-r	Removes files that have been filtered by lppmgr . Note: If the prompt flag is not specified (-p), lppmgr removes all filtered files without further user interaction. This flag cannot be used with the "-m" flag.
-s	Prints space usage information. This flag prints the amount of space a particular file set is using and the total amount of space in question. Some buffer space is added for file metadata.
-t	Specifies that lppmgr does <i>not</i> rebuild the .toc file. This flag can be useful for having a quick look without having to rebuild the entire .toc file, which can take some time. Also, this flag is required for read-only devices.
-u	Causes lppmgr to filter for duplicate updates and conflicting updates that are the same level as bases of the same file set.
-V	Specifies verbose mode. lppmgr gives more output in certain situations.
-x	Causes lppmgr to filter for superseded updates.
-X	Filters non-system images from a NIM lpp_source resource.

Exit Status

- 0**
All **lppmgr** related operations that are completed successfully.
- >0**
An error occurred.

Security

Only the root user can execute **lppmgr**.

Examples

1. To list all duplicate and conflicting updates in image source directory **/myimages**, enter the following command:

```
lppmgr -d /myimages -u
```

2. To remove all duplicate and conflicting updates in image source directory **/myimages**, enter the following command:

```
lppmgr -d /myimages -u -r
```

3. To remove all duplicate and conflicting updates, duplicate base levels, and all message/locale file sets other than "en_US" in prompted mode, enter the following command:

```
lppmgr -d /myimages -purb -k en_US
```

4. To move all superseded update images and non SIMAGES from NIM **lpp_source** location **/lpps/433** to directory **/backups**, enter the following command:

```
lppmgr -d /lpps/433 -x -X -m /backups
```

5. To list all the file sets associated with APAR numbers IX38794 and IX48523 in image source directory **/myimages**, enter the following command:

```
lppmgr -d /myimages -a "IX38794 IX48523"
```

Files

Item	Description
/usr/lib/instl/lppmgr	Contains the lppmgr command.

lpq Command

The **lpq** command includes information for the AIX Print Subsystem **lpq** and the System V Print Subsystem **lpq**.

AIX Print Subsystem lpq Command

Purpose

Examines the spool queue.

Syntax

```
lpq [ + [ Number ] ] [ -l | -W ] [ -P Printer ] [ JobNumber ] [ UserName ]
```

Description

The **lpq** command reports the status of the specified job or all jobs associated with the specified *UserName* and *JobNumber* variables. *JobNumber* variable specifies the number of the job in the spool queue that you want to view. A *UserName* variable specifies viewing the jobs for the name of the person who submitted the job to that queue.

The **lpq** command reports on any jobs currently in the default queue when invoked without any options. Parameters supplied that are not recognized as parameters are interpreted as user names or job numbers to filter out only those jobs of interest.

For each job submitted (each job called by the **lpr** command), the **lpq** command reports the user's name, current rank in the queue, the name of the job, the job identifier (a number that can be supplied to the **lprm** command for removing a specific job), and the total size in blocks. Normally, only as much information as will fit on one line is displayed. Job ordering depends on the algorithm used to scan the spooling directory and is supposed to be FIFO (first-in-first-out). File names making up a job may be unavailable (when the **lpr** command is used as a sink in a pipeline). In this case, the file is indicated as - (standard input).

The display generated by the **lpq** command contains two entries for remote queues. The first entry contains the client's local queue and local device name and its status information. The second entry follows immediately; it contains the client's local queue name (again), followed by the remote queue name. Any jobs submitted to a remote queue are displayed first on the local side and are moved to the remote device as the job is processed on the remote machine.

Since the status commands communicate with remote machines, the status display may occasionally appear to hang while waiting for a response from the remote machine. The command will eventually time out if a connection cannot be established between the two machines.

Flags

Item	Description
-l	Generates the long output format.
+ [Number]	Displays the spool queue until it empties. A <i>Number</i> variable is the time in seconds before the display regenerates.
-P Printer	Displays the spool queue for the printer specified by the <i>Printer</i> variable. Note: Any destination command line options override both the LPDEST and the PRINTER environment variables.
-W	Displays a wide version of status information with longer queue names, device names, and job numbers. This flag cannot be used with the -l flag. If the -l flag and the -W flag are used simultaneously, the first one specified takes precedence.

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 a job number in the print queue lp0, enter:

```
lpq -P lp0
```

This command displays a list similar to the following:

Queue	Dev	Status	Job	Files	User	PP	%	Blks	CP	Rnk
lp0	d1p0	running	39	motd	guest	10	83	12	1	1

2. To display the status of the default queue in wide format, enter:

lpq -W

Files

Item	Description
<u>/usr/bin/lpq</u>	Contains the lpq command.
<u>/usr/sbin/qdaemon</u>	Contains the queuing daemon.
<u>/etc/qconfig</u>	Contains the queue configuration file.
<u>/etc/qconfig.bin</u>	Contains the digested, binary version of the <u>/etc/qconfig</u> file.
<u>/var/spool/lpd/qdir/*</u>	Contains queue requests.
<u>/var/spool/lpd/stat/*</u>	Contains information on the status of the devices.
<u>/var/spool/qdaemon/*</u>	Contains temporary copies of enqueued files.

System V Print Subsystem lpq Command

Purpose (System V)

(BSD) Displays the queue of printer jobs

Syntax (System V)

```
/usr/bin/lpq [-Pprinter] [-l] [+ [interval] ] [job# ...] [username ... ]
```

Description (System V)

The **lpq** command displays the contents of a printer queue. It reports the status of jobs specified by *job#*, or all jobs owned by the user specified by *username*. **lpq** reports on all jobs in the default printer queue when invoked with no arguments.

For each print job in the queue, **lpq** reports the user's name, current position, the names of input files comprising the job, the job number (by which it is referred to when using **lprm**) and the total size in bytes. Normally, only as much information as will fit on one line is displayed. Jobs are normally queued on a first-in-first-out basis. Filenames comprising a job may be unavailable, such as when **lpr** is used at the end of a pipeline; in such cases the filename field indicates the standard input.

If **lpq** warns that there is no daemon present (that is, due to some malfunction), the **lpc** command can be used to restart a printer daemon.

Output formatting is sensitive to the line length of the terminal; this can result in widely-spaced columns.

Flags (System V)

-P printer

Display information about the queue for the specified *printer*. In the absence of the **-P** flag, the queue to the printer specified by the **PRINTER** variable in the environment is used. If the **PRINTER** variable is not set, the queue for the default printer is used.

-l

Display queue information in long format; includes the name of the host from which the job originated.

+*[interval]*

Display the spool queue periodically until it empties. This option clears the terminal screen before reporting on the queue. If an *interval* is supplied, **lpq** sleeps that number of seconds in between reports.

Security (System V)

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 (System V)

/var/spool/lp

spooling directory.

/var/spool/lp/tmp/system_name/*-0

request files specifying jobs

Error Codes (System V)

lpq: printer is printing

The **lpq** program queries the spooler **LPSCHED** about the status of the printer. If the printer is disabled, the system administrator can restart the spooler using **lpc**.

lpq: printer waiting for auto-retry (offline ?)

The daemon could not open the printer device. The printer may be turned off-line. This message can also occur if a printer is out of paper, the paper is jammed, and so on. Another possible cause is that a process, such as an output filter, has exclusive use of the device. The only recourse in this case is to kill the offending process and restart the printer with **lpc**.

lpq: waiting for host to come up

A daemon is trying to connect to the remote machine named *host*, in order to send the files in the local queue. If the remote machine is up, **lpd** on the remote machine is probably dead or hung and should be restarted using **lpc**.

lpq: sending to host

The files are being transferred to the remote *host*, or else the local daemon has hung while trying to transfer the files.

lpq: printer disabled reason:

The printer has been marked as being unavailable with **lpc**.

lpq: The LP print service isn't running or can't be reached.

The **lpsched** process overseeing the spooling queue does not exist. You can restart the printer daemon with **lpc**.

lpq: printer: unknown printer

The *printer* was not found in the System V LP database. Usually this is a typing mistake; however, it may indicate that the printer does not exist on the system. Use **lpstat -p** to find the reason.

lpq: error on opening queue to spooler

The connection to **lpsched** on the local machine failed. This usually means the printer server started at boot time has died or is hung. Check if the printer spooler daemon */usr/lib/lp/lpsched* is running.

lpq: Can't send message to LP print service

lpq: Can't establish contact with LP print service

These indicate that the LP print service has been stopped. Get help from the system administrator.

lpq: Received unexpected message from LP print service

It is likely there is an error in this software. Get help from system administrator.

lpr Command

The **lpr** command includes information for the AIX Print Subsystem **lpr** and the System V Print Subsystem **lpr**.

AIX Print Subsystem lpr Command

Purpose

Enqueues print jobs.

Syntax

```
lpr [ -f ] [ -g ] [ -h ] [ -j ] [ -l ] [ -m ] [ -n ] [ -p ] [ -r ] [ -s ] [ -P Printer ] [ -# NumberCopies ] [ -C Class ] [ -J Job ] [ -T Title ] [ -i [ NumberColumns ] ] [ -w Width ] [ File ... ]
```

Description

The **lpr** command uses a spooling daemon to print the named *File* parameter when facilities become available. If no files are specified, the **lpr** command reads from standard input.

Flags

Item	Description
-# Number	Produces multiple copies of output, using the <i>Number</i> variable as the number of copies for each file named.
-C Class	Specifies the print <i>Class</i> as the job classification on the burst page.
-f	Uses a filter that interprets the first character of each line as a standard FORTRAN carriage control character.
-g	The files are assumed to contain standard plot data.
-h	Suppresses printing of the burst page. Note: The default is to print a header page and not a trailer page.
-i [Number]	Indents output <i>Number</i> spaces. If the <i>Number</i> variable is not given, eight spaces are used as the default.
-j	Specifies that the message <code>Job number is: nnn</code> , where <i>nnn</i> is the assigned job number, be displayed to standard output. This occurs only if the job is submitted to a local print queue.
-J Job	Prints the <i>Job</i> variable as the job name on the burst page. Usually, the lpr command uses the name of the first file.
-l	(Lowercase L) Uses a filter that allows control characters to be printed.
-m	Sends mail upon completion of spooling.
-n	Uses a filter that formats files containing <i>ditroff</i> (device-independent <i>troff</i>) data.
-P Printer	Forces output to the <i>Printer</i> variable. If this flag is not specified, the following conditions occur: <ul style="list-style-type: none">• If a default exists, the lpr command uses the default printer.• If the LPDEST environment variable is set, then lpr uses the value specified by the LPDEST variable. If set, this value is always used, even if the PRINTER variable is also set.• If the PRINTER variable is set and no LPDEST variable is set, then lpr uses the value specified by the PRINTER environment variable. Note: Any destination command line options override both the LPDEST and the PRINTER environment variables.
-p	Uses the pr command to format the file (<code>lpr -p</code> is very much like <code>pr lpr</code>).
-r	Removes the file upon completion of spooling.

Item	Description
-s	Prints from the files specified on the command line rather than trying to copy them (so large files can be printed). This means the data files should not be modified or removed until they have been printed. Note that this flag only works on the local host (files sent to remote printer hosts are copied anyway), and only with named data files. It does not work if the lpr command is at the end of a pipeline.
-T Title	Uses the <i>Title</i> variable instead of the file name for the title used by the pr command.
-w Number	Uses the <i>Number</i> variable as the page width for the pr 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

1. To print three copies of the files `new.index.c`, `print.index.c`, and `more.c`, enter:

```
lpr -#3 new.index.c print.index.c more.c
```

Prints three copies of the `new.index.c` file, three copies of the `print.index.c` file, and three copies of the `more.c` file.

2. To print three copies of the concatenation of three files `new.index.c`, `print.index.c`, and `more.c`, enter:

```
cat new.index.c print.index.c more.c | lpr -#3
```

3. To print Operations on the burst page, followed by file `new.index.c`, enter:

```
lpr -C Operations new.index.c
```

This replaces the system name (the name returned by host name) with Operations on the burst page.

4. To queue the MyFile file and return the job number, enter:

```
lpr -j MyFile
```

Files

Item	Description
<u>/usr/sbin/qdaemon</u>	Queuing daemon.
<u>/etc/qconfig</u>	Queue configuration file.
<u>/etc/qconfig.bin</u>	Digested, binary version of the <u>/etc/qconfig</u> file.
<u>/var/spool/lpd/qdir/*</u>	Queue requests.
<u>/var/spool/lpd/stat/*</u>	Information on the status of the queues.
<u>/var/spool/qdaemon</u>	Temporary copies of enqueued files.

System V Print Subsystem lpr Command

Purpose (System V)

(BSD) Sends a job to the printer.

Syntax (System V)

```
/usr/bin/lpr [ -P printer ] [ -# copies ] [ -C class ] [ -J job ] [ -T title ] [ -i [indent] ] [ -w cols ] [ -r ] [ -m ] [ -h ] [ -s ]  
[ -filter_option ] [file ... ]
```

Description (System V)

The **lpr** command forwards printer jobs to a spooling area for subsequent printing as facilities become available. Each printer job consists of copies of each *file* you specify. The spool area is managed by the line printer spooler, **lp sched**. **lpr** reads from the standard input if no files are specified.

lp is the preferred interface.

Command-line options cannot be combined into a single argument as with some other commands. The command:

```
lpr -fs
```

is not equivalent to

```
lpr -f -s
```

Placing the **-s** flag first, or writing each option as a separate argument, makes a link as expected.

lpr -p is not precisely equivalent to **pr | lpr**. **lpr -p** puts the current date at the top of each page, rather than the date last modified.

Fonts for **troff** and T[E]X reside on the printer host. It is not possible to use local font libraries.

lpr objects to printing binary files.

If userA uses **su** to become userB and uses */usr/bin/lpr*, then the printer request will be entered as userB, not userA

Flags (System V)

-P printer

Send output to the named *printer*. Otherwise send output to the printer named in the **PRINTER** environment variable, or to the default printer, **lp**.

-# copies

Produce the number of *copies* indicated for each named file. For example:

```
lpr -#3 index.c lookup.c
```

produces three copies of *index.c*, followed by three copies of *lookup.c*. On the other hand,

```
cat index.c lookup.c | lpr -#3
```

generates three copies of the concatenation of the files.

-C class

Print *class* as the job classification on the burst page. For example,

```
lpr -C Operations new.index.c
```

replaces the system name (the name returned by ``hostname'') with **Operations** on the burst page, and prints the file *new.index.c*.

-J *job*

Print *job* as the job name on the burst page. Usually, **lpr** uses the first file's name.

-T *title*

Use *title* instead of the file name for the title used by **pr**.

-i[*indent*]

Indent output *indent* <Space> characters. Eight <Space> characters is the default.

-w *cols*

Use *cols* as the page width for **pr**.

-r

Remove the file upon completion of spooling, or upon completion of printing with the **-s** flag.

-m

Send mail upon completion.

-h

Suppress printing the burst page.

-s

Use the full pathnames (not symbolic links) of the files to be printed rather than trying to copy them. This means the data files should not be modified or removed until they have been printed. This flag only prevents copies of local files from being made. Jobs from remote hosts are copied anyway. The **-s** flag only works with named data files; if the **lpr** command is at the end of a pipeline, the data is copied to the spool.

filter_option

The following single letter options notify the line printer spooler that the files are not standard text files. The spooling daemon will use the appropriate filters to print the data accordingly.

-p

Use **pr** to format the files (**lpr -p** is very much like **pr | lpr**).

-l

Print control characters and suppress page breaks.

-t

The files contain **troff** (cat phototypesetter) binary data.

-n

The files contain data from *ditroff* (device independent **troff**).

-d

The files contain data from *tex* (DVI format from Stanford).

-g

The files contain standard plot data as produced by the routine **plot** for the filters used by the printer spooler.

-v

The files contain a raster image. The printer must support an appropriate imaging model such as PostScript in order to print the image.

-c

The files contain data produced by *cifplot*.

-f

Interpret the first character of each line as a standard FORTRAN carriage control character.

If no *filter_option* is given (and the printer can interpret PostScript), the string ``%!'` as the first two characters of a file indicates that it contains PostScript commands.

These filter options offer a standard user interface, and all options may not be available for, nor applicable to, all printers.

Security (System V)

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 (System V)

/usr/lib/lp/lpsched

System V line printer spooler

/var/spool/lp/tmp/*

directories used for spooling

/var/spool/lp/tmp/system/*-0

spooler control files

/var/spool/lp/tmp/system/*-N

(*N* is an integer and > 0) data files specified in `*-0' files

Error Codes (System V)

lpr: printer: unknown printer

The *printer* was not found in the LP database. Usually this is a typing mistake; however, it may indicate that the printer does not exist on the system. Use **lpstat -p** to find the reason.

lpr: error on opening queue to spooler

The connection to **lpsched** on the local machine failed. This usually means the printer server started at boot time has died or is hung. Check if the printer spooler daemon **/usr/lib/lpsched** is running.

lpr: printer: printer queue is disabled

This means the queue was turned off with

```
/usr/etc/lpc disable printer
```

to prevent **lpr** from putting files in the queue. This is usually done when a printer is going to be down for a long time. The printer can be turned back on by a privileged user with **lpc**.

lpr: Can't send message to the LP print service

lpr: Can't establish contact with the LP print service

These indicate that the LP print service has been stopped. Get help from the system administrator.

lpr: Received unexpected message from LP print service

It is likely there is an error in this software. Get help from system administrator.

lpr: There is no filter to convert the file content

Use the **lpstat -p -l** command to find a printer that can handle the file type directly, or consult with your system administrator.

lpr: cannot access the file

Make sure file names are valid.

lprm Command

The **lprm** command includes information for the AIX Print Subsystem **lprm** and the System V Print Subsystem **lprm**.

AIX Print Subsystem lprm Command

Purpose

Removes jobs from the line printer spooling queue.

Syntax

```
lprm [ -P Printer ] [ JobNumber ] [ UserName ... ] [ - ]
```

Description

The **lprm** command removes one or more jobs from the spool queue of a printer.

You cannot run the **lprm** command without specifying a job number, the - (minus sign) flag, or at least one user name.

Specifying a *UserName* parameter, or list of names, causes the **lprm** command to attempt to remove any jobs queued belonging to that user (or users).

You can remove an individual job from a queue by specifying its *JobNumber*. This job number is obtained by using the **lpq** command.

Flags

Item	Description
-	Removes all jobs a user owns. Someone with root user authority can use this flag to remove all jobs from a queue. This flag is not valid for remote print.
-P Printer	Specifies the queue associated with a specific <i>Printer</i> variable. If this flag is not specified, the following conditions occur: <ul style="list-style-type: none">• If the LPDEST environment variable is set, then lprm uses the value specified by the LPDEST variable. If set, this value is always used, even if the PRINTER variable is also set.• If the PRINTER variable is set and no LPDEST variable is set, then lprm uses the value specified by the PRINTER environment variable. If neither the LPDEST nor the PRINTER variable is set, the lprm command removes jobs from the default queue. <p>Note: Any destination command line options override both the LPDEST and the PRINTER environment variables.</p>

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 job number 13 from the default printer queue, enter:

```
lprm 13
```

2. To remove job number 13 from printer queue lp0, enter:

```
lprm -P lp0 13
```

3. To remove a job from the printer queue for a certain user, enter:

```
lprm guest
```

Files

Item	Description
<code>/usr/bin/lprm</code>	Contains the lprm command.
<code>/etc/qconfig</code>	Contains the configuration file.

System V Print Subsystem lprm Command

Purpose (System V)

(BSD) Remove jobs from the printer queue

Syntax (System V)

```
/usr/bin/lprm [-Pprinter] [-] [job # ...] [username...]
```

Description (System V)

The **lprm** command removes a job or jobs from a printer's spooling queue. Since the spool directory is protected from users, using **lprm** is normally the only method by which a user can remove a job.

Without any arguments, **lprm** deletes the job that is currently active, provided that the user who invoked **lprm** owns that job.

When the privileged user specifies a *username*, **lprm** removes all jobs belonging to that user.

You can remove a specific job by supplying its job number as an argument, which you can obtain using **lpq**. For example:

```
lpq -Phost
host is ready and printing
Rank      Owner   Job    Files      Total Size
active    wendy   385    standard input  35501 bytes
lprm -Phost 385
```

lprm reports the names of any files it removes, and is silent if there are no applicable jobs to remove.

lprm Sends the request to cancel a job to the print spooler, **LPSCHED**.

An active job may be incorrectly identified for removal by an **lprm** command issued with no arguments. During the interval between an **lpq** command and the execution of **lprm**, the next job in queue may have become active; that job may be removed unintentionally if it is owned by you. To avoid this, supply **lprm** with the job number to remove when a critical job that you own is next in line.

Only the privileged user can remove print jobs submitted from another host.

Flags (System V)

-Pprinter

Specify the queue associated with a specific printer. Otherwise the value of the **PRINTER** variable in the environment is used. If this variable is unset, the queue for the default printer is used.

-

Remove all jobs owned by you. If invoked by the privileged user, all jobs in the spool are removed. Job ownership is determined by the user's login name and host name on the machine where the **lprm** command was executed.

Security (System V)

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 (System V)

/var/spool/lp/*
spooling directories

Error Codes (System V)

lprm: printer: unknown printer

The *printer* was not found in the System V LP database. Usually this is a typing mistake; however, it may indicate that the printer does not exist on the system. Use **lpstat -p** to get the status of printers.

lprm: error on opening queue to spooler

The connection to **lp sched** on the local machine failed. This usually means the printer server started at boot time has died or is hung. Check if the printer spooler daemon **/usr/lib/lp/lpsched** is running.

lprm: Can't send message to the LP print service

lprm: Can't receive message from the LP print service

These indicate that the LP print service has been stopped. Get help from the system administrator.

lprm: Received unexpected message from the LP print service

It is likely there is an error in this software. Get help from system administrator.

lprm: Can't cancel request

You are not allowed to remove another user's print request.

lpsched Command

Note: This is a System V Print Subsystem command.

Purpose

Starts/stops the print service.

Syntax

/usr/lib/lp/lpsched

lpshut

Description

The **lpsched** command starts the LP print service.

The **lpshut** command shuts down the print service. All printers that are printing at the time the **lpshut** command is invoked stop printing. When **lpsched** is started again, requests that were printing at the time a printer was shut down are reprinted from the beginning.

You must have the appropriate privilege to run these commands.

If the scheduler fails to run, check the **lpsched** log file, which contains all failed attempts to load print requests, printer descriptions, forms, filters, classes, alerts, and systems. The log files are located in **/var/lp/logs**. Useful information on the networked print service can also be found in the **/var/lp/logs/lpNet** log file.

If you enter **lpsched -?**, the system displays the command usage message and returns 0.

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>/var/spool/lp/*</code>	

lpstat Command

The **lpstat** command includes information for the AIX Print Subsystem **lpstat** and the System V Print Subsystem **lpstat**.

AIX Print Subsystem lpstat Command

Purpose

Displays line printer status information.

Syntax

```
lpstat [ -aList ] [ -cList ] [ -d ] [ -oList ] [ -pList ] [ -r ] [ -s ] [ -t ] [ -uList ] [ -vList ] [ -W ]
```

Description

The **lpstat** command displays information about the current status of the line printer.

If no flags are given, **lpstat** prints the status of all requests made by the **lp** command.

Flags can appear in any order and can be repeated. Some flags take an optional list as a parameter. Enter the list as either a list of items separated by commas, as in `lpstat -aQueue1, Queue2`, or as a list of items enclosed in single or double quotes and separated either by commas or one or more spaces, as in, for example, `lpstat -a"Queue1 Queue2"` or `lpstat -a'Queue1, Queue2'` or `lpstat -a 'Queue1 Queue2'` or `lpstat -a'Queue1, Queue2'`.

If you specify a flag with no parameters, all information pertaining to that flag is printed.

The display generated by the **lpstat** command contains two entries for remote queues. The first entry contains the client's local queue and local device name and its status information. The second entry contains the client's local queue name followed by the remote queue name. The spooling subsystem first displays remote print requests on the local queue. When the remote machine begins to process the remote print job, the status display for the print job moves to the remote queue.

When a status command communicates with a remote host, the display occasionally appears to hang while the command waits for a response from the remote machine. The command eventually times out if no connection is established between the two machines.

Flags

Item	Description
-aList	Provides status and job information on queues. Specifying the lpstat command with this flag is the same as specifying the enq -q -PQueue1 -PQueue2 ... command (where <i>Queue1</i> , <i>Queue2</i> , etc., are items in <i>List</i>).
-cList	Provides status and job information on queues. Specifying the lpstat command with this flag is the same as specifying the enq -q -PQueue1 -PQueue2 ... command (where <i>Queue1</i> , <i>Queue2</i> , etc., are items in <i>List</i>).
-d	Prints the status information for the system default destination for the lp command. Specifying the lpstat command with this flag is the same as specifying the enq -q command.

Item	Description
-oList	Prints the status of print requests or print queues. <i>List</i> is a list of intermixed printer names and job numbers.
-pList	Prints the status of printers. Note: You cannot use both the -p flag and the -t flag at the same time.
-r	Provides status and job information on queues. Specifying the lpstat command with this flag is the same as specifying the enq -A command.
-s	Displays a status summary, including a list of printers and their associated devices. Specifying the lpstat command with this flag is the same as specifying the enq -A command.
-t	Displays all status information, including a list of printers and their associated devices. Specifying the lpstat command with this flag is the same as specifying the enq -AL command.
-uList	Prints the status of all print requests for users specified in <i>List</i> . <i>List</i> is a list of login names. Specifying the lpstat command with this flag is the same as specifying the enq -u UserName command.
-vList	Prints the status of printers. The <i>List</i> variable is a list of printer names.
-W	Displays a wide version of the status information with longer queue names, device names, and job numbers. This flag cannot be used with the -t flag. If the -t flag and the -W flag are used simultaneously, the first specified flag takes precedence. If the -W flag and -l flag are used simultaneously, the result displays the long status of the print job in the semicolon-separated format.

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 display the status for all print queues, enter:

```
lpstat
```

2. To display the long status for all printers, enter:

```
lpstat -t
```

3. To display a job number in the print queue lp0, enter:

```
lpstat -plp0
```

This command displays a list similar to the following:

Queue	Dev	Status	Job	Files	User	PP	%	Blks	CP	Rnk
lp0	d1p0	running	39	motd	guest	10	83	12	1	1

4. To display the status for users `root`, `ghandi`, and `king`, enter:

```
lpstat -u"root,ghandi,king"
```

5. To display the status of all print queues in wide format, enter:

```
lpstat -W
```

Files

Item	Description
<code>/var/spool/lpd/*</code>	Contains temporary copies of remote enqueued files.

System V Print Subsystem `lpstat` Command

Purpose (System V)

Prints information about the status of the LP print service.

Syntax (System V)

```
lpstat [flags] [request-ID-list]
```

Description (System V)

The **lpstat** command displays information about the current status of the LP print service. If no *flags* are given, **lpstat** displays the status of all print requests made by you.

The command **lpstat -o *printername*** is used to list all the requests queued on the specified printer. If *printername* points to a remote printer, then **lpstat -o *printername*** lists all the requests on the remote printer, not just those submitted locally.

Any arguments that are not *flags* are assumed to be *request-IDs* as returned by **lp**. The **lpstat** command displays the status of such requests. The *flags* may appear in any order and may be repeated and intermixed with other arguments. Some of the keyletters below may be followed by an optional *list* that can be in one of two forms:

- a list of items separated by commas, for example, **-p *printer1,printer2***
- a list of items separated by spaces and enclosed in quotes, for example, **-u "user1 user2 user3"**

Specifying **all** after any keyletter that takes *list* as an argument causes all information relevant to the keyletter to be displayed. For example, the command **lpstat -a all** lists the accepting status of all print destinations.

The omission of a *list* following such keyletters causes all information relevant to the keyletter to be displayed. For example, the command **lpstat -a** is equivalent to **lpstat -a all**.

There are two exceptions to the behavior of the **all** keyword. The first is when it is used in conjunction with the **-o** flag: **lpstat -o all** only lists requests submitted locally to remote printers. The second is when it is used with directory-enabled print queues. Use of the **all** keyword will only return non-directory-enabled print queues. **lpstat -a *list*** will report whether the both directory-enabled and non-directory-enabled print queues in *list* are accepting requests. For the **-a** and **-b** flags, **lpsched** will remember the directory-enabled print queues specified until it is restarted. Subsequent calls to **lpstat -a** and **lpstat -p** will report the status of all non-directory-enabled print queues as well as the directory-enabled print queues previously specified. Once **lpsched** has been restarted, the use of the **all** keyword with the **lpstat** command will once again only display non-directory-enabled print queues. The **dslpsearch** command should be used to search for defined directory-enabled print queues.

If you enter `lpstat -?`, the system displays the command usage message and returns 0.

Flags (System V)

-a [*list*]

Report whether print destinations are accepting requests. *list* is a list of intermixed printer names and class names.

-c [*list*]

Report names of all classes and their members. *list* is a list of class names.

-d

Report what the system default destination is (if any).

-f [*list*] [-l]

Verify that the forms in *list* are recognized by the LP print service. *list* is a list of forms; the default is **all**. The **-l** option will list the form parameters.

-o [*list*] [-l]

Report the status of print requests. *list* is a list of intermixed printer names, class names, and *request-IDs*. The keyletter **-o** may be omitted. The **-l** option lists for each request whether it is queued for, assigned to, or being printed on a local printer, the form required (if any), and the character set or print wheel required (if any). Note that required forms (if any) are not listed for remote printers.

-p [*list*] [-D] [-l]

If the **-D** flag is given, a brief description is printed for each printer in *list*. If the **-l** flag is given, a full description of each printer's configuration is given, including the form mounted, the acceptable content and printer types, a printer description, the interface used, and so on.

In order to maintain system security access information, the information needed to produce the printer status given by **lpstat -p** is available only if the LP scheduler is running.

-r

Report the status of the LP request scheduler (whether it is running).

-R

Report a number showing the rank order of jobs in the print queue for each printer.

-s [-l]

Display a status summary, including the status of the LP scheduler, the system default destination, a list of class names and their members, a list of printers and their associated devices, a list of the systems sharing print services, a list of all forms and their availability, and a list of all recognized character sets and print wheels. The **-l** flag displays all parameters for each form and the printer name where each character set or print wheel is available.

-S [*list*] [-l]

Verify that the character sets or the print wheels specified in *list* are recognized by the LP print service. Items in *list* can be character sets or print wheels; the default for *list* is **all**. If the **-l** flag is given, each line is appended by a list of printers that can handle the print wheel or character set. The list also shows whether the print wheel or character set is mounted or specifies the built-in character set into which it maps.

-t [-l]

Display all status information: all the information obtained with the **-s** flag, plus the acceptance and idle/busy status of all printers and status of all requests. The **-l** flag displays more detail as described for the **-f**, **-o**, **-p**, and **-s** flag.

-u [*login-ID-list*]

Display the status of output requests for users. The *login-ID-list* argument may include any or all of the following constructs:

login-ID

a user on the local system

system-name!login-ID

a user on system *system-name*

system-name!all
all users on system *system-name*

all!login-ID
a user on all systems

all
all users on the local system

all!all
all users on all systems

The default value of *login-ID-list* is **all**.

Security (System V)

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.

lpssystem Command

Purpose

Registers remote systems with the print service.

Note: This is a System V Print Subsystem command.

Syntax

lpssystem [*-t Type*] [*-T Timeout*] [*-R Retry*] [*-y Comment*] *SystemName* [*SystemName . . .*]

lpssystem -l [*SystemName . . .*]

lpssystem -r *SystemName* [*SystemName . . .*]

lpssystem -A

Description

The **lpssystem** command defines parameters for the LP print service, with respect to communication (via a high-speed network such as TCP/IP) with remote systems.

Specifically, the **lpssystem** command defines remote systems with which the local LP print service can exchange print requests. These remote systems are described to the local LP print service in terms of several parameters that control communication: type, retry, and timeout. These parameters are defined in **/etc/lp/Systems**. You can edit this file with a text editor (such as **vi**), but editing is not recommended. By using **lpssystem**, you can ensure that **lpsched** is notified of any changes to the *Systems* file.

The *Timeout* parameter specifies the length of time (in minutes) that the print service should allow a network connection to be idle. If the connection to the remote system is idle (that is, there is no network traffic) for *N* minutes, then drop the connection. (When there is more work, the connection is re-established.) Legal values are **n**, **0**, and *N*, where *N* is an integer greater than 0. If a decimal number is used for *N*, it is truncated to the whole number. The value **n** means never time out; **0** means as soon as the connection is idle, drop it. The default is **n**.

The *Retry* parameter specifies the length of time (in minutes) to wait before trying to re-establish a connection to the remote system, when the connection was dropped abnormally (that is, a network error). Legal values are **n**, **0**, and *N*, where *N* is an integer greater than 0. It means wait *N* minutes before trying to reconnect. If a decimal number is used for *N*, it is truncated to the whole number. (The default is 10

minutes.) The value **n** means do not retry dropped connections until there is more work; **0** means try to reconnect immediately.

The *Comment* parameter allows you to associate a free form comment with the system entry. This is visible when **lpsystem -l** is used.

The *SystemName* is the name of the remote system from which you want to be able to receive jobs and to which you want to be able to send jobs. A special entry is provided with the **/etc/lp/Systems** file by default, which allows all connections to **bsd** systems. That entry uses the asterisk (*) as the *SystemName*.

The command **lpsystem -l [SystemName]** prints out a description of the parameters associated with *SystemName* (if a system has been specified) or with all the systems in its database (if *SystemName* has not been specified).

The command **lpsystem -r SystemName** removes the entry associated with *SystemName*. The print service no longer accepts jobs from that system or send jobs to it, even if the remote printer is still defined on the local system. The scheduler must be running when the removal of a systems file entry occurs, because the scheduler checks whether the system entry is currently used by a printer destination. If currently used, the system entry cannot be removed.

If you use **lpsystem -r SystemName** to remove a system and you have active printers for that system, you will not be allowed to remove the system from the system file. **lpsystem -r SystemName** only works if no printers for that system exist.

With respect to the semantics of the *Timeout* and *Retry* values, the print service uses one process for each remote system with which it communicates, and it communicates with a remote system only when there is work to be done on that system or work is being sent from that system.

The system initiating the connection is the master process, and the system accepting the connection is the secondary process. This designation serves only to determine which process dies (the secondary) when a connection is dropped. This helps prevent more than one process communicating with a remote system. All connections are bi-directional, regardless of the master/secondary designation. You cannot control a system's master/secondary designation. Typically, a client machine has the master child, and the server machine has the secondary child. If a master process times out, then both the secondary and master exit. If a secondary process times out, then it is possible that the master may still live and retry the connection after the retry interval. Therefore, one system's resource management strategy can affect another system's strategy.

All forms of the **lpsystem** command accept ***** (asterisk enclosed in double quotes) for *SystemName*.

Depending upon the configuration of the name server, you may need to change the entry in the *SystemName* field in **/etc/lp/Systems** to a full domain name.

If you enter `lpsystem -?`, the system displays the command usage message and returns 0.

Flags

Item	Description
-A	Prints out the TCP/IP address in a format.
-l [SystemName]	Prints out a description of the parameters associated with <i>SystemName</i> or with all the systems in its database.
-r SystemName	Removes the entry associated with <i>SystemName</i> .
-R Retry	Specifies time to wait before trying to reestablish a connection for a remote system.
-T Timeout	Specifies the time allowed for a network connection to be idle. <i>Timeout</i> is in minutes. Default is to never time out.

Item	Description
-y <i>Comment</i>	Allows you to associate a free-form comment with the system entry.
-t <i>Type</i>	Specifies the type of remote system. The only supported value for the -t flag is <code>bsd</code> .

Security

Only a user with appropriate privileges may execute the **lpsystem** 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.

Files

Table 3. Files	
Item	Description
<code>/etc/lp/*</code>	<code>/var/spool/lp/*</code>

lptest Command

Purpose

Generates the line printer ripple pattern.

Syntax

lptest [;*Length Count*]

Description

The **lptest** command writes the traditional "ripple" test pattern on a standard output device such as a terminal or a printer. In 96 lines, this pattern will print all 96 printable ASCII characters in each position. While originally created to test printers, the ripple pattern is quite useful for testing terminals, driving terminal ports for debug purposes, or any other task where a quick supply of random data is needed.

Using the **lptest** command, you can specify the output line length if the default length of 79 is not appropriate. You can also specify the number of output lines to be generated if the default *Count* parameter of 200 is not appropriate. Note that if *Count* parameter is specified, *Length* must also be specified.

Examples

To display or print 100 lines of 80-column test output to standard output, enter:

```
lptest 80 100
```

lpusers Command

Note: This is a System V Print Subsystem command.

Purpose

Set printing queue priorities.

Syntax

lpusers -d *PriorityLevel*

lpusers -q *PriorityLimit* **-u** *LoginIDList*

lpusers -u *LoginIDList*

lpusers -q *PriorityLimit*

lpusers -l

Description

The **lpusers** command sets limits to the queue priority level that can be assigned to jobs submitted by users of the LP print service.

The first form of the command (with **-d**) sets the system-wide priority default to *PriorityLevel*, where *PriorityLevel* is a value of 0 to 39, with 0 being the highest priority. If a user does not specify a priority level with a print request, the default priority is used. Initially, the default priority level is 20.

The second form of the command (with **-q** and **-u**) sets the default *PriorityLimit* (from 0 to 39) that the users in the *LoginIDList* can request when submitting a print request. The *LoginIDList* parameter may include any or all of the following constructs:

Users that have been given a limit cannot submit a print request with a higher priority level than the one assigned, nor can they change a request already submitted to have a higher priority. Any print requests submitted with priority levels higher than allowed will be given the highest priority allowed.

The third form of the command (with **-u**) removes any explicit priority limit for the specified users.

The fourth form of the command (with **-q**) sets the default priority limit for all users not explicitly covered by the use of the second form of this command.

The last form of the command (with **-l**) lists the default priority level and the priority limits assigned to users.

If you enter `lpusers -?`, the system displays the command usage message and returns 0.

Parameters

Item	Description
<i>LoginID</i>	Specifies a user on the local system.
<i>system_name!login-ID</i>	User on the system <i>system_name</i>
<i>system_name!all</i>	Users on system <i>system_name</i>
all! <i>login-ID</i>	User on all systems
all	Users on the local system

Flags

Item	Description
-d <i>PriorityLevel</i>	Sets the system-wide priority default to <i>PriorityLevel</i> .
-l	Lists the default priority level and the priority limits assigned to users.

Item	Description
<code>-q PriorityLimit</code>	Sets the default highest priority level for all users not explicitly covered.
<code>-q PriorityLimit -u LoginIDList</code>	Sets the default highest priority level users in <i>LoginIDList</i> can request when submitting a print request.
<code>-u LoginIDList</code>	Removes any explicit priority level for the specified users.

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.

ls Command

Purpose

Displays the contents of a directory.

Syntax

To Display Contents of Directory or Name of File

```
ls [-1] [-A] [-C] [-E] [-F] [-H] [-L] [-N] [-R] [-S] [-X] [-a] [-b] [-c] [-d] [-e] [-f] [-g] [-i] [-k]
[-l] [-m] [-n] [-o] [-p] [-q] [-r] [-s] [-t] [-u] [-U] [-x] [File ...]
```

To Display Contents of Directory

```
ls -f [-C] [-d] [-i] [-m] [-s] [-X] [-x] [-1] [-U] [Directory ...]
```

Description

The **ls** command writes to standard output the contents of each specified *Directory* parameter or the name of each specified *File* parameter, along with any other information you ask for with the flags. If you do not specify a *File* or *Directory* parameter, the **ls** command displays the contents of the current directory.

Specifying more than one of the options in the mutually exclusive pairs is not considered an error. The last option specified in each pair determines the output format.

By default, the **ls** command displays all information in alphabetic order by file name. The collating sequence is determined by the **LANG** or **LC_COLLATE** environment variable.

When the **ls** command displays the contents of a directory, it does not show entries for files whose names begin with a . (dot) unless you use the **-a** or **-A** flag. If the command is executed by root, it uses the **-A** flag by default.

There are three main ways to format the output:

- List one entry per line.
- List entries in multiple columns by specifying either the **-C** or **-x** flag. The **-C** flag is the default format when output is to a TTY. The **ls** command displays single column output if file or directory names are too long.
- List entries in a comma-separated series by specifying the **-m** flag.

To determine the number of character positions in the output line, the **ls** command uses the **COLUMNS** environment variable. If this variable is not set, the command gets the current column value of the display. If the **ls** command cannot determine the number of character positions by either of these methods, it uses a default value of 80.

The mode displayed with the **-U** flag is the same as with the **-l** flag, except for the addition of an 11th character interpreted as follows:

Item	Description
E	Indicates a file has extended attributes (EA) information. The EA of a file is displayed by using the getea command.
-	Indicates a file does not have extended attributes information.
e	Indicates a file is encrypted.

Encryption takes precedence over the presence of the Access Control Lists (ACLs) and other EAs.

The mode displayed with the **-e** and **-l** flags is interpreted as follows:

If the first character is:

Item	Description
d	The entry is a directory.
b	The entry is a block special file.
c	The entry is a character special file.
l	The entry is a symbolic link, and either the -N flag was specified or the symbolic link did not point to an existing file.
p	The entry is a first-in, first-out (FIFO) special file.
s	The entry is a local socket.
-	The entry is an ordinary file.

The next nine characters are divided into three sets of three characters each. The first set of three characters show the owner's permission. The next set of three characters show the permission of the other users in the group. The last set of three characters shows the permission of anyone else with access to the file. The three characters in each set indicate, respectively, read, write, and execute permission of the file. Execute permission of a directory lets you search a directory for a specified file.

Permissions are indicated as follows:

Item	Description
r	Read
w	Write (edit)
x	Execute (search)
-	Corresponding permission not granted

The group-execute permission character is **s** if the file has set-group-ID mode. The user-execute permission character is **S** if the file has set-user-ID mode. The last character of the mode (usually **x** or **-**) is **T** if the 01000 (octal) bit of the mode is set (see the **chmod** command for the meaning of this mode). The indications of set-ID and 01000 bit of the mode are capitalized (**S** and **T**, respectively) if the corresponding execute permission is not set. The mode **t** indicates that the sticky bit is on for the file or the directory.

The mode displayed with the **-e** flag is the same as with the **-l** flag, except for the addition of an 11th character interpreted as follows:

Item	Description
-------------	--------------------

- | | |
|----------|--|
| + | Indicates a file has extended security information. For example, the file may have extended ACL , TCB , or TP attributes in the mode.

The access control information (ACL) of a file is displayed by using the aclget command. The value of the TCB and TP attributes are displayed by using the chtcb command. |
| - | Indicates a file does not have extended security information. |

When the size of the files in a directory are listed, the **ls** command displays a total count of blocks, including indirect blocks.

Flags

Item	Description
-------------	--------------------

- | | |
|-----------|---|
| -A | Lists all entries except . (dot) and .. (dot-dot). |
| -a | Lists all entries in the directory, including the entries that begin with a . (dot). |
| -b | Displays nonprintable characters in an octal (\nnn) notation. |
| -c | Uses the time of last modification of the i-node for either sorting (when used with the -t flag) or for displaying (when used with the -l flag). This flag has no effect if it is not used with either the -t or -l flag, or both. |
| -C | Sorts output vertically in a multicolumn format. This is the default method when output is to a terminal. |
| -d | Displays only the information for the directory named. Directories are treated like files, which is helpful when using the -l flag to get the status of a directory. |
| -e | Displays the mode (including security information), number of links, owner, group, size (in bytes), time of last modification, and name of each file. If the file is a special file, the size field contains the major and minor device numbers. If the file is a symbolic link, the path name of the linked-to file is printed preceded by a -> (minus, greater than) sign. The attributes of the symbolic link are displayed. |
| -E | Lists extent attributes of a file that exists in Vxfs file systems, such as space reservation, fixed extent size, and extent allocation flag information. You must specify the -l flag with this flag; otherwise, the ls command ignores the -E flag and completes the execution. |
| -f | Lists the name in each slot for each directory specified in the <i>Directory</i> parameter. This flag turns off the -l , -t , -s , and -r flags, and turns on the -a flag. The order of the listing is the order in which entries appear in the directory. |
| -F | Puts a / (slash) after each file name if the file is a directory, an * (asterisk) if the file can be executed, an = (equal sign) if the file is a socket, a (pipe) sign if the file is a FIFO, and an @ for a symbolic link. Symbolic links that are named as operands are not followed unless you have specified the -H or -L flag. |
| -g | Displays the same information as the -l flag, except the -g flag suppresses display of the owner and symbolic link information. |
| -H | If a symbolic link referencing a file of type directory is specified on the command line, the ls command shall evaluate the file information and file type to be those of the file referenced by the link, and not the link itself; however, the ls command shall write the name of the link itself and not the file referenced by the link. |

- | Item | Description |
|------|--|
| -i | Displays the i-node number in the first column of the report for each file. If the file system has an internal snapshot, the .snapshot directory and all its contents do not have unique i-node numbers. |
| -k | Sets the block size for -s option and the per-directory block count written for -l , -n , -g , and -o options to 1024 bytes. |
| -L | Lists the file or directory contents that the link references. This is the default action. Symbolic links are followed. If the -l option is used, the -N option becomes the default, and no symbolic links are followed. When the -l option is used, only the -L option can override the -N default. |
| -l | (Lower case L) Displays the mode, number of links, owner, group, size (in bytes), and time of last modification for each file. If the file is a special file, the size field contains the major and minor device numbers. If the time of last modification is greater than six months ago, the time field is shown in the format month date year where as files modified within six months the time field is shown as month date time format.

If the file is a symbolic link, the path name of the linked-to file is printed preceded by a -> . The attributes of the symbolic link are displayed. The -n , -g , and -o flag overrides the -l flag. |
| | <p>Notes:</p> <ol style="list-style-type: none"> 1. A symbolically linked file is followed by an arrow and the contents of the symbolic link. 2. The performance of the ls command when used with the -l option can be improved by executing the mkpasswd command. This is helpful when a directory contains files owned by different users, such as the /tmp directory. |
| -m | Uses stream output format (a comma-separated series). |
| -n | Displays the same information as the -l flag, except that the -n flag displays the user and the group IDs instead of the user and group names. |
| -N | Does not follow symbolic links when determining the status of a file.

Note: If both the -L and -N options are used, the last one will dominate. Also, any time a symbolic link is given that includes a / (slash) as the final character, the link will automatically be followed regardless of any options used. |
| -o | Displays the same information as the -l flag, except the -o flag suppresses display of the group and symbolic link information. |
| -p | Puts a slash after each file name if that file is a directory. This is useful when you pipe the output of the ls command to the pr command, as follows: |
| | <pre>ls -p pr -5 -t -w80</pre> |
| -q | Displays nonprintable characters in file names as a ? (question mark). |
| -r | Reverses the order of the sort, giving reverse alphabetic or the oldest first, as appropriate. |
| -R | Lists all subdirectories recursively. |
| -s | Gives size in kilobytes (including indirect blocks) for each entry. |
| -S | Sorts with the primary key being file size (in decreasing order) and the secondary key being file name in the collating sequence (in increasing order). |
| -t | Sorts by time of last modification (latest first) instead of by name. For a symbolic link, the time used as the sort key is that of the symbolic link itself. |

Item	Description
-U	Displays similar information as the -l flag. Displays the mode (including security information, named extended attribute information and encryption information), number of links, owner, group, size (in bytes), time of last modification, and name of each file. If the file is a special file, the size field contains the major and minor device numbers. If the file is a symbolic link, the path name of the linked-to file is printed preceded by a -> (minus, greater than) sign. The attributes of the symbolic link are displayed.
-u	Uses the time of the last access, instead of the time of the last modification, for either sorting (when used with the -t flag) or for displaying (when used with the -l flag). This flag has no effect if it is not used with either the -t or -l flag, or both.
-x	Sorts output horizontally in a multi-column format.
-X	Prints long user names when used with other flags that display user names. The upper limit is determined by the max_logname ODM attribute in the PdAt and CuAt object classes. If a user name is greater than the max_logname attribute, it will be truncated to the number of characters as specified by the max_logname attribute, less one character.
-1	Forces output into one-entry-per-line format. This is the default when the output is not directed to a terminal.

Notes:

- If any of the **-l**, **-n**, **-s**, **-g**, or **-o** flag is specified, each file that is present in the directory is preceded by a status line that indicates the number of file system blocks occupied by files.
- If the **-k** flag is not specified along with any of the **-l**, **-n**, **-s**, **-g**, or **-o** flag, the status line indicates the number of file system blocks occupied by files in units of 512 bytes.
- If the **-k** flag is specified along with any of the **-l**, **-n**, **-s**, **-g**, or **-o** flag, the status line indicates the number of file system blocks occupied by files in units of 1024 bytes.
- Additionally, if necessary, the number of file system blocks occupied by files in the directory is rounded off to the next integral number of units.
- In the POSIX locale environment, the total %u\n is the output format which represents number of units in the directory.

Exit Status

This command returns the following exit values:

Item	Description
0	All files were written 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 list all files in the current directory, type:

```
ls -a
```

This lists all files, including . (dot), .. (dot-dot), and other files with names beginning with a dot.

2. To display detailed information, type:

```
ls -l chap1 .profile
```

This displays a long listing with detailed information about `chap1` and `.profile`.

3. To display detailed information about a directory, type:

```
ls -d -l . manual manual/chap1
```

This displays a long listing for the directories `.` and `manual`, and for the file `manual/chap1`. Without the `-d` flag, this would list the files in the `.` and `manual` directories instead of the detailed information about the directories themselves.

4. To list the files in order of modification time, type:

```
ls -l -t
```

This displays a long listing of the files that were modified most recently, followed by the older files.

5. To display detailed information with expanded user and group name, type:

```
ls -lX .profile
```

This displays a long listing with detailed information about `.profile`.

6. To display data about whether extended attributes are set for the files in the current directory, type:

```
ls -U
```

- For releases AIX 5.3 and earlier:

Example output:

```
-rwsr-x---+ 1 root system 28 Apr 29 03:23 only_aixc
-rwsr-x---E 1 root system 4 Apr 29 03:23 only_aixc_ea
-rw-r--r--E 1 root system 4 Apr 29 03:23 only_ea
-----+ 1 root system 265 Apr 29 03:23 only_nfs4
-----E 1 root system 64 Apr 29 03:23 only_nfs4_ea
-rw-r--r--- 1 root system 4 Apr 29 03:23 only_regular
```

- For releases AIX 6.1 and later:

Example output:

```
-rwsr-x---+ 1 root system 28 Apr 29 03:23 only_aixc
-rwsr-x---E 1 root system 4 Apr 29 03:23 only_aixc_ea
-rw-r--r--E 1 root system 4 Apr 29 03:23 only_ea
-----+ 1 root system 265 Apr 29 03:23 only_nfs4
-----E 1 root system 64 Apr 29 03:23 only_nfs4_ea
-rw-r--r--- 1 root system 4 Apr 29 03:23 only_regular
-rwxrwxr-xe 2 root system 256 May 25 16:27 encry_ex
```

7. To display information about the number of files system blocks in units of 512 bytes for the files in the current directory, type:

```
ls -li
```

Example output:

```
total 16
-rw-r--r-- 1 root system 22 Feb 05 05:29 sample1
-rw-r--r-- 1 root system 12 Feb 05 05:29 sample2
```

8. To display information about the number of files system blocks in units of 1024 bytes for the files in the current directory, type:

```
ls -lk
```

Example output:

```
total 8
-rw-r--r--  1 root    system    22 Feb 05 05:29 sample1
-rw-r--r--  1 root    system    12 Feb 05 05:29 sample2
```

Files

Item	Description
<code>/usr/bin/ls</code>	Contains the ls command.
<code>/etc/passwd</code>	Contains user IDs.
<code>/etc/group</code>	Contains group IDs.
<code>/usr/share/lib/terminfo/*</code>	Contains terminal information.

ls-secdapclntd Command

Purpose

The **ls-secdapclntd** command lists the status of the **secdapclntd** daemon process.

Syntax

`/usr/sbin/ls-secdapclntd`

Description

The **ls-secdapclntd** command lists the **secdapclntd** daemon status. The information returned includes the following:

- The LDAP server the **secdapclntd** daemon is talking to
- The LDAP server port number
- The version of the LDAP protocol used
- User base DN
- Group base DN
- System (id) base DN
- User cache size
- User cache size used
- Group cache size
- Group cache size used
- Cache time out (time to live) value
- **secdapclntd** to LDAP server heart beat interval
- Number of thread used by **secdapclntd** daemon
- Authentication mechanism in use
- Attribute search mode
- Default user attribute entry location
- Timeout period (seconds) for LDAP client requests to the server
- User objectclass used in the LDAP server
- Group objectclass used in the LDAP server

Security

A user with the **aix.security.ldap** authorization is authorized to use this command.

Example

1. To list the status of the **secdapclntd** daemon, type:

```
/usr/sbin/ls-secdapclntd
```

Files

Item	Description
/etc/security/ldap/ldap.cfg	Contains information needed by the secdapclntd daemon to connect to the server.

lsactdef Command

Purpose

Displays the action definitions of a resource or a resource class.

Syntax

To display the action definitions of a *resource*:

```
lsactdef [-p property] [-s i | o] [-e] [-l | -i | -t | -d | -D delimiter] [-x] [-h] [-TV] resource_class [ action1 [ action2 ... ] ]
```

To display the action definitions of a *resource class*:

```
lsactdef -c [-p property] [-s i | o] [-e] [-l | -i | -t | -d | -D delimiter] [-x] [-h] [-TV] resource_class [ action1 [ action2 ... ] ]
```

To display all resource class names:

```
lsactdef
```

Description

The `lsactdef` command displays a list of the action definitions of a resource or a resource class. By default, this command displays the action definitions of a *resource*. To see the action definitions of a *resource class*, specify the `-c` flag.

If you do not specify any actions on the command line, this command only displays actions that are defined as `public`. To override this default, use the `-p` flag or specify on the command line the names of the actions that have definitions you want to display.

To see the structured data definition that is required as input when this action is invoked, specify the `-s i` flag. To see the structured data definition linked with the output that results from invoking this action, specify the `-s o` flag.

By default, this command does not display action descriptions. To display action definitions and descriptions, specify the `-e` flag.

Flags

- c**
Displays the action definitions for *resource_class*.

-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.

-e
Specifies expanded format. Displays descriptions along with the action definitions.

-i
Specifies input format. Generates a template of *resource_data_input_file*. The output is displayed in long (stanza) format. The attribute's SD element data types are displayed as the value in the *attr=value* pairs. It is suggested that when you use this flag, the output of the `lsactdef` command be directed to a file. This flag overrides the **-s o** flag.

-l
Specifies "long" format — one entry per line. This is the default display format. If the `lsactdef` command is issued with the **-l** flag, but without a resource class name, the **-l** flag is ignored when the command returns the list of defined resource class names.

-p property
Displays actions with the specified *property*. By default, only the definitions for public actions are displayed. To display all action definitions regardless of the action property, use the **-p 0** flag.

Action properties:

0x0001
long_running

0x0002
public

A decimal or hexadecimal value can be specified for the property. To request the action definitions for all actions that have one or more properties, "OR" the properties of interest together and then specify the "OR"ed value with the **-p** flag. For example, to request the action definitions for all actions that are long_running or public, enter:

```
-p 0x03
```

-s i | o
Displays the structured data definition for the action input or action response.

i
Displays the action input structured data definitions. This is the default.

o
Displays the action response (output) structured data definitions.

-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

resource_class

Specifies the name of the resource class with the action definitions that you want to display. If *resource_class* is not specified, a list of all of the resource class names is displayed.

action1 [action2...]

Specifies one or more actions. If *resource_class* is specified, zero or more action names can be specified. If no actions are specified, all of the action definitions for *resource_class* are displayed. Enter specific action names to control which actions are displayed and in what order. Use blank spaces to separate action names.

Security

The user needs read permission for the *resource_class* specified in `lsactdef` to run `lsactdef`. 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. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

1. To list the names of all of the resource classes, enter:

```
lsactdef
```

The output will look like this:

```
class_name
"IBM.Association"
"IBM.AuditLog"
"IBM.AuditLogTemplate"
"IBM.Condition"
"IBM.EventResponse"
"IBM.Host"
"IBM.Program"
"IBM.Sensor"
"IBM.ManagedNode"
...
```

2. To list the public resource action definitions for resource class IBM®.AuditLog, enter:

```
lsactdef IBM.AuditLog
```

The output will look like this:

```
Resource Action Definitions for
class_name: IBM.AuditLog
action 1:
  action_name   = "GetRecords"
  display_name  = ""
  description   = ""
  properties    = {"public"}
  confirm_prompt = ""
  action_id    = 0
  variety_list  = {{1..1}}
  variety_count = 1
  timeout      = 0
action 2:
  action_name   = "DeleteRecords"
  display_name  = ""
  description   = ""
  properties    = {"public"}
  confirm_prompt = ""
  action_id    = 1
  variety_list  = {{1..1}}
  variety_count = 1
  timeout      = 0
....
```

3. To list the structured data definition required for invoking the action on resources in resource class IBM.AuditLog, action GetRecords, enter:

```
lsactdef -s i IBM.AuditLog GetRecords
```

The output will look like this:

```
Resource Action Input for: IBM.AuditLog
action_name GetRecords:
sd_element 1:
  element_name      = "MatchCriteria"
  display_name      = ""
  description        = ""
  element_data_type = "char_ptr"
  element_index     = 0
sd_element 2:
  element_name      = "IncludeDetail"
  display_name      = ""
  description        = ""
  element_data_type = "uint32"
  element_index     = 1
```

Location

`/opt/rsct/bin/lsactdef`

lsallq Command

Purpose

Lists the names of all configured queues.

Syntax

```
lsallq [ -c]
```

Description

The **lsallq** command lists the names of all configured queues contained in the `/etc/qconfig` file. By specifying the -c flag, this listing is displayed in colon format. This flag is used mainly by SMIT.

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

Flag

Item	Description
------	-------------

-c	Causes colon format output for use by SMIT.
-----------	---

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 all of the queue names in the `/etc/qconfig` file, enter:

```
lsallq
```

A listing similar to the following is displayed:

```
lp0  
lp1  
lp2
```

2. To list all configured queues in colon format, enter:

```
lsallq -c
```

A listing similar to the following is displayed:

```
lp0  
lp0:queue1  
lp0:queue2  
lp1
```

Files

Item	Description
<code>/usr/bin/lsallq</code>	Contains the lsallq command.
<code>/etc/qconfig</code>	Configuration file.

lsallqdev Command

Purpose

Lists all configured printer and plotter queue device names within a specified queue.

Syntax

```
lsallqdev [ -c] -qName
```

Description

The **lsallqdev** command lists all configured device names within a specified queue in the `/etc/qconfig` file.

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

Flags

Item	Description
<code>-q Name</code>	Specifies the queue name.
<code>-c</code>	Causes colon format output for use by SMIT.

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 all of the configured queue devices within the `lp0` queue in the `/etc/qconfig` file, enter:

```
lsallqdev -q lp0
```

A listing similar to the following is displayed:

```
lpd0
lpd1
lpd2
```

2. To list the names of all of the configured queue device within the `lp0` queue in the `/etc/qconfig` file in colon format, enter:

```
lsallqdev -q lp0 -c
```

A listing similar to the following is displayed:

```
lp0:lpd1
lp0:lpd2
```

Files

Item	Description
<code>/usr/bin/lsallqdev</code>	Contains the lsallqdev command.
<code>/etc/qconfig</code>	Configuration file.

lsarm command

Purpose

Displays Application Response Measurement (ARM) application and process usage information.

Syntax

```
lsarm -a [ -g ] [ -t ] [ -u ] [ ApplicationName ... ]
```

or

```
lsarm -p [ -a [ -g ] [ -t ] [ ProcessID ... ] ]
```

Description

The **lsarm** command displays information about applications registered with the operating system using the Application Response Measurement (ARM) APIs. The **-a** flag displays information about applications by application name. The **-p** option displays information about the applications used by a process.

Flags

Item	Description
-a	Displays application names.
-g	Displays group names associated with the application.

Item	Description
-p	Displays transaction names associated with the application.
-u	Displays process numbers using the application.
-t	Displays the applications used by a process.

Parameters

Item	Description
<i>ApplicationName</i>	Specifies a list of one or more applications for which the lsarm command should display information.
<i>ProcessID</i>	Specifies a list of one or more process IDs for which the lsarm command should display information.

Exit Status

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

Examples

1. To display all application names, type:

```
lsarm -a
```

2. To display group names, transaction class names, and process numbers using the 'database' application, type:

```
lsarm -a -g -t -u database
```

3. To display the process IDs for all processes using ARM applications, type:

```
lsarm -p
```

4. To display the application names, group names, and transaction names used by process 25038, type:

```
lsarm -p -a -g -t 25038
```

Location

/usr/ewl/sbin/lsarm

Related Information

lsassocmap Command

Purpose

Displays an association map.

Syntax

```
lsassocmap [-c association_class] [-h] [-TV] [endpoint...]
```

Description

The `lsassocmap` command displays the association classes available on a cluster, including the endpoints of each association. Names and endpoints of Common Information Model (CIM) association classes that are registered with the CIM resource manager are listed in table format, similar to the output of the `lscondresp` command.

If you specify the `lsassocmap` command without any parameters, it displays all of the association classes, endpoints, and roles. A *role* is the name of the class reference property in the association class definition. Roles can be used as parameters to the `-o` and `-R` flags of the `lsrsrassoc` command to filter output. See “[lsrsrassoc Command](#)” on page 358 for more information.

The `-c` flag limits the associations displayed to only those provided by a specific association class. You can specify any number of classes by using the *endpoint* parameter; only associations containing those classes as references (endpoints) are displayed.

Parameters

endpoint...

Specifies one or more endpoint classes. Only association classes containing references to one of the *endpoint* classes are displayed.

Flags

-c *association_class*

Displays associations for *association_class*.

-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 to standard output.

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 are written to standard output.

Standard error

When the `-T` flag is specified, this command 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 The specified association class cannot be found.

Implementation specifics

This command is part of the `rsct.exp.cimim` fileset, in the `rsct.exp` package on the AIX Expansion Pack and the Reliable Scalable Cluster Technology (RSCT) package for the Linux® operating system.

Location

Item	Description
<code>/opt/rsct/bin/lsassocmap</code>	

Examples

To display associations that are available in a cluster, enter the following command:

```
lsassocmap
```

The following output is displayed for the AIX platform:

Association Class	Role 1	Associator 1	Role 2	Associator 2	Node
cimv2.IBMAIX_RunningOS	Antecedent	IBMAIX_OperatingSystem	Dependent	IBMAIX_ComputerSystem	c175nf14
cimv2.IBMAIX_OSPProcess	GroupComponent	IBMAIX_OperatingSystem	PartComponent	IBMAIX_UnixProcess	c175nf14
cimv2.IBMAIX_CSPProcessor	GroupComponent	IBMAIX_ComputerSystem	PartComponent	IBMAIX_Processor	c175nf14
cimv2.IBMAIX_HostedFileSystem	GroupComponent	IBMAIX_ComputerSystem	PartComponent	CIM_FileSystem	c175nf14

The following output is displayed for other platforms:

Association Class	Role 1	Associator 1	Role 2	Associator 2	Node
cimv2.Linux_RunningOS	Antecedent	Linux_OperatingSystem	Dependent	Linux_ComputerSystem	c175nf14
cimv2.Linux_OSPProcess	GroupComponent	Linux_OperatingSystem	PartComponent	Linux_UnixProcess	c175nf14
cimv2.Linux_CSPProcessor	GroupComponent	Linux_ComputerSystem	PartComponent	Linux_Processor	c175nf14
cimv2.Linux_HostedFileSystem	GroupComponent	Linux_ComputerSystem	PartComponent	CIM_FileSystem	c175nf14

lsattr Command

Purpose

Displays attribute characteristics and possible values of attributes for devices in the system.

Syntax

```
lsattr { -D [-O] | -E [-O] | -P [-O] | -F Format [-Z Character] } -l Name [-a Attribute] ... [-f File] [-h] [-H]
```

```
lsattr { -D [-O] | -F Format [-Z Character] } { [-c Class] [-s Subclass] [-t Type] } [-a Attribute] ... [-f File] [-h] [-H]
```

```
lsattr -R { -l Name | [-c Class] [-s Subclass] [-t Type] } -a Attribute [-f File] [-h] [-H]
```

```
lsattr -l Name { -o operation [ ... ] } -F Format [-Z Character] [-f File] [-h] [-H]
```

```
lsattr { [-c Class] [-s Subclass] [-t Type] } { -o operation [ ... ] } -F Format [-Z Character] [-f File] [-h] [-H]
```


Description

The **lsattr** command displays information about the attributes of a specific device or type of device. If you do not specify the device logical name with the **-l Name** flag, you must use a combination of one or all of the **-c Class**, **-s Subclass**, and **-t Type** flags to uniquely identify the predefined device.

You must specify one of the following flags with the **lsattr** command:

Item	Description
-D	Displays default values.
-E	Displays effective values (valid only for customized devices that are specified with the -l flag).
-F Format	Specifies the user-defined format.
-P	Displays device values when the device was last configured.
-R	Displays the range of legal values.

When you display the effective values of the attributes for a customized device, the information is obtained from the Configuration database, not the device. The database values reflect how the device is configured unless it is reconfigured with the **chdev** command by using the **-P** or **-T** flag. If the reconfiguration occurs, the information that is displayed by the **lsattr** command might not correctly indicate the current device configuration until after the next system boot.

If you use the **-D** or **-E** flag, the output defaults to the values for the attribute's name, value, description, and user-settable strings, unless it is also used with the **-O** flag.

The **-P** flag displays the attribute values when the device was last configured, or before modifying any of its attributes by using the **chdev** command with the **-P** or **-T** flag.

The **-O** flag displays the names of all the attributes that are specified, separated by colons. On the next line, the **-O** flag displays all of the corresponding attribute values, separated by colons. The **-H** flag can be used with either the **-D**, **-E**, or **-F** flag to display headers above the column names. You can define the format of the output with a user-specified format by using the **-F Format** flag, where the *Format* parameter is a quoted list of column names, separated by non-alphanumeric characters or white space. If the **-F Format** flag is specified, the **-Z Character** flag can also be specified to change the default record separator from a `newline` character to the indicated *Character*.

The **lsattr** command can display "*operation*" information from the Extended Predefined Attribute (**PdAtXtd**) object class. The operation information is accessed through the **-o operation** flag. The **-o operation** flag and the **-a attribute** flag cannot be specified in the same invocation of the **lsattr** command. The **-o operation** flag is also not valid with the **-R** flag. When the **-o operation** flag is specified, only fields from the **PdAtXtd** object class can be specified with the **-F Format** flag.

You can supply the flags either on the command line or by using the specified **-f File** flag.

Flags

Item	Description
-a Attribute	Displays information for the specified attributes of a specific device or type of device. You can use one -a flag for each attribute name or multiple attribute names. If you use one -a flag for multiple attribute names, the list of attribute names must be enclosed in quotation marks with spaces between the names. If you use the -R flag, you must specify only one -a flag with only one attribute name. If you do not specify either the -a or -R flag, the lsattr command displays all information for all attributes of the specified device. The -a Attribute flag cannot be used with the -o Operation flag. This combination of flags causes the lsattr command to exit with an error message.

Item	Description
-c <i>Class</i>	Specifies a device class name. This flag can be used to restrict the output to devices of a specified class. This flag cannot be used with the -E or -l flag.
-D	Displays the attribute names, default values, descriptions, and user-settable flag values for a specific device when it is not used with the -O flag. The -D flag displays only the attribute name and default value in colon format when it is used with the -O flag. This flag can be used with any combination of the -c , -s , and -t flags that uniquely identifies a device from the Predefined Devices object class, or with the -l flag. This flag cannot be used with the -E , -F , or -R flag.
-E	Displays the attribute names, current values, descriptions, and user-settable flag values for a specific device when it is not used with the -O flag. The -E flag displays only the attribute name and current value in colon format when it is used with the -O flag. This flag cannot be used with the -c , -D , -F , -R , -s , or -t flag.
-f <i>File</i>	Reads the necessary flags from the <i>File</i> parameter.
-F <i>Format</i>	Displays the output in a user-specified format, where the <i>Format</i> parameter is a quoted list of column names separated by nonalphanumeric characters or white space. If white space is used as the separator, the lsattr command displays the output in aligned columns. Only column names from the Predefined Attributes (PdAt), Customized Attributes (CuAt), and the Extended Predefined Attributes (PdAtXtd) object classes can be specified. In addition to the column names, there are two special purpose names that can be used: the name <i>description</i> can be used to obtain a display of attribute descriptions and <i>user_settable</i> can be used to determine whether an attribute can be changed. This flag cannot be used with the -E , -D , -O , or -R flag.
-H	Displays headers above the column output. The -O and -R flags take precedence over the -H flag.
-h	Displays the command usage message.
-l <i>Name</i>	Specifies the device logical name in the Customized Devices object class whose attribute names or values you want displayed.
-o <i>Operation</i>	<p>Displays information for the specified operations of a specific device or type of device. You can use one -o flag for each operation name or multiple operation names.</p> <p>If you use one -o flag for multiple operation names, the list of operation names must be enclosed in quotation marks with spaces between the names.</p> <p>Wildcard characters can also be used for the operation name. The valid set of wildcard characters is the same set that is used by the odmget command. All operations that are associated with a specific device, or type of device, can be displayed by using an operation value of "?*". The -o <i>Operation</i> flag cannot be used with the -a <i>attribute</i> flag or the -R flag. Any combination of these flags causes the lsattr command to exit with an error message.</p>
-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 specified and default values when the -D flag is specified. This flag cannot be used with the -F and -R flags.

Item	Description
-P	Displays the attribute names, values, descriptions, and user-settable flag values for a specific device when it is not used with the -O flag. The values that are displayed are those values with which the device was configured, before any of the device attributes were modified by using the chdev command with the -P or -T flag. When the -P flag is used with the -O flag, the -P flag displays only the attribute name and value in colon-separated format. This flag can be used with any combination of the -c , -s , and -t flags that uniquely identifies a device from the Predefined Devices object class, or with the -l flag. This flag cannot be used with the -D , -E , -O , or -R flag.
-R	Displays the legal values for an attribute name. The -R flag cannot be used with the -D , -E , -F and -O flags, but can be used with any combination of the -c , -s , and -t flags that uniquely identifies a device from the Predefined Devices object class, or with the -l flag. The -R flag displays the list attribute values in a vertical column as follows:
	<pre>Value1 Value2 . . ValueN</pre>
	The -R flag displays the range attribute values as $x . . n(+i)$ where x is the start of the range, n is the end of the range, and i is the increment.
-s Subclass	Specifies a device subclass name. This flag can be used to restrict the output to that for devices of a specified subclass. This flag cannot be used with the -E or -l flag.
-t Type	Specifies a device type name. This flag can be used to restrict the output to that for devices of a specified class. This flag cannot be used with the -E or -l flag.
-Z Character	The -Z Character flag is used with programs that must deal with ODM fields that might have embedded new line characters. The -Z Character flag is used to change the record separator character for each record, or line, of output generated. The new record separator is specified by using the <i>Character</i> argument to this flag. The -Z Character flag is only relevant when the -F Format flag is specified. The -Z Character flag cannot be used with the -D , -E , -O , or -R 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 that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

- To list the current attribute values for the `rmt0` tape device, enter the following command:

```
lsattr -l rmt0 -E
```

The system displays a message similar to the following output:

```
mode          yes    Use DEVICE BUFFERS during writes    True
block_size    1024   BLOCK size (0=variable length)     True
extfm         yes    Use EXTENDED file marks             True
ret           no     RETENSION on tape change or reset   True
density_set_1 37     DENSITY setting #1                  True
density_set_2 36     DENSITY setting #2                  True
compress      yes    Use data COMPRESSION                True
```

size_in_mb	12000	Size in Megabytes	False
ret_error	no	RETURN error on tape change or reset	True

2. To list the default attribute values for the `rmt0` tape device, enter the following command:

```
lsattr -l rmt0 -D
```

The system displays a message similar to the following output:

mode	yes	Use DEVICE BUFFERS during writes	True
block_size	1024	BLOCK size (0=variable length)	True
extfm	yes	Use EXTENDED file marks	True
ret	no	RETENSION on tape change or reset	True
density_set_1	37	DENSITY setting #1	True
density_set_2	36	DENSITY setting #2	True
compress	yes	Use data COMPRESSION	True
size_in_mb	12000	Size in Megabytes	False
ret_error	no	RETURN error on tape change or reset	True

3. To list the current value of the `bus_intr_lvl` attribute for the `scsi0` SCSI adapter, enter the following command:

```
lsattr -l scsi0 -a bus_intr_lvl -E
```

The system displays a message similar to the following output:

```
bus_intr_lvl 1 Bus interrupt level False
```

4. To list the possible values of the `login` attribute for the `tty0` tty device, enter the following command:

```
lsattr -l tty0 -a login -R
```

The system displays a message similar to the following output:

```
enable
disable
share
delay
```

5. To list the default attribute values for an IBM 4340 parallel printer, enter the following command:

```
lsattr -c printer -s parallel -t ibm4340 -D
```

The system displays a message similar to the following output:

ptop	600	Printer TIME OUT period	True
line	60	Number of LINES per page	True
col	80	Number of COLUMNS per page	True
ind	0	Number of columns to INDENT	True
plot	no	Send all characters to printer UNMODIFIED	True
backspace	yes	Send BACKSPACES	True
cr	yes	Send CARRIAGE RETURNS	True
form	yes	Send FORM FEEDS	True
lf	yes	Send LINE FEEDS	True
addcr	yes	Add CARRIAGE RETURNS to LINE FEEDS	True
case	no	Convert lowercase to UPPERCASE	True
tabs	yes	EXPAND TABS on eight position boundaries	True
wrap	no	WRAP CHARACTERS beyond the specified width	True
mode	no	Return on ERROR	True
interface	standard	Type of PARALLEL INTERFACE	True
autoconfig	available	STATE to be configured at boot time	True
busy_delay	0	Microseconds to delay between characters	True

6. To list the possible values of the `ptop` attribute for an IBM 4340 parallel printer, enter the following command:

```
lsattr -c printer -s parallel -t ibm4340 -a ptop -R
```

The system displays a message similar to the following output:

```
1...1000 (+1)
```

7. To list the current attribute values for the `rmt0` tape device in colon-separated format, enter the following command:

```
lsattr -l rmt0 -E -O
```

The system displays a message similar to the following output:

```
#mode:block_size:extfm:ret:density_set_1:density_set_2:compress:size_in_mb:ret_error
yes:1024:yes:no:37:36:yes:12000:no
```

8. To display system attributes, enter the following command:

```
lsattr -E -l sys0
```

The system displays output similar to the following output:

keylock	normal	State of system keylock at boot time	False
maxbuf	20	Maximum number of pages in block I/O BUFFER CACHE	True
maxmbuf	0	Maximum Kbytes of real memory allowed for MBUFS	True
maxuproc	128	Maximum number of PROCESSES allowed per user	True
autorestart	false	Automatically REBOOT system after a crash	True
iostat	false	Continuously maintain DISK I/O history	True
realmem	4194304	Amount of usable physical memory in Kbytes	False
conslogin	enable	System Console Login	False
fwversion	IBM,SPH00221	Firmware version and revision levels	False
maxpout	0	HIGH water mark for pending write I/Os per file	True
minpout	0	LOW water mark for pending write I/Os per file	True
fullcore	false	Enable full CORE dump	True
pre430core	false	Use pre-430 style CORE dump	True
ncargs	256	ARG/ENV list size in 4K byte blocks	True
rtasversion	1	Open Firmware RTAS version	False
modelname	IBM,7044-270	Machine name	False
systemid	IBM,011037D1F	Hardware system identifier	False
boottype	disk	N/A	False
SW_dist_intr	false	Enable SW distribution of interrupts	True
cpuguard	disable	CPU Guard	True
frequency	93750000	System Bus Frequency	False

Note: The same information is available in a more readable format by using SMIT. Select the **System Environments** -> **Change / Show Characteristics of Operating Systems** options to view this information.

Files

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

lsaudrec Command

Purpose

Lists records from the audit log.

Syntax

```
lsaudrec [-l] [-a | -n node_name1[, node_name2...] [-S subsystem_name]
[-s selection_string] [-x] [-h] [field_name1 [field_name2...]]
```

Description

The `lsaudrec` command is used to list 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 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 displayed. A selection string is an expression that is made up of field names, constants, and operators. The syntax of a selection string is similar to an expression in the C programming language or the SQL "where" clause. The selection string is matched against each record using the referenced fields of each record to perform the match. Any records that match are displayed. The selection string is specified with the -s flag. For information on how to specify selection strings, see the *Administering RSCT* guide.

You can also specify field names as parameters to this command to choose which fields are displayed and the order in which they are displayed. The common field names are:

Field	Description
Time	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 below for specifying the time in more user-friendly formats.
Subsystem	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	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	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	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 are, from left to right: *mm* = month, *dd* = day, *hh* = hour, *mm* = minutes, *yyyy* = year. For example, #010523042004 corresponds to January 5, 11:04 PM, 2004. The fields can be omitted from right to left. If not present, the following defaults are used: year = the current year, minutes = 0, hour = 0, day = 1, and month = the current month.

#-mmddhhmmyyyy

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 can be omitted starting from the right and are replaced by 0.

The audit records considered for display 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 display and matched against the selection string can be restricted to a set of specific nodes by using the -n flag. If this flag is specified, the search is limited to the set of nodes listed. Otherwise, the search is performed for all nodes defined within the current management scope, as determined by the CT_MANAGEMENT_SCOPE environment variable.

The audit records are displayed in a table. Field names specified as parameters control which fields are displayed and the order in which they appear on each line. By default, the columns displayed are: the date and time, the subsystem name that generated the record, the severity of the situation, and the subsystem-specific message that describes the situation. If the management scope is not local, the node name is displayed in the first column.

Flags

-l

Indicates that long output should be produced. Long output includes subsystem-specific fields that are not included in the formatted message text.

-a

Specifies that records from all nodes in the domain are to be displayed. If both the -n and the -a flags are omitted, records from the local node only are displayed.

-n node_name1[,node_name2]...

Specifies the list of nodes containing audit log records that will be examined and displayed 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 are displayed.

-S subsystem_name

Specifies a subsystem name. If this flag is present, only records identified by *subsystem_name* are considered for display. The records displayed 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. All records that match the selection string will be displayed. If the selection string contains any spaces, it must be enclosed in single or double quotation marks. For information on how to specify selection strings, see the *Administering RSCT* guide.

The names of fields in 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 omitted, the records that are displayed will depend on the -S flag. If the -S flag is omitted, all records from the audit log are displayed. Otherwise, all records for the subsystem identified by the -S flag are displayed.

-x

Excludes the header (suppresses header printing).

-h

Writes the command's usage statement to standard output.

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 list records from an audit log when the -S flag is omitted, you must have read access to the target resource class on each node from which records are to be listed. When the -S flag is specified, you must have read access to the audit log resource corresponding to the subsystem identified by the -S flag on each node from which records are to be listed.

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 listed. 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) fileset for AIX.

Standard Output

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

Examples

1. To list all records in the audit log on every node in the current management scope as determined by the `CT_MANAGEMENT_SCOPE` environment variable, enter:

```
lsaudrec
```

2. To list all records that were logged in the last hour on every node in the current management scope as determined by the `CT_MANAGEMENT_SCOPE` environment variable, enter:

```
lsaudrec -s "Time > #-000001"
```

3. To list the time and sequence number of every record in the audit log for the subsystem `abc` on nodes `mynode` and `yournode`, enter:

```
lsaudrec -n mynode,yournode -S abc Time SequenceNumber
```

4. To list the records that are generated by the event-response resource manager (ERRM), enter:

```
lsaudrec -SERRM
```

5. To list the records that are related to a condition called `Condition1`, enter:

```
lsaudrec -SERRM -s"ConditionName=='Condition1'"
```

6. To list the records that are related to an event from `Condition1`, enter:

```
lsaudrec -SERRM -s"ConditionName=='Condition1' && Etype==91"
```

7. To list the records that are related to a rearm event from **Condition1**, enter:

```
lsaudrec -SERRM -s"ConditionName=='Condition1' && Etype==92"
```

8. To list the sensor resource manager records in the audit log on the local node, enter:

```
lsaudrec -SSSRM
```

The output will look like this:

Time	Subsystem	Category	Description
11/10/05 21:52:32	SSRM	Error	The Command /SENSOR/sensor.ksh 1 in Sensor SENSOR_NOUSER_1 execution fails.
11/10/05 21:52:36	SSRM	Error	The Command /SENSOR/sensor.nocmd 1 in Sensor SENSOR_NOCMD_1 exits with error 127.

9. To list, in long format, the sensor resource manager records in the audit log on the local node, enter:

```
lsaudrec -l -SSRM
```

The output will look like this:

```
Time          = 11/10/05 21:52:32 243097
Subsystem     = SSRM
Category      = Error
Description   = The Command /SENSOR/sensor.ksh 1 in Sensor SENSOR_NOUSER_1 execution fails.
ErrorMsg      = 2645-202 The user name "guest" that was specified for running the command does not exist.

Time          = 11/10/05 21:52:36 361726
Subsystem     = SSRM
Category      = Error
Description   = The Command /SENSOR/sensor.nocmd 1 in Sensor SENSOR_NOCMD_1 exits with error 127.
StandardOut   =
StandardErr   = ksh: /u/diane/drmc/scripts/SENSOR/sensor.nocmd: not found
```

10. To list error records only, enter:

```
lsaudrec -s"Category=1"
```

Location

`/opt/rsct/bin/lsaudrec`

lsauth Command

Purpose

Displays user and system-defined authorization attributes.

Syntax

```
lsauth [-R load_module] [-C] [-f] [-a List] {ALL | Name [,Name] ...}
```

Description

The **lsauth** command displays attributes of user-defined and system-defined authorizations from the authorization database. The command can be used to list attributes of all authorizations or specific authorizations. By default, the **lsauth** command displays all authorization attributes. To view selected attributes, use the **-a** *List* flag. If one or more attributes cannot be read, the **lsauth** command lists the information that is available.

By default, the **lsauth** command lists the attributes of each authorization on one line. It displays attribute information in the form of *Attribute = Value*, each separated by a blank space. To list the authorization attributes in stanza format, use the **-f** flag. To list the information as colon-separated records, use the **-C** flag.

If the system is configured to use multiple domains for the authorization database, the authorizations, as specified by the *Name* parameter, are searched from the domains in the order specified by the **secorder** attribute of the authorizations stanza in the `/etc/nscontrol.conf` file. If duplicate entries exist in multiple domains, only the first entry instance is listed. Use the **-R** flag to list the authorizations from a specific domain.

Flags

Item	Description
-a <i>List</i>	<p>Lists the attributes to display. The <i>List</i> parameter requires a blank space between attributes to list multiple attributes. If you specify an empty list, only the authorization names are displayed. The <i>List</i> parameter can include any attribute defined in the chauth command, in addition to the following two attributes:</p> <p>description The text description of the authorization as indicated by the dfltmsg, msgcat, msgset and msgnum attributes for the authorization.</p> <p>roles A comma-separated list of roles containing the specified authorization in their authorization set.</p>
-C	<p>Displays the authorization attributes in colon-separated records, as follows:</p> <pre>#authorization:attribute1:attribute2: ... authorization:value1:value2: ... authorization2:value1:value2: ...</pre> <p>The output is preceded by a comment line that has details about the attribute represented in each colon-separated field. If you specify the -a flag, the order of the attributes matches the order specified in the -a flag. If an authorization does not have a value for a given attribute, the field is still displayed but is empty. The last field in each entry is ended by a newline character rather than a colon.</p>
-f	<p>Displays the output in stanzas, with each stanza identified by an authorization name. Each <i>Attribute = Value</i> pair is listed on a separate line:</p> <pre>Authorization: attribute1=value attribute2=value attribute3=value</pre>
-R <i>load_module</i>	<p>Specifies the loadable module to list authorizations from.</p>

Parameters

Item	Description
ALL	<p>Specifies to list attributes from all authorizations.</p>
<i>Name</i>	<p>Specifies the authorization name to list. Optionally, a wild card (*) can be used at the end of a name to list an entire hierarchy. The entire string specified before the wild card must be a valid authorization name.</p>

Security

The **lsauth** command is a privileged command. You must assume a role that has the following authorization to run the command successfully.

Item	Description
aix.security.auth.list	<p>Required to run the command.</p>

Files Accessed

Item	Description
File	Mode
/etc/security/authorizations	r

Examples

1. To display all attributes of the custom authorization, use the following command:

```
lsauth custom
```

All the attribute information appears, with each attribute separated by a blank space.

2. To display all attributes of the custom authorization from LDAP, use the following command:

```
lsauth -R LDAP custom
```

All the attribute information appears, with each attribute separated by a blank space.

3. To display the authorization ID and description for the custom authorization in stanza format, use the following command:

```
lsauth -f -a id description custom
```

Information similar to the following appears:

```
custom:
  id=11000
  description="Custom Authorization"
```

4. To display the **msgcat**, **msgset** and **msgnum** attributes for the custom.test authorization in a colon format, use the following command:

```
lsauth -C -a msgcat msgset msgnum custom.test
```

Information similar to the following example appears:

```
#name:ID:msgcat:msgset:msgnum
custom.test:12000:custom_auths.cat:5:24
```

5. To display the description for the entire authorization hierarchy that begins with aix.security, use the following command:

```
lsauth -a description aix.security.*
```

The aix.security authorization and all its children are listed with one authorization per line and a space between the authorization name and the description attribute.

lsauthent Command

Purpose

Lists the authentication methods currently configured on the system.

Syntax

lsauthent

Description

The **lsauthent** command calls the **get_auth_method** subroutine in the **libauthm.a** library, translates a list of authentication methods returned, and prints the authentication methods configured to **stdout**. Each authentication method is outputted on a separate line.

The authentication methods are listed in the order in which they are configured. If none of the authentication methods are configured, **lsauthent** returns without printing anything.

The **lsauthent** command writes an error message to **stderr** and returns a -1 if **get_auth_method** fails.

Examples

If all of the authentication methods are configured as:

```
lsauthent
```

the output would consist of:

```
Kerberos 5  
Kerberos 4  
Standard AIX
```

lsC2admin Command

Purpose

Display the name of the current C2 System Administrative Host.

Syntax

lsC2admin

Description

The **lsC2admin** command displays the name of the administrative host. An administrative host must have been defined, and the system must have been installed in C2 mode for this command to operate successfully.

Exit Status

- 0** The administrative host information has been successfully displayed.
- 1** This system was not installed with C2 security.
- 2** This system has not been initialized to operate in C2 mode.
- 3** An error occurred while displaying the name of the administrative host.

Files

Item	Description
/usr/sbin/lsC2admin	Contains the lsC2admin command.

lsCCadmin Command

Purpose

Display the name of the current Common Criteria enabled System Administrative Host.

Syntax

lsCCadmin

Description

The **lsCCadmin** command displays the name of the administrative host. An administrative host must have been defined, and the system must have been installed in Common Criteria enabled mode for this command to operate successfully.

Exit Status

- 0** The administrative host information has been successfully displayed.
- 1** This system was not installed with Common Criteria enabled security.
- 2** This system has not been initialized to operate in Common Criteria enabled mode.
- 3** An error occurred while displaying the name of the administrative host.

Files

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

lscfg Command

Purpose

Displays configuration, diagnostic, and vital product data (VPD) information about the system.

Syntax

To Display Specific Data on all Systems

lscfg [**-v**] [**-p**] [**-s**] [**-l** *Name*]

Description

If you run the **lscfg** command without any flags, it displays the name, location and description of each device found in the current Customized VPD object class that is a child device of the **sys0** object. It will not display any device that has been marked `missing` in the Customized Device Object Class. The list is sorted by parent, child, and device location. Information on a specific device can be displayed with the **-l** flag.

Use the **lscfg** command to display vital product data (VPD) such as part numbers, serial numbers, and engineering change levels from either the Customized VPD object class or platform specific areas. Not all devices contain VPD data.

VPD data that is preceded by ME signifies that the VPD data was entered manually using a diagnostic service aid. For some devices, the vital product data is collected automatically from the devices through methods and added to the Customized VPD object class.

If you run the **lscfg** command with the **-p** flag, it displays device information stored in the platform specific data areas. When used with the **-v** flag, VPD data stored for these devices is also displayed. This information is obtained on a Common Hardware Reference Platform (CHRP) system from the open firmware device tree.

Item	Description
-l Name	Displays device information for the named device.
-p	Displays the platform-specific device information.
-v	Displays the VPD found in the Customized VPD object class. Also, displays platform specific VPD when used with the -p flag.
-s	Displays the device description on a separate line from the name and location.

Examples

1. To display the system configuration, enter:

```
lscfg
```

The system displays a message similar to the following:

```
INSTALLED RESOURCE LIST
```

The following resources are installed on the machine:

```
+/- = Added or deleted from Resource List.
*   = Diagnostic support not available.

+ indicates the resource has been added to the Diagnostic Resource List. The resource is
added to the Diagnostic Resource list by default when the diagnostic fileset is installed. A
resource must be in the Diagnostic Resource List before diagnostics tasks can be
performed on the resource.

- indicates the resource was deleted from the Diagnostic Resource List.

The resource can be added or deleted from the Resource List by running the diag
command, and using the Task Selection menu to select either Add Resource to Resource
List, or Delete Resource from Resource List.

Diagnostic support for a resource, indicated by the + character is not necessarily
inclusive of all diagnostic tasks. Some resources are only supported with a subset of
diagnostic task, and that subset might or might not include the Run Diagnostics Task.

Model Architecture: chrp
Model Implementation: Multiple Processor, PCI bus

+ sys0                               System Object
+ sysplanar0                          System Planar
+ mem0                                 Memory
+ L2cache0                             L2 Cache
+ proc0                                U1.1-P1-C1      Processor
* pci3                                 U0.2-P1       PCI Bus
+ scsi0                                U0.1-P1/Z1    Wide/Ultra-2 SCSI I/O Controller
+ rmt0                                 U1.1-P1/Z1-A3 SCSI 4mm Tape Drive (12000 MB)
+ cd0                                  U1.1-P1/Z1-A5 SCSI Multimedia CD-ROM Drive (650 MB)
+ hdisk0                               U1.1-P1/Z1-A9 16 Bit LVD SCSI Disk Drive (4500 MB)
+ fd0                                  U0.1-P1-D1    Diskette Drive
..
..
```

2. To display the system configuration with the device description on a separate line, enter:

```
lscfg -s
```

The system displays a message similar to the following:

```
INSTALLED RESOURCE LIST
```

The following resources are installed on the machine:

```
+/- = Added or deleted from Resource List.
*   = Diagnostic support not available.

Model Architecture: chrp
Model Implementation: Multiple Processor, PCI bus

+ indicates the resource has been added to the Diagnostic Resource List. The resource is
added to the Diagnostic Resource list by default when the diagnostic fileset is installed. A
resource must be in the Diagnostic Resource List before diagnostics tasks can be
performed on the resource.

- indicates the resource was deleted from the Diagnostic Resource List.

The resource can be added or deleted from the Resource List by running the diag
command, and using the Task Selection menu to select either Add Resource to Resource
List, or Delete Resource from Resource List.

Diagnostic support for a resource, indicated by the + character is not necessarily
inclusive of all diagnostic tasks. Some resources are only supported with a subset of
diagnostic task, and that subset might or might not include the Run Diagnostics Task.

+ sys0
    System Object
+ sysplanar0
    System Planar
+ mem0
    Memory
+ L2cache0
    L2 Cache
+ proc0
    Processor      U5734.100.1234567-P1-C1
+ proc1
    Processor      U5734.100.1234567-P1-C2
+hdisk0
    16 Bit LVD SCSI Disk Drive (4500 MB) U5734.100.1234567-P1-D9
+fd0
    Diskette Drive U5734.100.1234567-P1-D1

::
::
```

3. To display the name, location, and description for devices specified by the logical name proc without VPD, enter:

```
lscfg -lproc\*
```

The system displays information for all devices with logical names beginning with proc, as follows:

```
proc0      U1.1-P1-C1 Processor
proc1      U1.1-P1-C1 Processor
proc2      U1.1-P1-C1 Processor
proc3      U1.1-P1-C1 Processor
proc4      U1.1-P1-C2 Processor
proc5      U1.1-P1-C2 Processor
proc6      U1.1-P1-C2 Processor
proc7      U1.1-P1-C2 Processor
```

4. To display the VPD for a specific device specified by the logical name ent0, enter:

```
lscfg -v -l ent0
```

The system displays the following:

```
ent0      U0.1-P1-I2/E1 Gigabit Ethernet-SX PCI Adapter (14100401)
          Network Address.....0004AC7C00C4
          Displayable Message.....Gigabit Ethernet-SX PCI Adapter (14100401)
          EC Level.....E77998
```



```
Part Number.....07L8916
FRU Number.....07L8918
Device Specific.(YL).....U0.1-P1-I2/E1
```

5. To display the VPD in the open firmware device tree for the corresponding node to the ent0 device, enter:

```
lscfg -vp -lent0
```

The following is displayed:

```
ent0          U0.1-P1-I2/E1  Gigabit Ethernet-SX PCI Adapter (14100401)

Network Address.....0004AC7C00C4
Displayable Message.....Gigabit Ethernet-SX PCI Adapter (14100401)

EC Level.....E77998
Part Number.....07L8916
FRU Number.....07L8918
Device Specific.(YL).....U0.1-P1-I2/E1

PLATFORM SPECIFIC

Name: ethernet
Model: Galaxy, EtherLink 1000-SX-IBM
Node: ethernet@1
Device Type: network
Physical Location: U0.1-P1-I2/E1
```

Files

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

lscifscred Command

Purpose

Lists the server or user entries stored in the `/etc/cifs_fs/cifscred` file.

Syntax

```
lscifscred [-h RemoteHost] [-u user]
```

Description

The `lscifscred` command lists all of the server or user entries that have passwords stored in the `/etc/cifs_fs/cifscred` file.

Flags

Item	Description
<code>-h <i>RemoteHost</i></code>	Lists credentials matching the given remote host (CIFS server) only.
<code>-u <i>user</i></code>	Lists credentials matching the given user name only.

Exit Status

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

Examples

1. To list all server and user entries on a server named `server1`, enter:

```
lscifscred -h server1
```

Location

/usr/sbin/lscifscred

lscifsmnt Command

Purpose

Lists the CIFS mounts defined in the `/etc/filesystems` file.

Syntax

```
lscifsmnt [-c | -l | -p] [FileSystem]
```

Description

The `lscifsmnt` command lists the specified CIFS mounts that are defined in the `/etc/filesystems` file.

Flags

Item	Description
-c	Specifies that the CIFS mount be listed in colon delimited format.
-l	Specifies that the CIFS mount be listed in standard format with each field separated by whitespace. This is the default.
-p	Specifies that the CIFS mount be listed in pipe delimited format.

Parameters

Item	Description
<i>FileSystem</i>	Specifies which file system to list the characteristics of. The default is to list all CIFS file systems.

Exit Status

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

Examples

1. To list all CIFS mounts defined in `/etc/filesystems`, enter:

```
lscifsmnt
```

Location

`/usr/sbin/lscifsmnt`

lsclass Command

Purpose

List Workload Management classes and their limits.

Syntax

```
lsclass [ -C | -D | -f ] [ -r ] [ -d Config_Dir ] [ -S SuperClass ] [ Class ]
```

Description

The **lsclass** command, with no argument, returns the list of superclasses, one per line. With a class name as argument, it prints the class. The subclasses can be displayed with the **-r** (recursive) flag, or with the **-S** *Superclass* flag.

When WLM is started, if an empty string is passed as the name of the configuration with the **-d** flag, **lsclass** lists the classes defined in the in-core WLM data structures.

The **lsclass** command does not require any special level of privilege and is accessible for all users.

Note: If this command is given a set of time-based configurations (either specified with the **-d** flag, or because the current configuration is a set), the **lsclass** command returns the classes of the regular configuration which applies (or would apply) at the time the command is issued.

Flags

Item	Description
-C	Displays the class attributes and limits in colon-separated records, as follows: <pre>lsclass -C myclass #name:description:tier:inheritance:authuser:authgroup: adminuser:admingroup:rset:CPUshares:CPUmin: CPUsoftmax:CPUhardmax:memoryshares:memorymin: memorysoftmax:memoryhardmax:diskIOshares:diskIOmin: diskIOsoftmax:diskIOhardmax:totalCPUhardmax: totalCPUunit:totalDiskIOhardmax:totalDiskIOunit: totalConnecttimehardmax:totalConnecttimeunit: totalProcesseshardmax:totalThreadshardmax: totalLoginshardmax: classRealMem:classRealMemunit:classVirtMem: classVirtMemunit:classLargePages:classLargePagesunit: procVirtMem:procVirtMemunit:localshm:vmenforce:delshm myclass::0:no::::-:0:100:100:-:1:100:100:-:0:100: 100:-:s:-:KB:-:s:-:KB:-:KB:-:KB:-:KB:-:KB:no:proc:no</pre>
-d <i>Config_Dir</i>	Use /etc/wlm/Config_Dir as alternate directory for the definition files. If an empty string is passed (for example, -d ""), lsclass lists the classes defined in the in-core WLM data structures. If this flag is not present, the current configuration files in the directory pointed to by /etc/wlm/current are used.
-D	Displays the default values for the class attributes and limits in colon-separated records. Any other flag or argument used in conjunction with -D is ignored. For example: <pre>lsclass -D #name:description:tier:inheritance:authuser: authgroup:adminuser:admingroup:rset:CPUshares:CPUmin: CPUsoftmax:CPUhardmax:memoryshares:memorymin: memorysoftmax:memoryhardmax:diskIOshares:diskIOmin: diskIOsoftmax:diskIOhardmax:totalCPUhardmax: totalCPUunit:totalDiskIOhardmax:totalDiskIOunit: totalConnecttimehardmax:totalConnecttimeunit: totalProcesseshardmax:totalThreadshardmax:totalLoginshardmax: classRealMem:classRealMemunit:classVirtMem: classVirtMemunit:classLargePages:classLargePagesunit: procVirtMem:procVirtMemunit:localshm:vmenforce:delshm ::0:no::::-:0:100:100:-:0:100:100:-:0:100:100:-:s:-: KB:-:s:-:KB:-:KB:-:KB:-:KB:no:proc:no</pre>
-f	Displays the output in stanzas, with each stanza identified by a class name. Each <i>Attribute=Value</i> pair is listed on a separate line: <pre>Class: attribute1=value attribute2=value attribute3=value</pre>
-r	Displays, recursively, the superclasses with all their subclasses. When specifying -r : <ul style="list-style-type: none">• If <i>Class</i> is not specified, lsclass shows all the superclasses with all their subclasses.• If the name of a superclass is specified, lsclass displays the superclass with all its subclasses.• If the name of a subclass is specified, -r is ineffective (displays only the subclass).
-S <i>SuperClass</i>	Restricts the scope of the command to the subclasses of the specified superclass. Only subclasses are shown with the -S 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 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 limits enforced on the classes.
shares	Contains the resource shares attributes for each class.

lscluster Command

Purpose

Lists the cluster configuration information.

Syntax

```
lscluster { -i | -g [-k] | -d | -c [-n clustername] } { -m [nodename] | -s | -i interfacename | -g interfacename | -d diskname }
```

Description

The **lscluster** command displays the attributes associated with the cluster and the cluster configuration.

Flags

Item	Description
-c	Lists the cluster configuration.
-d	Lists the cluster storage interfaces.
-g	Lists the currently active network gateway interfaces as reported by each node.
-i	Lists the network device driver (NDD) and pseudo NDD interfaces that are currently configured on each of the Cluster Aware AIX (CAA) nodes. CAA might not use all of the interfaces to exchange heartbeat packets. Note: The storage framework communication (s _{fw} com) interface is displayed as UP only if this interface is configured and available. Otherwise, it is not displayed.
-k	Lists the assigned network gateway interfaces for each node. The interfaces can be down or not even configured. You can specify this flag only with the -g flag.
-m	Lists the cluster node configuration information. This information includes a list of points of contact. <i>Points of contact</i> are cluster configuration interfaces that are used by the cluster to exchange heartbeat packets. If a point of contact has no CAA traffic for an extended period, it is removed from the list of points of contact.
-n	Allows the cluster names to be queried for all interfaces, storage, or cluster configurations (applicable only with -i , -d , or -c flags).
-s	Lists the cluster network statistics on the local node.

Examples

1. To list the cluster configuration for all nodes, enter:

```
lscluster -m
```

The sample of the output follows:

```
# lscluster -m
Calling node query for all nodes...
Node query number of nodes examined: 2

      Node name: nodeA.ibm.com
      Cluster shorthand id for node: 1
      uuid for node: 84088524-b124-11e3-8210-32c8e74b1e02
      State of node: UP NODE_LOCAL
      Smoothed rtt to node: 0
      Mean Deviation in network rtt to node: 0
      Number of clusters node is a member in: 1
      CLUSTER NAME      TYPE  SHID  UUID
      Sample local      84ee37f4-b124-11e3-8210-32c8e74b1e02

      Number of points_of_contact for node: 0
      Point-of-contact interface & contact state
      n/a

      -----

      Node name: nodeB.ibm.com
      Cluster shorthand id for node: 2
      uuid for node: 8492a5a6-b124-11e3-8210-32c8e74b1e02
      State of node: UP
      Smoothed rtt to node: 70
      Mean Deviation in network rtt to node: 82
      Number of clusters node is a member in: 1
      CLUSTER NAME      TYPE  SHID  UUID
      Sample local      84ee37f4-b124-11e3-8210-32c8e74b1e02

      Number of points_of_contact for node: 2
      Point-of-contact interface & contact state
      dpcom UP RESTRICTED
      en0 UP
```

2. To list the cluster configuration for the local node, enter:

```
lscluster -s
```

The sample of the output follows:

```
# lscluster -s
Cluster Network Statistics:

pkts seen: 33861217                passed: 32052241
IP pkts: 5778096                   UDP pkts: 1934943
gossip pkts sent: 1463320          gossip pkts rcv: 688759
cluster address pkts: 0            CP pkts: 1808962
bad transmits: 5                   bad posts: 4
Bad transmit (overflow - disk ): 0
Bad transmit (overflow - tcpsock): 0
Bad transmit (host unreachable): 0
Bad transmit (net unreachable): 0
Bad transmit (network down): 0
Bad transmit (no connection): 0
short pkts: 0                      multicast pkts: 1808880
cluster wide errors: 0              bad pkts: 0
dup pkts: 0                         dropped pkts: 14
pkt fragments: 1                   fragments queued: 0
fragments freed: 0
pkts pulled: 0                     no memory: 0
rxmit requests rcv: 10              requests found: 3
requests missed: 7                  ooo pkts: 0
requests reset sent: 7              reset rcv: 0
remote tcpsock send: 0              tcpsock rcv: 0
rxmit requests sent: 0
alive pkts sent: 0                  alive pkts rcv: 0
ahafs pkts sent: 2                  ahafs pkts rcv: 0
nodedown pkts sent: 0               nodedown pkts rcv: 1
```

socket pkts sent: 62	socket pkts recv: 54
cwide pkts sent: 275321	cwide pkts recv: 275318
socket pkts no space: 0	pkts recv notforhere: 0
Pseudo socket pkts sent: 0	Pseudo socket pkts recv: 0
Pseudo socket pkts dropped: 0	
arp pkts sent: 1	arp pkts recv: 2
stale pkts recv: 0	other cluster pkts: 4
storage pkts sent: 1	storage pkts recv: 1
disk pkts sent: 174	disk pkts recv: 0
unicast pkts sent: 275364	unicast pkts recv: 82
out-of-range pkts recv: 0	
IPv6 pkts sent: 0	IPv6 pkts recv: 122
IPv6 frags sent: 0	IPv6 frags recv: 0
Unhandled large pkts: 0	
mxrmit overflow : 0	urxmit overflow: 0

3. To list the interface information for the local node, enter:

```
lscluster -i
```

The sample of output follows:

```
# lscluster -i
Network/Storage Interface Query

Cluster Name: Sample
Cluster uuid: 84ee37f4-b124-11e3-8210-32c8e74b1e02
Number of nodes reporting = 2
Number of nodes expected = 2

Node nodeA.ibm.com
Node uuid = 84088524-b124-11e3-8210-32c8e74b1e02
Number of interfaces discovered = 2
  Interface number 1 en0
    ifnet type = 6 ndd type = 7
    Mac address length = 6
    Mac address = 32:C8:E7:4B:1E:02
    Smoothed rrt across interface = 0
    Mean Deviation in network rrt across interface = 0
    Probe interval for interface = 100 ms
    ifnet flags for interface = 0x1E080863
    ndd flags for interface = 0x0021081B
    Interface state UP
    Number of regular addresses configured on interface = 1
    IPv4 ADDRESS: 9.3.199.216 broadcast 9.3.199.255 netmask
255.255.254.0
    Number of cluster multicast addresses configured on interface = 1
    IPv4 MULTICAST ADDRESS: 228.3.199.216 broadcast 0.0.0.0
netmask 0.0.0.0
  Interface number 2 dpcom
    ifnet type = 0 ndd type = 305
    Mac address length = 0
    Mac address = 00:00:00:00:00:00
    Smoothed rrt across interface = 750
    Mean Deviation in network rrt across interface = 1500
    Probe interval for interface = 22500 ms
    ifnet flags for interface = 0x00000000
    ndd flags for interface = 0x00000009
    Interface state UP RESTRICTED AIX_CONTROLLED
  Pseudo Interface
  Interface State DOWN

Node nodeB.ibm.com
Node uuid = 8492a5a6-b124-11e3-8210-32c8e74b1e02
Number of interfaces discovered = 2
  Interface number 1 en0
    ifnet type = 6 ndd type = 7
    Mac address length = 6
    Mac address = 32:C8:EF:AD:7C:02
    Smoothed rrt across interface = 0
    Mean Deviation in network rrt across interface = 0
    Probe interval for interface = 990 ms
    ifnet flags for interface = 0x1E084863
    ndd flags for interface = 0x0021081B
    Interface state UP
    Number of regular addresses configured on interface = 1
    IPv4 ADDRESS: 9.3.199.128 broadcast 9.3.199.255 netmask
255.255.254.0
    Number of cluster multicast addresses configured on interface = 1
    IPv4 MULTICAST ADDRESS: 228.3.199.216 broadcast 0.0.0.0
```

```

netmask 0.0.0.0
Interface number 2 dpcom
    ifnet type = 0 ndd type = 305
    Mac address length = 0
    Mac address = 00:00:00:00:00:00
    Smoothed rrt across interface = 750
    Mean Deviation in network rrt across interface = 1500
    Probe interval for interface = 22500 ms
    ifnet flags for interface = 0x00000000
    ndd flags for interface = 0x00000009
    Interface state UP RESTRICTED AIX_CONTROLLED
Pseudo Interface
    Interface State DOWN

```

4. To list the storage interface information for the cluster, enter:

```
lscluster -d
```

The sample of output follows:

```

# lscluster -d
Storage Interface Query

Cluster Name: Sample
Cluster uuid: 84ee37f4-b124-11e3-8210-32c8e74b1e02
Number of nodes reporting = 2
Number of nodes expected = 2
Node nodeA.ibm.com
Node uuid = 84088524-b124-11e3-8210-32c8e74b1e02
Number of disk discovered = 1
    hdisk4
        state : UP
        uDid :
        uUid : 76c94719-7335-ded6-10e2-77d61ff7998c
        type : REPDISK
Node nodeB.ibm.com
Node uuid = 8492a5a6-b124-11e3-8210-32c8e74b1e02
Number of disk discovered = 1
    hdisk0
        state : UP
        uDid : 382300c4f4f700004c0000000140799c6e39.3105VDASD03AIXvscsi
        uUid : 76c94719-7335-ded6-10e2-77d61ff7998c
        type : REPDISK

```

5. To list the cluster configuration, enter:

```
lscluster -c
```

The sample of the output follows:

```

# lscluster -c
Cluster Name: Sample
Cluste UUID: 8e1d89da-b39d-11e3-91e7-d24dc2d9d309
Number of nodes in cluster = 2
    Cluster ID for node nodeA.ibm.com: 1
    Primary IP address for node r5r3m25.aus.stglabs.ibm.com: 9.3.207.132
    Cluster ID for node nodeB.ibm.com: 2
    Primary IP address for node r5r3m26.aus.stglabs.ibm.com: 9.3.207.218
Number of disks in cluster = 1
    Disk = hdisk6 UUID = 57208624-fda4-d404-a7c0-8e425e2941a4 cluster_major = 0
    cluster_minor = 1
Multicast for site LOCAL: IPv4 228.3.207.132 IPv6 ff05::e403:cf84
Communication Mode: multicast
Local node maximum capabilities: HNAME_CHG, UNICAST, IPV6, SITE
Effective cluster-wide capabilities: HNAME_CHG, UNICAST, IPV6, SI

```

6. To list the currently assigned network gateway interfaces for each node, enter the following command:

```
lscluster -g -k
```

The sample of the output follows:

```

# lscluster -g -k
Configured Network Gateway Interfaces

Cluster Name: Sample

```



```

Cluster UUID: 6b535f42-b3d3-11e9-8004-00145e742bf8
Number of nodes found = 2

Node node1.ibm.com
Node SHID = 1
Node UUID = 6b2d5f36-b3d3-11e9-8004-00145e742bf8
    Number of network gateways Configured = 1
    IPv4 ADDRESS: 9.3.xxx.xxx

Node node2.ibm.com
Node SHID = 2
Node UUID = 3f5c5970-b789-11e9-800a-00145e742bf8
    Number of network gateways Configured = 2
    IPv4 ADDRESS: 9.3.xxx.xxx
    IPv4 ADDRESS: 12.12.12.12

```

In the output, the IP address, 12.12.12.12, is not active but is displayed because of the **-k** flag.

lscmg Command

Purpose

Displays information about the communication groups of a peer domain.

Syntax

```
lscmg [-l | -t | -d | -D delimiter] [-x] [-i] [-h] [-TV] [communication_group]
```

Description

The `lscmg` command displays information about the communication groups that are defined to the online peer domain on which the command runs. If you specify the name of a communication group, the `lscmg` command displays information about that communication group only.

Some of the communication group information that is displayed follows:

Field	Description
Name	The name of the communication group
Sensitivity	The number of missed heartbeats that constitute a failure
Period	The number of seconds between heartbeats
Priority	The relative priority of the communication group
Broadcast	Indicates whether broadcast should be used if it is supported by the underlying media
SourceRouting	Indicates whether source routing should be used if it is supported by the underlying media
NIMPath	The path to the Network Interface Module (NIM) that supports the adapter types in the communication group
NIMParameters	The NIM start parameters

Interface resources

Use the `-i` flag to display information about the interface resources that refer to *communication_group*.

For IP communication groups (MediaType = 1), `lscmg -i` displays the following information:

Field	Description
Name	The name of the interface resource that refers to <i>communication_group</i> .

Field	Description
NodeName	The host name of the interface resource that refers to <i>communication_group</i> .
IPAddress	The IP address of the interface resource that refers to <i>communication_group</i> .
SubnetMask	The subnet mask of the interface resource that refers to <i>communication_group</i> .
Subnet	The subnet of the interface resource that refers to <i>communication_group</i> .

For disk heartbeating (MediaType = 2) and other non-IP types of communication groups (MediaType = 0), `lscomg -i` displays the following information:

Field	Description
Name	The name of the interface resource that refers to <i>communication_group</i> .
NodeName	The host name of the interface resource that refers to <i>communication_group</i> .
DeviceInfo	Information about the device.
MediaType	The type of interfaces that make up this communication group.

Flags

- l**
Displays the information on separate lines (long format).
- t**
Displays the information in separate columns (table format). This is the default format.
- d**
Displays the information using delimiters. The default delimiter is a colon (:). Use the **-D** flag if you want to change the default delimiter.
- D delimiter**
Displays the information using the specified delimiter. Use this flag to specify a delimiter other than the default colon (:) — when the information you want to display contains colons, for example. You can use this flag to specify a delimiter of one or more characters.
- x**
Excludes the header (suppresses header printing).
- i**
Displays information about the interface resource that refers to *communication_group*.
- 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 communication group about which you want to display information. You can specify a communication group name or a substring of a communication group name for this parameter. If you specify a substring, the command displays information about any defined communication group with a name that contains the substring.

Security

The user of the `lscomg` command needs read permission for the `IBM.CommunicationGroup` resource class. Read permission for the `IBM.NetworkInterface` resource class is required to display the

network interface information. By default, root on any node in the peer domain has read and write access to these resource classes 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 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 defined and online to the peer domain on which the communication group exists.

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 these examples, nodeA is defined and online to peer domain App1Domain.

1. To display general information about the communication groups for App1Domain, run this command on nodeA:

```
lscomg
```

The following output is displayed:

Name	Sensitivity	Period	Priority	Broadcast	SourceRouting	NIMPath	
NIMParameters							
ComG1	2	2	1	no	yes	/opt/rsct/bin/hats_nim	-l 5

2. To display information about the interface resources that refer to the communication group ComGrp1 for the peer domain App1Domain, run this command on nodeA:

```
lscomg -i ComGrp1
```

The following output is displayed:

Name	NodeName	IPAddr	SubnetMask	Subnet
eth0	n24	9.234.32.45	255.255.255.2	9.235.345.34
eth0	n25	9.234.32.46	255.255.255.2	9.235.345.34

Location

/opt/rsct/bin/lscomg

lscondition Command

Purpose

Lists information about one or more conditions.

Syntax

```
lscondition [-a] [-m | -n | -e] [-C | -l | -t | -d | -D delimiter] [-A] [-q] [-U] [-x] [-h] [-TV]
[condition1 [, condition2, ...]:node_name]
```

Description

The lscondition command lists the following information about defined conditions:

Field	Description
Name	The name of the condition.
Node	The location of the condition (for management domain scope or peer domain scope).
MonitorStatus	The status of the condition.
ResourceClass	The resource class that is monitored by this condition.
EventExpression	The expression that is used in monitoring this condition.
EventDescription	A description of the EventExpression field.
RearmExpression	The expression used in determining when monitoring should restart for this condition after an event has occurred.
RearmDescription	A description of the RearmExpression field.
SelectionString	The selection string that is applied to the attributes of ResourceClass to determine which resources are included in the monitoring of this condition.

Field	Description
Severity	The severity of the condition: critical, warning, or informational.
NodeNames	The host names of the nodes where the condition is registered.
MgtScope	The RMC scope in which the condition is monitored.
Toggle	Specifies whether the condition toggles between the event and the rearm event.
Locked	Specifies whether the resource is locked or unlocked.
EventBatchingInterval	Specifies the time in seconds that is used to determine when the accumulated events are batched together and sent to the response. A value of 0 indicates that no batching is used.
EventBatchingMaxEvents	Specifies the maximum number of events that can be in a single batch of events. A value of 0 indicates that there is no maximum if the value of EventBatchingInterval is not 0.
BatchedEventRetentionPeriod	Specifies the time in hours that the batched event file is kept after all associated response scripts are run.
BatchedEventMaxTotalSize	Specifies that the total saved batched event file size can't exceed a certain size in megabytes (MB) per condition. RecordAuditLog Specifies the level of detail for ERRM log entries to the audit log (ALL, Error Only, or None).

For a list of all conditions, enter the `lscondition` command without any condition names specified. A list of all the condition names is returned with the monitoring status for each condition. The default format in this case is tabular. Specifying a node name following the condition names limits the display to the conditions defined on that node. You can list all of the conditions on a node by specifying a colon (:) followed by the node name. The node name is a node within the management scope, which is determined by the `CT_MANAGEMENT_SCOPE` environment variable. The management scope determines the list of nodes from which the conditions are listed. For local scope, only conditions on the local node are listed. Otherwise, the conditions from all nodes within the domain are listed.

For all of the information about all condition names, specify the `-A` flag with the `lscondition` command. The `-A` flag causes all information about a condition to be listed when no condition names are specified. When all the information about all conditions is listed, the default format is long. If a monitoring-status flag (`-e`, `-m`, or `-n`) is specified, the conditions with that status are listed.

When more than one condition is specified, the condition information is listed in the order in which the condition names are entered.

By default, when a condition name is specified with the `lscondition` command, all of the condition's attributes are displayed.

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.

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, `lscondition -a` with `CT_MANAGEMENT_SCOPE` not set will list the management domain. In this case, to list the peer domain, set `CT_MANAGEMENT_SCOPE` to 2.

- A**
Displays all of the attributes of the condition.
- C**
Displays a `mkcondition` command template based on the condition. By modifying this template, you can create new conditions. If more than one condition is specified, the template for each `mkcondition` command appears on a separate line. This flag is ignored when no conditions are specified. This flag overrides the `-l` flag.
- d**
Produces delimiter-formatted output. The default delimiter is a colon (:). Use the `-D` flag if you want to change the default delimiter.
- D *delimiter***
Produces delimiter-formatted output that uses the specified delimiter. Use this flag to specify something 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.
- e**
Lists only those conditions that are monitored in error.
- l**
Produces long-formatted output. Displays the condition information on separate lines.
- m**
Lists only those conditions that are being monitored without error.
- n**
Lists only those conditions that are not being monitored.
- q**
Does not return an error when the condition does not exist.
- t**
Displays the condition information in separate columns (table format).
- U**
Indicates whether the resource is locked.
- 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

***condition1* [, *condition2* , ...]**

Specifies the name of an existing condition that is defined on the host name *node_name*. You can specify more than one condition name. This parameter can be a condition name or a substring of a condition name. When it is a substring, any defined condition name that contains the substring will be listed.

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 read permission for the `IBM.Condition` resource class to run `lscondition`. 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 list all conditions and their monitoring status, run this command:

```
lscondition
```

The output will look like this:

Name	Node	MonitorStatus
"FileSystem space used"	"nodeA"	"Monitored"
"tmp space used"	"nodeA"	"Not monitored"
"var space used"	"nodeA"	"Error"

2. To list general information about the condition "FileSystem space used" in long form, run this command:

```
lscondition "FileSystem space used"
```

The output will look like this:

```
Name           = "FileSystem space used"
Node           = "nodeA"
MonitorStatus  = "Monitored"
ResourceClass  = "IBM.FileSystem"
EventExpression = "PercentTotUsed > 99"
EventDescription = "Generate event when space used is
greater than 99 percent full"
RearmExpression = "PercentTotUsed < 85"
RearmDescription = "Start monitoring again after it is
less than 85 percent"
SelectionString = ""
Severity       = "w"
NodeNames     = "{}"
MgtScope      = "1"
Toggle        = "Yes"
Locked        = "No"
```

3. To list the command that would create the condition "FileSystem space used", run this command:

```
lscondition -C "FileSystem space used"
```

The output will look like this:

```
mkcondition -r IBM.FileSystem -a PercentTotUsed \
-e "PercentTotUsed > 99" -E "PercentTotUsed < 85" \
-d "Generate event when space used is greater than 99 percent full" \
-D "Start monitoring after it is less than 85 percent" \
-S w "FileSystem space used"
```

4. To list all conditions that have the string space in their names, run this command:

```
lscondition space
```

The output will look like this:

Name	= "FileSystem space used"
MonitorStatus	= "Monitored"
Name	= "tmp space used"


```
MonitorStatus = "Not Monitored"
Name          = "var space used"
MonitorStatus = "Monitored"
```

5. To list the conditions that are in error, run this command:

```
lscondition -e
```

The output will look like this:

```
Name          MonitorStatus
"var space used" "Error"
```

This example applies to clustered systems:

1. To list all conditions and their monitoring status, run this command:

```
lscondition -a
```

The output will look like this:

```
Name          Node          MonitorStatus
"FileSystem space used" "nodeA"      "Monitored"
"tmp space used"      "nodeB"      "Not monitored"
"var space used"      "nodeC"      "Error"
```

Location

`/opt/rsct/bin/lscondition`

lscondresp Command

Purpose

Lists information about a condition and any of its condition/response associations.

Syntax

To list the link between a condition and one or more responses:

```
lscondresp [-a | -n] [-l | -t | -d | -D delimiter] [-q] [-U] [-x] [-z] [-h] [-TV] [condition[:node_name]] [response1 [response2...]]
```

To list all of the links to one or more responses:

```
lscondresp [-a | -n] [-l | -t | -d | -D delimiter] [-q] [-x] [-z] -r [-U] [-h] [-TV] response1[:node_name] [response2...]
```

Description

The `lscondresp` command lists information about a condition and its linked responses. A link between a condition and a response is called a *condition/response association*. The information shows which responses are linked with a condition and whether monitoring is active for a condition and its linked response. The following information is listed:

Field	Description
Condition	The name of the condition linked with a response.
Response	The name of the response linked with the condition.

Field	Description
State	The state of the response for the condition. The state indicates whether a specified response is active or not.
Node	The location of the condition and the response.
Locked	Indicates whether the resource is locked or unlocked.

To list a particular condition and response, specify both the condition and the response. To list all responses to a condition, specify the condition only. To list all conditions to which a response is linked, specify the response and the `-r` flag. To list all conditions and their linked responses, do not specify any condition or response parameters.

Specifying a node name limits the display to the condition/response associations that are defined on that node. List all of the condition/response associations on a node by specifying a colon (:) followed by the node name. The node name is a node within the management scope determined by the `CT_MANAGEMENT_SCOPE` environment variable. The management scope determines the list of nodes from which the condition/response associations are listed. For local scope, only condition/response associations on the local node are listed. For management domain scope and peer domain scope, the condition/response associations from all nodes within the domain are listed.

When neither the `-a` flag nor the `-n` flag is specified, all selected conditions for the responses are listed. Tabular format is the default.

Flags

-a

Lists only those responses that are active for the condition.

-n

Lists only those responses that are not active for the condition.

-l

Displays the condition information and response information on separate lines (long format).

-t

Displays the condition information and response information in separate columns (table format).

-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 *delimiter*. Use this flag to specify something other than the default colon (:). For example, when the data to be displayed contains colons, use this flag to specify another delimiter of one or more characters.

-q

Does not return an error if either the *condition* or the *response* does not exist.

-U

Indicates whether the resource is locked.

-x

Suppresses header printing.

-z

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, `lscondresp -z` with `CT_MANAGEMENT_SCOPE` not set will list the management domain. In this case, to list the peer domain, set `CT_MANAGEMENT_SCOPE` to 2.

- r**
Lists information about all of the condition/response associations for the specified responses. Use this flag to indicate that all command parameters specified are responses, not conditions.
- 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

The *condition* can be a condition name or a substring of a condition name. When it is a substring, any defined condition name that contains the substring and is linked to the response will be listed.

response1 [response2...]

This parameter can be a response name or a substring of a response name. You can specify more than one response name. When it is a substring, any defined response name that contains the substring and is linked to the condition will be listed.

node_name

Specifies the node where the condition or 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 read permission for the IBM.Association resource class to run `lscondresp`. 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

To see which resources are locked, run this command:

```
lscondresp -U
```

The output will look like this:

Condition	Response	Node	State	Locked
"/tmp space used"	"E-mail root off-shift"	"nodeA"	"Not active"	"Yes"
"Page space in rate"	"E-mail root anytime"	"nodeA"	"Not active"	"No"

These examples apply to standalone systems:

1. To list all conditions with their linked responses, run this command:

```
lscondresp
```

The output will look like this:

Condition	Response	Node	State
"FileSystem space used"	"Broadcast event on-shift"	"nodeA"	"Active"
"FileSystem space used"	"E-mail root anytime"	"nodeA"	"Not Active"
"Page in Rate"	"Log event anytime"	"nodeA"	"Active"

2. To list information about the condition "FileSystem space used", run this command:

```
lscondresp "FileSystem space used"
```

The output will look like this:

Condition	Response	Node	State
"FileSystem space used"	"Broadcast event on-shift"	"nodeA"	"Active"
"FileSystem space used"	"E-mail root anytime"	"nodeA"	"Not Active"

3. To list information about the condition "FileSystem space used" for responses that are active, run this command:

```
lscondresp -a "FileSystem space used"
```

The output will look like this:

Condition	Response	Node	State
"FileSystem space used"	"Broadcast event on-shift"	"nodeA"	"Active"

4. To list information about the condition "FileSystem space used" with the linked response "Broadcast event on-shift", run this command:

```
lscondresp "FileSystem space used" "Broadcast event on-shift"
```

The output will look like this:

Condition	Response	Node	State
"FileSystem space used"	"Broadcast event on-shift"	"nodeA"	"Active"

5. To list all conditions that have the string space in their names with their linked responses, run this command:

```
lscondresp space
```

The output will look like this:

Condition	Response	Node	State
"FileSystem space used"	"Broadcast event on-shift"	"nodeA"	"Active"
"FileSystem space used"	"E-mail root anytime"	"nodeA"	"Not Active"

These examples apply to management domains:

1. In this example, the condition "FileSystem space used" is defined on the management server. To list information about "FileSystem space used", run this command on the management server:

```
lscondresp "FileSystem space used"
```

The output will look like this:

Condition	Response	Node	State
"FileSystem space used"	"Broadcast event on-shift"	"nodeB"	"Active"
"FileSystem space used"	"E-mail root anytime"	"nodeB"	"Not Active"

2. In this example, the condition "FileSystem space used" is defined on the managed node nodeC. To list information about "FileSystem space used", run this command on the management server:

```
lscondresp "FileSystem space used":nodeC
```

The output will look like this:

Condition	Response	Node	State
"FileSystem space used"	"Broadcast event on-shift"	"nodeC"	"Active"

```
"FileSystem space used" "E-mail root anytime" "nodeC" "Not Active"
```

This example applies to a peer domain:

1. In this example, the condition "FileSystem space used" is defined in the domain. To list information about "FileSystem space used", run this command on one of the nodes in the domain:

```
lscondresp "FileSystem space used"
```

The output will look like this:

Condition	Response	Node	State
"FileSystem space used"	"Broadcast event on-shift"	"nodeD"	"Active"
"FileSystem space used"	"E-mail root anytime"	"nodeD"	"Not Active"
"FileSystem space used"	"Broadcast event on-shift"	"nodeE"	"Active"
"FileSystem space used"	"E-mail root anytime"	"nodeE"	"Not Active"

Location

`/opt/rsct/bin/lscondresp`

lsconn Command

Purpose

Displays the connections a given device, or kind of device, can accept.

Syntax

```
lsconn { -p ParentName | [ -c ParentClass ] [ -s ParentSubclass ] [ -t ParentType ] } { -l ChildName | -k ChildConnectionKey } [ -f File ] [ -F Format ] [ -h ] [ -H ]
```

Description

The **lsconn** command, when used with the **-p** *ParentName* flag, displays the connection locations on the parent device to which the device specified by the **-l** *ChildName* flag can be connected, or to which devices of the connection type specified by the **-k** *ChildConnectionKey* flag can be connected. If the **-k** and **-l** flags are not used, the **lsconn** command displays information about where a child device can be connected on the specified parent.

If the **-p** *ParentName* flag is not used, you must use a combination of one or all of the **-c** *ParentClass*, **-s** *ParentSubclass*, and **-t** *ParentType* flags to uniquely identify the predefined parent device.

You can display the default output, which is the connection location (or connection location and connection key if no child is specified), from the Predefined Connection object class. If you do not display the default, you can display the output in a user-specified format where the *Format* parameter is a quoted list of column names separated by nonalphanumeric characters or white space using the **-F** *Format* flag. You can insert headers above the columns using the **-H** flag.

Use the flags either on the command line or in the specified **-f** *File* flag.

Flags

Item	Description
-c <i>ParentClass</i>	Specifies the class name of a possible parent device in the Predefined Devices object class. This flag cannot be used with the -p flag.
-f <i>File</i>	Reads the necessary flags from the <i>File</i> parameter.

Item	Description
-F <i>Format</i>	Formats the output in a user-specified format, where the <i>Format</i> parameter is a quoted list of column names from the Predefined Connection object class separated, and possibly terminated, by non-alphanumeric characters or white space. If white space is used as the separator, the lsconn command displays the output in aligned columns.
-H	Displays headers above the column output.
-h	Displays the command usage message.
-k <i>ChildConnectionKey</i>	Specifies the connection key that identifies the subclass of the child device. This flag cannot be used with the -l flag.
-l <i>ChildName</i>	Specifies the logical name of a possible child device. This flag cannot be used with the -k flag.
-p <i>ParentName</i>	Specifies the parent device's logical name from the Customized Devices object class. This flag cannot be used with the -c , -s , or -t flag.
-s <i>ParentSubclass</i>	Specifies the subclass of a possible parent device in the Predefined Devices object class. This flag cannot be used with the -p flag.
-t <i>ParentType</i>	Specifies the device type of a possible parent device from the Predefined Devices object class. This flag cannot be used with the -p 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 that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To list all of the possible connection locations on the sa2 IBM 8-Port EIA-232/RS-422A (PCI) Adapter that will accept an RS-232 device connection, type the following:

```
lsconn -p sa2 -k rs232
```

The system displays a possible connections similar to the following:

```
0
1
2
3
4
5
6
7
```

2. To list all of the possible connection locations and connection types on the sa2 IBM 8-Port EIA-232/RS-422A (PCI) Adapter, type the following:

```
lsconn -p sa2
```

The system displays a message similar to the following:

```
0 rs232
1 rs232
2 rs232
3 rs232
```

```
4 rs232
5 rs232
6 rs232
7 rs232
0 rs422
1 rs422
2 rs422
3 rs422
4 rs422
5 rs422
6 rs422
7 rs422
```

Files

Item	Description
<code>/usr/sbin/lscnnc</code>	Specifies the command file.

lscons Command

Purpose

Writes the name of the current console device to standard output.

Syntax

```
lscons [ -s ] [ -a | -O ]
```

```
lscons -b [ -s ] [ -a | -O ]
```

```
lscons -d [ -s ]
```

Description

The **lscons** command writes the name of the current console device to standard output. This command is also used to write the name of the device that is to be the console on the next start of the system to standard output. You can change the current console device using the **swcons** command. You can change the device to be the system console on the next start of the system using the **chcons** command.

Flags

Item	Description
------	-------------

-a	Displays a list of <i>attribute name = attribute value</i> pairs for the console device and console logging and tagging attributes. When used with the -b flag, the values are retrieved from the ODM. Without the -b flag, the values are retrieved from the console device driver.
-----------	--

Note: This flag is not valid with the **-O** flag or the **-d** flag.

-b	Displays the full path name of the system console selected for the next startup of the system.
-----------	--

-d	Displays the full path name of the system console selected on the current startup of the system.
-----------	--

Note: This flag is not valid with the **-O** flag or the **-a** flag.

-O	Similar to the -a flag but outputs the attribute names and values in a format suitable for use by SMIT. This flag is NOT valid with the -d flag.
-----------	--

Note: This flag is not valid with the **-d** flag or the **-a** flag.

-s	Suppresses reporting of the path name.
-----------	--

Exit Status

This command returns the following exit values:

Item	Description
0	The device you are using is the current system console.
1	The device you are using is not the current system console.
2	The device you are using is the console device selected at system start but is not currently the device supporting console message output.
3	Flags specified are not valid.
4	System error occurred.

Examples

1. To display the full path name of the current system console, type:

```
lscons
```

2. To display the full path name of the system console effective on the next startup of the system, type:

```
lscons -b
```

3. To display the full path name of the system console selected on the current startup of the system, type:

```
lscons -d
```

4. To test whether or not the current system console is directed to your display, type:

```
if lscons -s
then
echo "System messages are directed to my display" >/dev/tty
fi
```

Files

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

lscore Command

Purpose

Views the current core settings.

Syntax

```
lscore [ -R registry ] [ username | -d ]
```

Description

The `lscore` command will be the user interface to view the current core settings. It will have the following usage:

```
lscore [-R registry] [username|-d]
```

As with `chcore`, the `-d` flag will show the default values. Viewing settings for another user is a privileged operation; however, any user may view the default values.

Flags

Item	Description
<code>-d</code>	Changes the default setting for the system.
<code>-R registry</code>	Specifies the loadable I&A module.

Security

May only be run by root or another user with system authority.

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 settings for root, type:

```
lscore root
```

The output will look like:

```
compression: on
path specification: default
corefile location: default
naming specification: off
```

2. To list the default settings for the system, type:

```
lscore -d
```

The output will look like:

```
compression: off
path specification: on
corefile location: /corefiles
naming specification: off
```

lscosi Command

Purpose

Lists information related to a Common Operating System Image (COSI).

Syntax

```
lscosi [ [-l{1|2|3}] . . . ] [-v] [COSI]
```

Description

The `lscosi` command lists the status and detailed information related to a Common Operating System Image (COSI). The level of information to be listed depends on the numeric value specified by the `-l` flag, with a level ranging from 1 - 3 (3 being the most detailed). If a level is not specified, a default of level 1 information is displayed. If no argument is specified, the `lscosi` command lists any common images that

exist in the environment. The `bos.sysmgmt.nim.master` fileset must be present on the system in order for the `lscosi` command to be successful. This command can also be executed on a thin server.

Flags

Item	Description
<code>-l{1 2 3}</code>	Specifies the level of information to display. 1 This level displays very limited information related to a COSI. The information listed shows only a brief summary of the COSI and the thin servers that might be using it. 2 This level displays more than just basic information related to a COSI. The level includes information pertaining to the software content of the COSI. 3 This level displays more in-depth information related to a COSI. The level includes information pertaining to the installation log of the COSI.
<code>-v</code>	Enables verbose debug output when the <code>lscosi</code> 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 `lscosi` command.

Examples

1. To check if any common images exist in an environment, enter:

```
lscosi
```

When this command is entered without an argument, it merely lists common images in the environment. The output might be similar to the following list:

```
52H_0442A_cosi  
52I_0444B2_GOLD_cosi  
52L_0534A_cosi  
53E_0545A_cosi  
53D_GOLD_cosi  
53A_GOLD_cosi  
52M_0544A_cosi
```

2. To list brief status information for a common image named `cosi1`, enter:

```
lscosi cosi1
```

Information similar to the following is displayed:

```
53H_0538A_spot:
  class      = resources
  type       = spot
  plat_defined = chrp
  Rstate     = ready for use
  prev_state = verification is being performed
  location   = /export/nimvg/spot/53H_0538A_spot/usr
  version    = 5
  release    = 2
  mod        = 0
  oslevel_r  = 5300-05
  alloc_count = 2
  server     = master
  if_supported = chrp.mp ent
  Rstate_result = success

Thin Server:
Client1
Client2
```

3. To list software content for a common image named `cosi1`, enter:

```
lscosi -l2 cosi1
```

Software content similar to the following is displayed from the common image:

Fileset (Uninstaller)	Level	State	Type	Description
-----	-----	-----	-----	-----
bos.64bit	5.2.0.75	C	F	Base Operating System 64 bit Runtime
bos.diag.com	5.2.0.75	C	F	Common Hardware Diagnostics
bos.diag.rte	5.2.0.75	C	F	Hardware Diagnostics
.				
.				
.				

4. To list both software content and status information for a common image named `cosi1`, enter:

```
lscosi -l1 -l2 cosi1
```

Location

/usr/sbin/lscosi

Files

Item

/etc/niminfo

Description

Contains variables used by NIM.

lsdev Command

Purpose

Displays devices in the system and their characteristics.

Syntax

```
lsdev [ -C ] [ -c Class ] [ -s Subclass ] [ -t Type ] [ -f File ] [ -F Format / -r ColumnName ] [ -h ] [ -H ] [ -l { Name | - } ] [ -p Parent ] [ -S State ] [ -x ]
```

```
lsdev -P [ -c Class ] [ -s Subclass ] [ -t Type ] [ -f File ] [ -F Format | -r ColumnName ] [ -h ] [ -H ] [ -x ]
```

Description

The **lsdev** command displays information about devices in the Device Configuration database. You can display information about all devices in the Customized Devices object class using the **-C** flag. Any combination of the **-c Class**, **-s Subclass**, **-t Type**, **-l Name**, **-p Parent**, and **-S State** flags selects a subset of the customized devices. You can display information about all devices in the Predefined Devices object class using the **-P** flag. Any combination of the **-c Class**, **-s Subclass**, and **-t Type** flags selects a subset of the predefined devices.

You can display the default output one of the following ways:

- From the Customized Devices object class by using the **-C** flag
- From the Predefined Devices object class by using the **-P** flag

To override these two default outputs, use the **-F Format** flag to display the output in a format that you specify by using the *Format* parameter. The *Format* parameter is a quoted list of column names that are separated and possibly ended by non-alphanumeric characters or white space.

The **lsdev** command shows information only about devices that are based upon information in the Customized Devices (**Cudv**) object class or the Predefined Devices (**PdDv**) object class. Other object classes (such as the Customized Path (**CuPath**) object class) are not examined. This situation means that there might be conditions where a device might not be displayed. For example, if the **-p Parent** flag is used, but the parent that is identified in the Customized Devices object for a device does not match the *Parent* that is specified through the **-p** flag, the device is not displayed. However, the device might have a path to the specified *Parent* that is defined in the Customized Paths object class. Use the **lspath** command to show all MPIO-capable child devices of the specified parent.

You can use the System Management Interface Tool (SMIT) **smit lsdev** fast path to change device characteristics.

Flags

Item	Description
-C	Lists information about a device that is in the Customized Devices object class. The default information that is displayed is <i>name</i> , <i>status</i> , <i>location</i> , and <i>description</i> . The -C flag is not required, but is maintained for compatibility reasons. The -C flag cannot be specified with the -P flag. If neither is specified, the lsdev command behaves as if the -C flag was specified.
-c Class	Specifies a device class name. This flag can be used to restrict output to devices in a specified class.
-f File	Reads the necessary flags from the <i>File</i> parameter.
-F Format	Displays the output in a user-specified format, where the <i>Format</i> parameter is a quoted list of column names from the Predefined or Customized Devices object class, separated and possibly ended by nonalphanumeric characters or white space. If white space is used as the separator, the lsdev command displays the output in aligned columns. If you specify the -F Format flag with the -C flag, you can specify column names from both the Customized and Predefined Devices object classes. If you specify the -F Format flag with the -P flag, you can specify only column names from the Predefined Devices object class. In addition to the column names, the special purpose name <i>description</i> can be used to obtain a display of device descriptions. This flag cannot be used with the -r ColumnName flag. Also, the <i>physloc</i> special purpose name can be used to display a physical location code of the device.
-H	Displays headers above the column output.
-h	Displays the command usage message.

Item	Description
-l <i>Name</i>	Specifies the device logical name from the Customized Devices object class of the device for which information is listed. The <i>Name</i> argument to the -l flag can contain the same wildcard characters that can be used with the odmget command. If the <i>Name</i> argument is a dash, names are read from STDIN. Names on STDIN must be separated by a comma, a tab, a space, or a "newline" character. Names cannot contain wildcard characters. This flag cannot be used with the -P flag.
-p <i>Parent</i>	Specifies the device logical name from the Customized Devices object class for the parent of devices to be displayed. The -p <i>Parent</i> flag can be used to show the child devices of the specified <i>Parent</i> . The <i>Parent</i> argument to the -p flag might contain the same wildcard characters that can be used with the odmget command. This flag cannot be used with the -P flag.
-P	Lists information about a device that is in the Predefined Devices object class. The default information that is displayed is <i>class</i> , <i>type</i> , <i>subclass</i> , and <i>description</i> . This flag cannot be used with the -C , -l , or -S flags.
-r <i>ColumnName</i>	Displays the set of values in a column. For example, the <i>ColumnName</i> parameter takes the value of the <i>Class</i> parameter to list all of the classes. If you specify the -r <i>ColumnName</i> flag with the -C flag, you can specify column names from both the Customized and Predefined Devices object classes. If you specify the -r <i>ColumnName</i> flag with the -P flag, you can specify only column names from the Predefined Devices object class. This flag cannot be used with the -F <i>Format</i> flag.
-S <i>State</i>	Lists all devices in a specified state as named by the <i>State</i> parameter. The <i>State</i> parameter can have one of the following values: <ul style="list-style-type: none"> • <i>d</i>, <i>D</i>, <i>0</i> or <i>defined</i> for the Defined state • <i>a</i>, <i>A</i>, <i>1</i>, or <i>available</i> for the Available state • <i>s</i>, <i>S</i>, <i>2</i>, or <i>stopped</i> for the Stopped state This flag can be used to restrict output to devices in a specified state. This flag cannot be used with the -P flag.
-s <i>Subclass</i>	Specifies a device subclass name. This flag can be used to restrict output to devices in a specified subclass.
-t <i>Type</i>	Specifies a device type name. This flag can be used to restrict output to devices of a specified type.
-x	Displays the exported status for devices that are exported to a Workload Partition (WPAR).

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 list all of the devices in the Predefined Devices object class with column headers, type the following command:

```
lsdev -P -H
```

The system displays a message similar to the following output:

```

class      type          subclass  description
logical_volume vgtype      vgsubclass Volume group
logical_volume lvtype      lvsubclass Logical volume
lvm        lvdd        lvm       LVM Device Driver
posix_aio  posix_aio   node      Posix Asynchronous I/O
aio        aio         node      Asynchronous I/O (Legacy)
pty        pty         pty       Asynchronous Pseudo-Terminal
mouse      030102     usbif     USB mouse
keyboard   030101     usbif     USB keyboard
.
.
disk       540mb2     scsi      540 MB SCSI Disk Drive
disk       540mb3     scsi      540 MB SCSI Disk Drive
disk       540mb4     scsi      540 MB SCSI Disk Drive
disk       540mb5     scsi      540 MB SCSI Disk Drive
disk       730mb2     scsi      730 MB SCSI Disk Drive
disk       810mb      scsi      810 MB SCSI Disk Drive
disk       810mb2     scsi      810 MB SCSI Disk Drive
bus        pcic        pci       PCI Bus
bus        isac        pci       ISA Bus
adapter    df1000f9   pci       FC Adapter
adapter    df1000f7   pci       FC Adapter
driver     efscsi     iocb     FC SCSI I/O Controller Protocol Device
adapter    c1110358   pci       USB OHCI Adapter (c1110358)
adapter    ad100501   pci       ATA/IDE Controller Device
adapter    4f111100   pci       IBM 8-Port EIA-232/RS-422A (PCI) Adapter
adapter    ccm        pci       Name of the Common Character Mode device driver
driver     hdlc       331121b9 IBM HDLC Network Device Driver
adapter    331121b9   pci       IBM 2-Port Multiprotocol Adapter (331121b9)
adapter    2b102005   pci       GXT130P Graphics Adapter
adapter    2b101a05   pci       GXT120P Graphics Adapter
adapter    23100020   pci       IBM 10/100 Mbps Ethernet PCI Adapter (23100020)
.
.
if         tr         TR        Token Ring Network Interface
if         vi         VI        Virtual IP Address Network Interface
if         xt         XT        X.25 Network Interface
tcpip     inet       TCPIP     Internet Network Extension
swap      paging     nfs       NFS Swap DEVICE
drawer    media1     media     SCSI Device Drawer
drawer    scsi1     dasd     SCSI DASD Drawer
adapter    4f111b00   pci       IBM 128-Port Async (PCI) Adapter
concentrator 16c232   sync_pci 16-Port RAN EIA-232 for 128-Port Adapter
concentrator 16e232   sync_pci 16-Port Enhanced RAN EIA-232 for 128-Port Adapter
concentrator 16e422   sync_pci 16-Port Enhanced RAN RS-422 for 128-Port Adapter
if        at         AT        ATM Network Interface
adapter    14105300   pci       IBM PCI 25MBPS ATM Adapter (14105300)

```

2. To list all of the devices in the Customized Devices object class, type the following command:

```
lsdev -C
```

The system displays a message similar to the following output:

```

sys0      Available      System Object
sysplanar0 Available      System Planar
mem0      Available      Memory
L2cache0 Available      L2 Cache
proc0     Available 00-00   Processor
pci0      Available      PCI Bus
pci1      Available      PCI Bus
isa0      Available 10-58   ISA Bus
siota0    Available 01-Q1   Tablet Adapter
ppa0      Available 01-R1   CHRP IEEE1284 (ECP) Parallel Port Adapter
sa0       Available 01-S1   Standard I/O Serial Port
sa1       Available 01-S2   Standard I/O Serial Port

```

```

paud0      Available 01-Q2      Ultimedia Integrated Audio
siokma0    Available 01-K1      Keyboard/Mouse Adapter
fda0       Available 01-D1      Standard I/O Diskette Adapter
scsi0      Available 10-60     Wide/Ultra-2 SCSI I/O Controller
scsi1      Available 10-61     Wide/Ultra-2 SCSI I/O Controller
sa2        Available 10-68     IBM 8-Port EIA-232/RS-422A (PCI) Adapter
sa3        Available 10-70     IBM 8-Port EIA-232/RS-422A (PCI) Adapter
sa4        Available 10-78     IBM 8-Port EIA-232/RS-422A (PCI) Adapter
.
hd3        Defined          Logical volume
hd1        Defined          Logical volume
hd10opt    Defined          Logical volume
inet0      Available         Internet Network Extension
en0        Available 10-80     Standard Ethernet Network Interface
et0        Defined 10-80     IEEE 802.3 Ethernet Network Interface
lo0        Available         Loopback Network Interface
pty0       Available         Asynchronous Pseudo-Terminal
gxme0     Defined          Graphics Data Transfer Assist Subsystem
rcm0       Available         Rendering Context Manager Subsystem
aio0       Defined          Asynchronous I/O (Legacy)
posix_aio0 Defined          Posix Asynchronous I/O
tty0       Available 01-S1-00-00 Asynchronous Terminal
tty1       Available 01-S2-00-00 Asynchronous Terminal

```

3. To list the adapters that are in the Available state in the Customized Devices object class, type the following command:

```
lsdev -C -c adapter -S a
```

The system displays a message similar to the following output:

```

sa0        Available 01-S1      Standard I/O Serial Port
sa1        Available 01-S2      Standard I/O Serial Port
siokma0    Available 01-K1      Keyboard/Mouse Adapter
fda0       Available 01-D1      Standard I/O Diskette Adapter
scsi0      Available 10-60     Wide/Fast-20 SCSI I/O Controller
fcs0       Available 10-68     FC Adapter
scsi1      Available 10-88     Wide/Ultra-2 SCSI I/O Controller
fcs1       Available 20-60     FC Adapter
sioka0     Available 01-K1-00     Keyboard Adapter
siota0     Available 01-Q1      Tablet Adapter
ppa0       Available 01-R1      CHRP IEEE1284 (ECP) Parallel Port Adapter
paud0      Available 01-Q2      Ultimedia Integrated Audio
tok0       Available 10-70     IBM PCI Tokenring Adapter (14101800)
ent0       Available 10-80     IBM 10/100 Mbps Ethernet PCI Adapter (23100020)
sioma0     Available 01-K1-01     Mouse Adapter

```

4. To list all of the tape devices in the Predefined Devices object class, type the following command:

```
lsdev -P -c tape
```

The system displays a message similar to the following output:

```

tape ost    fcp    Other FC SCSI Tape Drive
tape scsd   fcp    FC SCSI Tape Drive
tape ost    iscsi  Other iSCSI Tape Drive
tape scsd   iscsi  iSCSI Tape Drive
tape 1200mb-c scsi  1.2 GB 1/4-Inch Tape Drive
tape 150mb  scsi  150 MB 1/4-Inch Tape Drive
tape 3490e  scsi  3490E Autoloading Tape Drive
tape 4mm2gb scsi  2.0 GB 4mm Tape Drive
tape 4mm4gb scsi  4.0 GB 4mm Tape Drive
tape 525mb  scsi  525 MB 1/4-Inch Tape Drive
tape 8mm    scsi  2.3 GB 8mm Tape Drive
tape 8mm5gb scsi  5.0 GB 8mm Tape Drive
tape 8mm7gb scsi  7.0 GB 8mm Tape Drive
tape 9trk   scsi  1/2-inch 9-Track Tape Drive
tape ost    scsi  Other SCSI Tape Drive
tape scsd   scsi  SCSI Tape Drive
tape 4mm2gb2 scsi  2.0 GB 4mm Tape Drive

```

5. To list the supported device classes from the Predefined Devices object class, type the following command:

```
lsdev -P -r class
```


The system displays a message similar to the following output:

```
PCM
adapter
aio
array
bus
cdrom
concentrator
container
dial
disk
diskette
drawer
driver
gxme
if
keyboard
lft
logical_volume
lpfk
lvm
memory
mouse
pdisk
planar
port
posix_aio
printer
processor
pseudo
pty
rcm
rwoptical
swap
sys
tablet
tape
tcpip
tm SCSI
tty
```

6. To list the supported subclasses in the Predefined Devices object class for the **disk** class, type the following command:

```
lsdev -P -c disk -r subclass
```

The system displays a message similar to the following output:

```
dar
fcp
fdar
ide
iscsi
scraid
scsi
vscsi
```

7. To list the name, class, subclass, and type of every device in the Available state in the Customized Devices object class with column headers, type the following command:

```
lsdev -C -H -S a -F 'name class subclass type'
```

The system displays a message similar to the following output:

name	class	subclass	type
sys0	sys	node	chrp
sysplanar0	planar	sys	sysplanar_rspc
mem0	memory	sys	totmem
L2cache0	memory	sys	L2cache_rspc
proc0	processor	sys	proc_rspc
pci0	bus	chrp	pci
pci1	bus	chrp	pci
isa0	bus	pci	isac
siota0	adapter	isa_sio	isa_tablet
ppa0	adapter	isa_sio	chrp_ecp

```

sa0      adapter isa_sio pnp501
sa1      adapter isa_sio pnp501
paud0    adapter isa_sio baud4232
siokma0  adapter isa_sio kma_chrp
fda0     adapter isa_sio pnp700
scsi0    adapter pci      sym896
scsi1    adapter pci      sym896
sa2      adapter pci      4f111100
sa3      adapter pci      4f111100
sa4      adapter pci      4f111100
ent0     adapter pci      23100020
mg20     adapter pci      2b102005
sa5      adapter pci      4f111100
sioka0   adapter kma_chrp keyboard
sioma0   adapter kma_chrp mouse
fd0      diskette siofd   fd
cd0      cdrom   scsi    scsd
hdisk0   disk    scsi    scsd
kbd0     keyboard std_k   ps2
mouse0   mouse   std_m   mse_3b
lvdd     lvm     lvm     lvdd
lft0     lft     node    lft
inet0    tcpip   TCPIP   inet
en0      if      EN      en
lo0      if      L0      lo
pty0     pty     pty     pty
rcm0     rcm     node    rcm
tty0     tty     rs232   tty
tty1     tty     rs232   tty

```

8. To list the name, class, location, and physloc of all adapter devices in the Customized Devices object class with column headers, type the following command:

```
lsdev -C -c adapter -F 'name class location physloc'
```

The system displays a message similar to the following output:

```

ent0     adapter 02-08 UTMPO.02F.00004BA-P1-C3-T1
scsi0    adapter 01-08 UTMPO.02F.00004BA-P1-C2-T1
scsi1    adapter 01-09 UTMPO.02F.00004BA-P1-C2-T2
scsi2    adapter 03-08 UTMPO.02F.00004BA-P1-C4-T1
scsi3    adapter 03-09 UTMPO.02F.00004BA-P1-C4-T2
vsa0     adapter          U9111.520.10004BA-V4-C0
vscsi0   adapter          U9111.520.10004BA-V4-C2
vscsi1   adapter          U9111.520.10004BA-V4-C3

```

9. To list all of the children of the pci0 bus, type the following command:

```
lsdev -p pci0
```

The system displays a message similar to the following output:

```

ent0 Available 10-80 IBM 10/100 Mbps Ethernet PCI Adapter (23100020)
isa0 Available 10-58 ISA Bus
scsi0 Available 10-60 Wide/Fast-20 SCSI I/O Controller
scsi1 Available 10-88 Wide/Ultra-2 SCSI I/O Controller
tok0 Available 10-70 IBM PCI Tokenring Adapter (14103e00)

```

10. To list the devices whose names are contained in the file /tmp/f, type:

```
cat /tmp/f | lsdev -l -
```

The system displays a message similar to the following output:

```

pci0 Available PCI Bus
scsi0 Available 10-60 Wide/Fast-20 SCSI I/O Controller
hdisk0 Available 10-60-00-8,0 16 Bit SCSI Disk Drive

```

11. To display the status of the devices that are exported to a WPAR as **Exported**, enter the following command:

```
# lsdev -c disk -x
```

The system displays a message similar to the following output:

```
hdisk0 Available 01-08-00-1,0 16 Bit LVD SCSI Disk Drive
hdisk1 Exported 01-08-00-2,0 Other SCSI Disk Drive
```

where the `hdisk1` device is exported to a WPAR.

Files

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

lsdisp Command

Purpose

Lists the displays available on the system.

Syntax

lsdisp [**-l**]

Description

The **lsdisp** command lists the displays currently available on the system, displaying a logical name of the display, a physical slot number of a display adapter, the type of bus to which a graphics display is attached, a display name and a description of each of the displays. This command also lists the default display.

Flags

Item	Description
------	-------------

-l	Specifies the removal of all header information and 'Default display' from format.
-----------	--

Examples

To list all available displays, enter:

```
lsdisp
```

The following output of the **lsdisp** command lists three available displays:

```
DEV_NAME  SLOT    BUS  ADPT_NAME  DESCRIPTION
ppr0      00-01   mca  POWER_G4   Midrange Graphics Adapter
gda0      00-03   mca  color_dga  Color Graphics Display Adapter
ppr1      00-04   mca  POWER_Gt3  Midrange Entry Graphics Adapter

Default display = gda0
```

Files

Item	Description
<code>bin/lsdisp</code>	Contains the lsdisp command.

lsdom Command

Purpose

Displays domain attributes.

Syntax

```
lsdom [-C] [-f] [-a Attr [Attr]...] { ALL | Name [, Name] ...}
```

Description

The **lsdom** command displays the attributes of the domain, which is defined from the domain database.

The command enables you to list attributes of all domains or specific domains. By default the **lsdom** command displays all domain attributes. To view selected attributes, use the **-a** list flag. If one or more attributes cannot be read, the **lsdom** command lists as much information as possible.

By default, the **lsdom** command lists each domain's attributes in one line. It displays attribute information as *Attribute=Value* definitions, each separated by a blank space. To list the domain attributes in stanza format, use the **-f** flag. To list the information as colon-separated records, use the **-C** flag.

Flags

Item	Description
-C	Displays the domain attributes in colon-separated records. <i>#domain:attribute1:attribute2: ...</i> <i>domain1:value1:value2: ...</i> <i>domain2:value1:value2: ...</i>
-f	Displays the output in stanzas, with each stanza identified by a domain name. Each <i>Attribute=Value</i> pair is listed in a separate line. Domain: <i>attribute1=value</i>

Parameters

Item	Description
<i>ALL</i>	Indicates that the attributes of all domains must be listed.
<i>Name</i>	Indicates the domain name whose attributes must be listed.

Security

The **lsdom** command is a privileged command. Callers of the command must have activated a role that has the following authorization to run the command successfully.

Item	Description
aix.security.domains.list	Required to run the command.

Files Accessed

Item	Description
File	Mode

Item	Description
<code>/etc/security/domains</code>	r

Examples

1. To display all attributes of the domain `hrdom`:

```
lsdom hrdom
```

All the attribute information is shown with each attribute separated by a blank space.

lsevent Command

Purpose

Lists event-monitoring information from the audit log.

Syntax

To list events from the audit log:

```
lsevent [ -O entries ] [ -B MMddhhmmyyyy ] [ -E MMddhhmmyyyy ] [ -e a | r | b ] [-i] [ -a | n node1[, node2...] ] [ -w event_node ] [-h] [-TV]
```

To list responses from the audit log:

```
lsevent -r [ -O entries ] [ -B MMddhhmmyyyy ] [ -E MMddhhmmyyyy ] [ -e { a | r | b | e | A } ... ] [-i] [ -a | n node1[, node2...] ] [-h] [-TV] [ response [response...] ]
```

To list events for a condition from the audit log:

```
lsevent [ -O entries ] [ -B MMddhhmmyyyy ] [ -E MMddhhmmyyyy ] [ -e a | r | b ] [-i] [ -a | n node1[, node2...] ] [ -w event_node ] [-h] [-TV] condition
```

To list responses for a condition from the audit log:

```
lsevent -R [ -O entries ] [ -B MMddhhmmyyyy ] [ -E MMddhhmmyyyy ] [ -e { a | r | b | e | A } ... ] [-i] [ -a | n node1[, node2...] ] [ -w event_node ] [-h] [-TV] condition [ response [response...] ]
```

To list events and responses for a condition from the audit log:

```
lsevent -A [ -O entries ] [ -B MMddhhmmyyyy ] [ -E MMddhhmmyyyy ] [ -e { a | r | b | e | A } ... ] [-i] [ -a | n node1[, node2...] ] [ -w event_node ] [-h] [-TV] condition [ response [response...] ]
```

Description

The `lsevent` command lists event-monitoring information from the audit log. The audit log contains information about monitored events or conditions, and responses that were run as a result. This information allows a system administrator to see how events are being processed. The `lsevent` command lists only the information from the audit log recorded by RSCT event response resource manager (ERRM). By using `lsevent`, you can list audit log information without knowing detailed information about ERRM audit log templates, as you would need using the `lsaudrec` command.

By default, without using options and operands, the `lsevent` command lists the events that are recorded in the audit log. These events describe the monitored events that occurred. To list the events for a particular condition, specify the condition name.

Response information can be listed separately or with the event information. Responses are run based on a condition or event occurring. Information about a response includes when it was run, what the response script was, the return code, the expected return code, standard error output, and standard output. To see standard output and the expected return code, the response resource must be defined to record it

by `mkresponse` or `chresponse`. To list only response information, use the `-r` flag. You can optionally specify one or more response names to limit the number of responses listed.

To list event information and response information for a condition, you can use the `-R` and `-A` flags with a condition name. Without `-R` and `-A`, when a condition is specified, the events for the condition are listed. Specify `-R` to list the responses for the condition. You can specify one or more response names to limit the output to those responses. Specify `-A` to list the events and the responses. You can specify one or more response names to limit the response output for `-A` as well. If a condition and at least one response are specified without specifying the `-R`, `-A`, or `-r` flags, `-R` is assumed.

The type of event listed can be controlled using the `-e` flag. You can list events, rearm events, and error events for a condition. The `-w` flag can be used to list events that occurred on a particular node. The `-w` flag has meaning when it is used in listing events. Status information is displayed when the `-i` flag is specified. When listing conditions, the status information includes showing when the condition was registered and unregistered, and when event errors occur. For response information, the status information shows that a response is about to run.

Use the `-B` and `-E` flags if you need to specify a time to limit the command output. By default, `lsevent` lists all audit log entries according to the flags specified, but you can specify a beginning time or an ending time if you are interested in a certain period. The time format is described below. The `-O` flag is used to limit the search of the audit log to the most recent records. The value used with the `-O` flag determines how many of the most recent records are searched for the other `lsevent` criteria specified. For example, using `lsevent -O 1000` causes `lsevent` to search the most recent 1000 records in the audit log for events. If `-a` or `-n` is used, `-O` cannot be used.

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

condition

Specifies the name of a condition for which audit log information is listed.

response

Specifies the name of a response for which audit log information is listed.

Flags

-a

Specifies that the `lsevent` command retrieves audit log information from all of the nodes in the cluster. 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.

-A

Specifies that event and response information for a condition is to be listed.

-B *MMddhhmmyyyy*

Specifies to list the audit log entries beginning at the time indicated. This time indicates when the audit log entry was created. Time stamps are in the form *MMddhhmmyyyy*, where *MM* is the two-digit month (01-12), *dd* is the two-digit day (01-31), *hh* is the two-digit hour (00-23), *mm* is the two-digit minute (00-59), and *yyyy* is the four-digit year. The time can be truncated from right to left, except for *MM*. If not all digits are specified, the year defaults to the current year, minutes to 0, hour to 0, and day to 01. At a minimum, the month must be specified.

-e a | r | b | e | A

Specifies the type of event to list from the audit log. The following parameters can be specified along with the **-e** flag:

a

Lists events from conditions. It is the default setting.

r

Lists rearm events from conditions.

b

List events and rearm events from conditions.

e

Lists response information that is triggered by error events. This setting is meaningful only when **-r**, **-R**, or **-A** is specified.

A

Lists all types of events (events, rearm events, and error events).

More than one event type can be specified, for example: **-e ae**.

If the **-e** flag is specified with the **-r** or **-R** flags, the response log entry for the batch-enabled condition is always displayed because the batched events file can contain all type of events.

-E MMddhhmmyyyy

Specifies to list the audit log entries up to or ending at the time indicated. This time indicates when the audit log entry was created. Time stamps are in the form *MMddhhmmyyyy*, where *MM* is the two-digit month (01-12), *dd* is the two-digit day (01-31), *hh* is the two-digit hour (00-23), *mm* is the two-digit minute (00-59), and *yyyy* is the four-digit year. The time can be truncated from right to left, except for *MM*. If not all digits are specified, the year defaults to the current year, minutes to 0, hour to 0, and day to 01. At a minimum, the month must be specified.

-i

Specifies that status information for a condition or response is to be listed. The status information includes information about event registration, event errors, and responses about to be run.

n node1[,node2...]

Specifies the node or nodes from which the audit log information is to be retrieved. If node is not specified, the local node is used. *node* is a node within the scope determined by the `CT_MANAGEMENT_SCOPE` environment variable.

-O entries

Specifies that only the latest entries in the audit log are searched for information. *entries* determines how many of the most recent records are search for the other **lsevent** criteria specified. For example, using **-O 1000** causes the `lsevent` command to search the most recent 1000 records in the audit log for events.

-r

Specifies that all command parameters are response names and that response information is to be returned for the responses specified. There are no condition names in the parameter list. If no response names are specified, then information is listed for all responses.

-R

Specifies that only the response information for a condition is to be listed.

-w event_node

Specifies the node on which the event occurred. This flag is only meaningful in listing events.

-h

Writes this 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 to standard output.

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 usage statement is 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 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.

Restrictions

If you are using the `lseven` command, you must have read access to the ERRM audit log resource on each node from which records are to be listed.

Authorization is controlled by the RMC access control list (ACL) file that exists on each node.

Implementation specifics

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

Location

`/opt/rsct/bin/lsevent`

Examples

1. To list the information for events that occurred, enter:

```
lsevent
```

2. To list the event information for a condition named `Condition1`, enter:

```
lsevent Condition1
```

3. To list the event response information, enter:

```
lsevent -r
```

4. To list the event response information for a response named `Response1`, enter:

```
lsevent -r Response1
```

5. To view the output of the event response named `Response1`, which is defined to save its output, enter:

```
lsevent -r Response1
```

6. To see the events found in the latest 1000 audit log records, enter:

```
lsevent -O 1000
```

7. To list the rearm event information for a condition named `Condition1`, enter:

```
lsevent -e r Condition1
```

lsfilt Command

Purpose

Lists filter rules from either the filter table or the IP Security subsystem.

Syntax

```
lsfilt -v 4|6 [-n fid_list] [-a] [-d]
```

Description

Use the **lsfilt** command to list filter rules and their status.

Note: Filter description fields are not listed in the kernel. No filter description text will be displayed when active or dynamic filter rules are listed.

Flags

Item	Description
-a	List only the active filter rules. The active filter rules are the rules being used by the filter kernel currently. If omitted, all the filter rules in the filter rule table will be listed.
-d	Lists the dynamic filter rules used for Internet Key Exchange (IKE) tunnels. This table is built dynamically as IKE negotiations start creating IP Security tunnels and their corresponding filter rules are added to the dynamic IKE filter table.
-n	Specifies the ID(s) of filter rule(s) that are displayed. The <i>fid_list</i> is a list of filter IDs separated by a space or "," or "-". The -n is not for active filter rules. This flag cannot be used with the -a flag.
-v	IP version of the filter rule you want to list. Valid values for this flag are 4 and 6 . If this flag is not used, both IP version 4 and IP version 6 are listed.

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.

lsfont Command

Purpose

Lists the fonts available to the display.

Syntax

lsfont [**-l**]

Description

The **lsfont** command displays a list of the fonts available to the display. The font identifier can help you change fonts using the **chfont** command.

You can use the System Management Interface Tool (SMIT) **smit lsfont** fast path to run this command.

Flags

Item	Description
------	-------------

-l	Specifies the removal of all header information from format of data.
-----------	--

Examples

To list all fonts available to the display, enter:

```
lsfont
```

The following example displays the font identifier, font name, glyph size, and font encoding for each available font:

FONT ID	FILE NAME	GLYPH SIZE	FONT ENCODING
0	Erg22.iso1.snf	12x30	ISO8859-1
1	Erg11.iso1.snf	8x15	ISO8859-1

Files

Item	Description
<code>/bin/lfont</code>	Contains the lfont command.
<code>/usr/lpp/fonts</code>	Contains fonts directory.

lsfs Command

Purpose

Displays the characteristics of file systems.

Syntax

```
lsfs [ -q ] [ -c | -l ] [ -a | -v VfsType | -u MountGroup ] [ FileSystem... ]
```

Description

The **lsfs** command displays characteristics of file systems, such as mount points, automatic mounts, permissions, and file system size. The *FileSystem* parameter reports on a specific file system. The following subsets can be queried for a listing of characteristics:

- All file systems
- All file systems of a certain mount group
- All file systems of a certain virtual file system type
- One or more individual file systems

The **lsfs** command displays additional Journaled File System (JFS) or Enhanced Journaled File System (JFS2) characteristics if the **-q** flag is specified.

You can use the System Management Interface Tool (SMIT) **smit lsfs** fast path to run this command.

Flags

Item	Description
-a	Lists all file systems (default).
-c	Specifies that the output should be in colon format.
-l	Specifies that the output should be in list format.
-q	Displays additional Journaled File System (JFS) or Enhanced Journaled File System (JFS2) characteristics specific to the file system type. This information is not reported for other virtual file system types. It is displayed in addition to other file system characteristics reported by the lsfs command.
-u MountGroup	Reports on all file systems of a specified mount group.
-v VfsType	Reports on all file systems of a specified type.

Examples

1. To show all file systems in the `/etc/filesystems` file, enter:

```
lsfs
```

2. To show all file systems of vfs type jfs, enter:

```
lsfs -v jfs
```

3. To show the file system size, the fragment size, the compression algorithm (if any), and the number of bytes per i-node as recorded in the superblock of the root file system, enter:

```
lsfs -q /
```

Files

Item	Description
<code>/etc/filesystems</code>	Lists the known file systems and defines their characteristics.

lsgroup Command

Purpose

Displays group attributes.

Syntax

```
lsgroup [ -R load_module ] [ -c | -C | -f ] [ -a List ] { ALL | Group [ ,Group ] ... }
```

Description

The **lsgroup** command displays group attributes. You can use this command to list all the system groups and their attributes or you can list all the attributes of individual groups. Since there is no default parameter, you must enter the **ALL** keyword to list all the system groups and their attributes. All the attributes that are described in the **chgroup** command are displayed. If the **lsgroup** command cannot read one or more attributes, it lists as much information as possible, but does not display empty attributes. To view a selected attribute, use the **-a** *List* flag.

Note: If the *domainlessgroups* attribute is set in the `/etc/secvars.cfg` file, the **lsgroup** command lists the users from the LDAP module and the LOCAL module, if present.

By default, the **lsgroup** command lists each group on one line. It displays attribute information as *Attribute=Value* definitions, each separated by a blank space. To list the group attributes in stanza format, use the **-f** flag. To list the information in colon-separated records, use the **-c** or **-C** flag.

You can use the System Management Interface Tool (SMIT) **smit lsgroup** fast path to run this command.

Flags

Item	Description
-a <i>List</i>	Specifies the attributes to display. The <i>List</i> parameter can include any attribute that is defined in the chgroup command, and requires a blank space between attributes. If you specify an empty list, only the group names are listed.

Item	Description
------	-------------

-c	Displays the attributes for each group in colon-separated records, as follows:
-----------	--

```
#name: attribute1: attribute2: ...
Group: value1: value2: ...
```

If a value contains a `:` symbol, then in the output `:` symbol is prefixed with the `#!` symbols.

-C	Displays the group attributes in colon-separated records that are easier to parse than the output of the -c flag:
-----------	--

```
#name:attribute1:attribute2: ...
Group1:value1:value2: ...
Group2:value1:value2: ...
```

The output is preceded by a comment line that has details about the attribute represented in each colon-separated field. If you also specify the **-a** flag, the order of the attributes matches the order specified in the **-a** flag. If you do not have a value for a given attribute, the field is still displayed, but is empty. If a value contains a `:` symbol, then in the output the `:` symbol is prefixed with `#!` symbols. The last field in each entry ends with a newline character rather than a colon.

-f	Displays the group attributes in stanzas. Each stanza is identified by a group name. Each <i>Attribute=Value</i> pair is listed on a separate line:
-----------	---

```
group:
  attribute1=value
  attribute2=value
  attribute3=value
```

-R	Specifies the loadable I&A module that is used to get the group attribute list.
-----------	---

*load_mod
ule*

If the *domainlessgroups* attribute is set in the */etc/secvars.cfg* file and the **-R LDAP** command is used, then the attribute list is obtained from the LOCAL module, if the group exists on the LOCAL module, and does not exist on the LDAP module. This condition also applies to the **-R files** command.

Exit Status

This command returns the following exit values:

Item	Description
0	The command runs successfully and all requested changes are made.
>0	An error occurred. The printed error message lists further details about the type of failure.

Security

Access Control: This command must be a general user program with execute (x) access for all users. Attributes are read with the access rights of the invoker, so all users might not be able to access all the information. This attribute depends on the access policy of your system. This command must have the *trusted computing base* attribute.

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 Accessed:

Mode	File
r	<code>/etc/group</code>
r	<code>/etc/security/group</code>
r	<code>/etc/passwd</code>

Limitations

Listing a group might not be supported by all loadable I&A modules. If the loadable I&A module does not support listing a group, then an error is returned.

Examples

1. To display the attributes of the `finance` group in the default format, enter the following command:

```
lsigroup finance
```

2. To display the `id`, members (`users`), and administrators (`adms`) of the `finance` group in stanza format, enter the following command:

```
lsigroup -f -a id users adms finance
```

3. To display the attributes of all the groups in colon-separated format, enter the following command:

```
lsigroup -c ALL
```

All the attribute information is displayed, with each attribute separated by a blank space.

4. To display the attributes of the LDAP I&A loadable module group `monsters`, enter the following command:

```
lsigroup -R LDAP monsters
```

Files

Item	Description
<code>/usr/sbin/lsigroup</code>	Contains the lsigroup command.
<code>/etc/group</code>	Contains the basic attributes of groups.
<code>/etc/security/group</code>	Contains the extended attributes of groups.
<code>/etc/passwd</code>	Contains user IDs, user names, home directories, login shell, and finger information.

lsiscsi Command

Purpose

Displays information for iSCSI target data.

Syntax

```
lsiscsi [-l AdapterName] [-g group] [-p] [-u] [-F Format]
```

Description

The `lsiscsi` command displays iSCSI target data from 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	Description
<code>-F <i>Format</i></code>	Displays the output in a user-specified format, where the <i>Format</i> parameter is a quoted list of column names, separated and possibly ended by nonalphanumeric characters or white space. If white space is used as the separator, the <code>lsiscsi</code> command displays the output in aligned columns.
<code>-g <i>group</i></code>	Specifies which group this iSCSI target is associated with. There two valid groups are <code>static</code> and <code>auto</code> . The <code>static</code> 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 <code>auto</code> group is for iSCSI targets that are automatically discovered, but require authentication information such as passwords.
<code>-l <i>AdapterName</i></code>	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.
<code>-p</code>	Displays the iSCSI target's password used for iSCSI logins from this adapter.
<code>-u</code>	Displays the Challenge Handshake Authentication Protocol (CHAP) user name that can be used for each iSCSI target.

Exit Status

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

Security

The `lsiscsi` command is executable only by root.

Examples

1. To list all iSCSI target data for the iSCSI TOE adapter `ics0`, enter:

```
lsiscsi -l ics0
```

The system displays output similar to the following:

```
10.1.2.116 3260 iqn.sn9216.iscsi-hw1
10.1.2.116 3260 iqn.sn2105.iscsi-target
```

2. To list all iSCSI target data for this host, enter:

```
lsiscsi
```

The system displays output similar to the following:

```
ics0 1 10.1.2.116 3260 iqn.sn9216.iscsi-hw1
ics0 10.1.2.116 3260 iqn.sn2105.iscsi-target
ics1 11.23.45.67 iqn.mds9216.iscsi_hw2.116 3260 iqn.sn2105.iscsi-target
```

Location

/usr/sbin/lsiscsi

Files

Item	Description
src/bos/usr/sbin/iscsia	Contains the common source files from which the iSCSI commands are built.

lsitab Command

Purpose

Lists records in the **/etc/inittab** file.

Syntax

```
lsitab { -a | Identifier }
```

Description

The **lsitab** command displays a record in the **/etc/inittab** file. You can display all of the records in the **/etc/inittab** file, or use the *Identifier* parameter to display a specific record. The *Identifier* parameter is a 14-character field that uniquely identifies an object.

Flags

It	Description
m	

-a Specifies that all records in the **/etc/inittab** file are listed.

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 record for tty2, enter:


```
lsitab "tty002"
```

The output is similar to: `tty002:2:respawn:/usr/sbin/getty /dev/tty2`

2. To display all of the records in the `/etc/inittab` file, enter:

```
lsitab -a
```

All of the records in the `/etc/inittab` file are displayed.

Files

Item	Description
<code>/etc/inittab</code>	Indicates which processes the init command starts.

lskbd Command

Purpose

List the current software keyboard map loaded into the system.

Syntax

lskbd

Description

The **lskbd** command displays the absolute pathname of the current software keyboard map loaded into the system.

To list the current software keyboard map enter:

```
lskbd
```

You can use the System Management Interface Tool (SMIT) **smit lskbd** fast path to run this command.

Note: This command can be used only on an LFT display.

Example

Following is an example of the listing displayed by the **lskbd** command:

```
The current software keyboard map = /usr/lib/nls/loc/C.lftkeymap
```

Files

Item	Description
<code>/usr/bin/lskbd</code>	Contains the lskbd command.
<code>/usr/lib/nls/loc</code>	Software keyboard map directory.

lskst Command

Purpose

Lists the entries in the kernel security tables.

Syntax

lskst **-t** *table* [**-C** | **-f**] [*Name* [, *Name*]...]

lskst **-l**

Description

The **lskst** command reads the kernel security tables (KST) and displays the information on standard output (**stdout**). The output of the **lskst** command might differ from what is displayed by the **lsauth**, **lsrole** and **lssecattr** commands if the associated file databases are modified after the databases are sent to the KST through the **setkst** command.

Specify the table to be displayed with the **-t** flag. By default, all the information in the specified table is displayed. Alternatively, a specific entry in the table can be selected by specifying the *Name* parameter.

By default, the **lskst** command lists the attributes of each entry on one line. It displays attribute information as *Attribute = Value* definitions, each separated by a blank space. To list the table attributes in stanza format, use the **-f** flag. To list the information as colon-separated records, use the **-C** flag.

Flags

Item	Description
-C	Displays the table attributes in colon-separated records as follows: <pre>#name:attribute1:attribute2:... entry_name:value1:value2:...</pre>
-f	Displays the output in stanzas, with each stanza identified by the entry name. Each <i>Attribute = Value</i> pair is listed on a separate line: <pre>entry_name: attribute1=value attribute2=value attribute3=value</pre>
-l	Displays the current value of the <code>loglevel</code> variable that is set in the kernel by using the setkst command.
-t <i>table</i>	Retrieves data from the specified security table from the KST. The parameter for the -t flag can be one of the following values: auth Authorizations table role Role table cmd Privileged command table dev Privileged device table dom Domains domobj Domain objects

Parameters

Item	Description
<i>Name</i>	Represents a specific entry of a kernel table. It can be an authorization, a role, a privileged command or a privileged device, depending on the table specified by the -t table flag.

Security

The **lskst** command is a privileged command. You must assume a role that has the following authorization to run the command successfully.

Item	Description
aix.security.kst.list	Required to run the command.

Examples

1. To retrieve all the entries in the role table from the KST, use the following command:

```
lskst -t role
```

2. To display the entry for the **/usr/bin/mycmd** command from the privileged command table in stanza format, use the following command:

```
lskst -t cmd -f /usr/bin/mycmd
```

3. To display the **aix.security** authorization table in the kernel, use the following command:

```
lskst -t auth aix.security
```

4. To retrieve all the entries in the domain object table from the KST, use the following command:

```
lskst -t domobj
```

lsldap Command

Note: If an LDAP user is created with a UID > 2 ^ 31, the **lsldap** command displays it as a negative number.

Purpose

Displays naming service objects from the configured LDAP directory server.

Syntax

```
lsldap [-a] [ entity [ entry_name | filter ] ]
```

Description

The **lsldap** command displays the naming service objects from the configured LDAP directory server. It queries the LDAP server through the **secldapclntd** daemon. Some or all of the objects of a particular entity can be listed by the **lsldap** command. By default, the **lsldap** command displays only the distinguished name (DN) of the returned objects. In addition, the **-a** flag can be used to view the attributes.

The **lsldap** command supports the following entities:

Entity	objectClass	Default attribute name
aapolicies	ibm-aixAccountingAdminPolicy	ibm-aixAdminPolicyName
aaprojects	ibm-aixAccountingProject	ibm-aixProjectName
admkeystore	ibm-usrkeystore	cn
aixpert	ibm-aixAixpert	ibm-aixpertLabel
aliases	mailGroup	cn
auditclass	AIXAuditClassStanza	auditclasstanza
auditconfig	AIXAuditConfig	auditconfig
authorizations	ibm-authorization	cn
automount	automountMap nisObject	automountMapNamenisMapName
bootparams	bootableDevice	cn
domains	ibm-aixRBACdomain	ibm-aixRBACdomainName
domobjs	ibm-aixRBACdomainObject	ibm-aixRBACdomainObjectName
efscookies	ibm-efskcookies	cn
ethers	ieee802Device	cn
group	posixgroupAIXAccessGroup	cn
grpkeystore	ibm-grpkeystore	cn
hosts	ipHost	cn
netgroup	ipNetgroup	cn
networks	ipNetwork	cn
passwd	posixAccountAIXAccount	uidusername
privcmds	ibm-privcmd	cn
privdevs	ibm-privdev	cn
protocols	ipProtocol	cn
roles	aixaccessroles	rolename
rpc	oncRpc	cn
services	ipService	cn
privfiles	ibm-privfile	cn
usrkeystore	ibm-usrkeystore	cn

The automount entity has two object classes. The `lsldap` command treats `automountMap` with higher precedence over `nisMap` by always returning `automountMap` objects if it finds any, and returning `nisMap` objects only in the absence of `automountMap` objects.

For the **passwd** and **group** entities, the `lsldap` command returns the correct objects according to the LDAP client configuration. However, the correct attribute name corresponding to the object classes must be supplied for `lsldap passwd attribute=value` queries.

If an entity name is not specified from the command line, the `lsldap` command displays container entries of the entities and any other entries that are siblings of these containers. Users must have root permissions to list the container entries.

The *entry_name* parameter is the name of the object to be queried. For example, if the entity is **passwd**, the *entry_name* is the user account name. The *entry_name* parameter is equivalent to `default attribute name = entry_name`. The `lsldap` command accepts the `*` wildcard in *entry_name* for a substring search. All entries are returned if *entry_name* is not specified.

Instead of *entry_name*, a *filter* can also be supplied to search for entries that match certain criteria. Simple filters can be specified as *attributename=attributevalue*, where *attributename* is the LDAP attribute name.

The `lsldap` command prints the result to `stdout`. If the `-a` flag is not specified, `lsldap` prints entries that are found in the form of DNs, with each DN separated by a blank line. If the `-a` flag is specified, each entry is printed in the `ldif` format, with a blank line between entries.

Flags

Item	Description
<code>-a</code>	Displays all attributes of returned objects. By default only the DN of the objects are displayed.

Exit Status

Upon success, the `lsldap` command returns 0. Upon failure, a nonzero value is returned, with one of the following error messages that are written to `stderr`:

Item	Description
EIO	Connection error.
EINVAL	Invalid parameters.
EPERM	No permission to run the operation.
ENOMEM	Not enough memory.
other errno	Other errors.

Security

The `lsldap` command can be run by any user. It is owned by the root user and security group, and has access permissions of 555.

When a non-privileged user runs the `lsldap -a passwd` command for a `netgroup` enabled LDAP module, the `lsldap` command does not display the user information if the `DisplayNetgroupUserInfo` attribute is set to `no` in the `ldap.cfg` file. By default, the user information is displayed in a `netgroup` enabled LDAP module by running the `lsldap` command irrespective of your user privileges.

When you list the **passwd** entity with the `-a` flag by root user, `lsldap` returns all attributes of the found users. However, when the same command is run by a nonprivileged user, `lsldap` returns only the same commonly readable attributes as returned by the `lsuser` command in addition to the object class information. For all other entities, regardless of which user runs the command the same output is generated.

Examples

1. To list all entries of the host entity, enter the following command:

```
lsldap hosts
```

Information similar to the following is returned:

```
dn: cn=myhost+ipHostNumber=192.3.193.46,ou=Hosts,cn=aixdata
```

```
dn: cn=starfish+ipHostNumber=192.3.193.47,ou=Hosts,cn=aixdata
```

```
dn: cn=loopback+ipHostNumber=127.0.0.1,ou=Hosts,cn=aixdata
```

2. To list host starfish and all of its attributes, enter the following command:

```
lsldap -a hosts starfish
```

Information similar to the following is returned:

```
dn: cn=starfish+ipHostNumber=192.3.193.47,ou=Hosts,cn=aixdata
objectClass: top
objectClass: ipHost
objectClass: device
ipHostNumber: 192.3.193.47
cn: loopback
cn: localhost
```

3. To list users with names that begin with the letter b, enter the following command:

```
lsldap passwd "b*"
```

Information similar to the following is returned:

```
dn: uid=bin,ou=people,cn=aixdata
dn: uid=bob,ou=people,cn=aixdata
```

4. To list user foo and its attributes, enter the following command:

```
lsldap -a passwd foo
```

Information similar to the following is returned:

```
dn: uid=foo,ou=people,cn=aixdata
uid: foo
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
objectClass: top
cn: foo
uidNumber: 259
gidNumber: 1
homeDirectory: /home/foo
loginShell: /usr/bin/ksh
shadowlastchange: 12740
userpassword: {crypt}rNnLQ9TAD2u/k
shadowmin: 5
```

5. To list users who run `/usr/bin/ksh`, enter the following command:

```
lsldap passwd loginshell=/usr/bin/ksh
```

Information similar to the following is returned:

```
dn: uid=bin,ou=people,cn=aixdata
```

```
dn: uid=bob,ou=people,cn=aixdata
```

```
dn: uid=foo,ou=people,cn=aixdata
```

Restrictions

The `lsldap` command relies on the `secldapclntd` daemon to work.

Location

`/usr/sbin/lsldap`

lslicense Command

Purpose

Displays the number of fixed licenses and the status of the floating licensing.

Syntax

```
lslicense [ -A ] [ -c ]
```

Description

The `lslicense` command displays the number of fixed licenses and the status of the floating licensing.

Flags

Item	Description
------	-------------

-A	The -A flag causes the <code>lslicense</code> command to report the current number of available fixed licenses. When the -A flag is not specified, the maximum number of fixed licenses and license status is reported.
-----------	---

-c	Displays the output in <code>:</code> (colon) form.
-----------	---

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 number of fixed licenses and the floating license status, enter:

```
lslicense
```

Example output would be:

```
Maximum number of fixed licenses is 10.  
Floating licensing is enabled.
```

2. To display the number of fixed licenses and the floating license status in a colon format, enter:

```
lslicense -c
```

Example output would be:

```
#fixed:floating  
10:on
```

3. To display license information including the number of available fixed licenses, enter:

```
lslicense -A
```

Output similar to the following will display:

```
Maximum number of fixed licenses is 2.  
Floating licensing is disabled.  
Number of available fixed licenses is 2.
```

lslpclacl Command

Purpose

Displays the access controls for the least-privilege (LP) resource class (IBM.LPCommands).

Syntax

To display the access controls for the IBM.LPCommands resource class:

- On the local node:

```
lslpclacl [-l | -i | -t | -d | -D delimiter] [-p] [-E] [-x] [-h] [-TV]
```

- On all nodes in a domain:

```
lslpclacl -a [-l | -i | -t | -d | -D delimiter] [-p] [-E] [-x] [-h] [-TV]
```

- On a subset of nodes in a domain:

```
lslpclacl { -n host1 [, host2, ... ] } [-l | -i | -t | -d | -D delimiter] [-p] [-E] [-x] [-h] [-TV]
```

Description

The `lslpclacl` command displays the access control list (ACL) that is associated with the least-privilege (LP) resource class (IBM.LPCommands). The accesses contained in the ACL entries are displayed. The IBM.LPCommands Class ACL controls access to the IBM.LPCommands class operations. By default, this command displays information in table format (-t).

This command displays the following ACL information:

Field	Description
Identity	The network identity of the user. See the lpacl command for a description of the network identity.

Field	Description
Permissions	The permissions allowed for Identity. The valid values are: a Administrator permission r Read permission (consists of the e, l, q, and v permissions) w Write permission (consists of the c, d, o, and s permissions) x Execute permission c Refresh permission d Define and undefine permission e Event permission l Enumerate permission o Online, offline, and reset permission q Query permission s Set permission v Validate permission 0 No permission
NodeName	The location of the IBM.LPCCommands resource class (for management domain scope or peer domain scope).
PeerDomain	The name of the RSCT peer domain in which the IBM.LPCCommands resource class is defined. This field is displayed when the -p flag is specified.

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

Displays the IBM.LPCCommands Class ACLs on all nodes in the domain. The CT_MANAGEMENT_SCOPE environment variable 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 ls1pc1acl command runs once for the first valid scope that the LP resource manager finds. For example, suppose that a management domain and a peer domain exist and the CT_MANAGEMENT_SCOPE environment variable is not set. In this case, ls1pc1acl -a runs

in the management domain. To run `ls1pclacl -a` in the peer domain, you must set `CT_MANAGEMENT_SCOPE` to 2.

- i**
Generates a template in a form that can be used, after appropriate editing, as file input to the `ch1pclacl` command.
- l**
Displays the information about separate lines (long format).
- t**
Displays the information in separate columns (table format). It is the default.
- d**
Displays the information using delimiters. The default delimiter is a pipe symbol (`|`). Use the `-D` flag if you want to change the default delimiter.
- D *delimiter***
Displays the information using the specified delimiter. Use this flag to specify a delimiter other than the default pipe symbol (`|`) when the information that you want to display contains pipe symbols, for example. You can use this flag to specify a delimiter of one or more characters.
- n *host1[,host2,...]***
Specifies the node in the domain from which the IBM .LPCommands Class ACL is displayed. By default, the IBM .LPCommands Class ACL is displayed on the local node. This flag is valid only in a management domain or a peer domain. 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 runs once for the first valid scope found.
- p**
Displays the name of the RSCT peer domain in which the IBM .LPCommands resource class is defined.
- E**
Displays read permission as `e1qv` instead of `r` and write permission as `cdos` instead of `w`.
- x**
Excludes the header (suppresses header printing).
- h**
Writes the command usage statement to standard output.
- T**
Writes the command trace messages to standard error.
- V**
Writes the command verbose messages to standard output.

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` 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 in processing the resources of the least-privilege (LP) resource manager. The management scope determines the set of possible target nodes where 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, unless the -a flag or the -n flag is specified.

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 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** The resource was not found.

Security

To run the `lslpclacl` command, you need read permission in the Class ACL of the `IBM.LPCommands` resource class. Permissions are specified in the LP ACLs on the contacted system. See [“lpacl Information” on page 101](#) for general information about LP ACLs and the *Administering RSCT* guide for information about modifying them.

Implementation specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) filesset for the AIX and Linux operating systems.

Location

`/opt/rsct/bin/lslpclacl`

Examples

1. To list the IBM.LPCommands Class ACLs on nodeA in table format, run this command on nodeA:

```
lslpclacl
```

The following output is displayed:

Identity	Permissions	NodeName
joe@LOCALHOST	ra	nodeA
bill@0x374bdcbe384ed38a	rwa	nodeA
jane@0x374bdcbe384ed38a	rwa	nodeA

2. To list the IBM.LPCommands Class ACLs on nodeA in long format, run this command on nodeA:

```
lslpclacl -l
```

The following output is displayed:

```
Class ACLs for LPRM
NodeName nodeA
  Identity = joe@LOCALHOST
  Permissions = ra
  Identity = bill@0x374bdcbe384ed38a
  Permissions = rwa
  Identity = jane@0x374bdcbe384ed38a
  Permissions = rwa
```

3. To list the IBM.LPCommands Class ACLs on nodeA in delimited format, run this command on nodeA:

```
lslpclacl -d
```

The following output is displayed:

```
Identity|Permissions|NodeName
joe@LOCALHOST|ra|nodeA
bill@0x374bdcbe384ed38a|rwa|nodeA
jane@0x374bdcbe384ed38a|rwa|nodeA
```

4. To list the IBM.LPCommands Class ACLs on nodeA in the active domain, run this command:

```
lslpclacl -a
```

The following output is displayed:

Identity	Permissions	NodeName
joe@LOCALHOST	ra	node1.pok.ibm.com
bill@0x374bdcbe384ed38a	rwa	node1.pok.ibm.com
jane@0x374bdcbe384ed38a	rwa	node1.pok.ibm.com
joe@LOCALHOST	ra	node2.pok.ibm.com
jane@0x374bdcbe384ed38a	rwa	node2.pok.ibm.com

5. To list the IBM.LPCommands Class ACLs on nodeA in the active domain and list the peer domain name, run this command:

```
lslpclacl -ap
```

The following output is displayed:

Identity	Permissions	NodeName	PeerDomain
joe@LOCALHOST	ra	node1.pok.ibm.com	PD1
bill@0x374bdcbe384ed38a	rwa	node1.pok.ibm.com	PD1
jane@0x374bdcbe384ed38a	rwa	node1.pok.ibm.com	PD1
joe@LOCALHOST	ra	node2.pok.ibm.com	PD1
jane@0x374bdcbe384ed38a	rwa	node2.pok.ibm.com	PD1

ls1pcmd Command

Purpose

Lists information about the least-privilege (LP) resources on one or more nodes in a domain.

Syntax

To display LP resource information:

- On the local node:

```
ls1pcmd [-A | resource_name1 [ , resource_name2 , ... ] -R RunCmdName1 [ , RunCmdName2 , ... ] [-h] [-TV]
```

- On all nodes in a domain:

```
ls1pcmd -a [-A | resource_name1 [ , resource_name2 , ... ] -R RunCmdName1 [ , RunCmdName2 , ... ] [-h] [-TV]
```

- On a subset of nodes in a domain:

```
ls1pcmd -n host1 [ , host2 , ... ] [-A | resource_name1 [ , resource_name2 , ... ] -R RunCmdName1 [ , RunCmdName2 , ... ] [-h] [-TV]
```

Description

The `ls1pcmd` command displays information about LP resources on one or more nodes in a domain. LP resources are `root` commands or scripts to which users are granted access based on permissions in the LP access control lists (ACLs). Use this command to display the attributes of one or more LP commands by specifying the `resource_name1`, [`resource_name2`, ...] parameter. If you omit this parameter, the `ls1pcmd` command lists the names of all of the LP commands. Use the `-A` flag to list all of the LP commands and all of their attributes and values. Use the `-R` flag to list one or more LP resources that have a particular `RunCmdName` value.

The `ls1pcmd` command lists the following information about defined LP resources:

Field	Description
Name	The name of the LP resource.
CommandPath	The fully-qualified path of the LP resource.
Description	A description of the LP resource.
Lock	The lock setting. Valid values are: 0 (the lock is not set) and 1 (the lock is set).
Checksum	The CheckSum value of the LP resource to which CommandPath points. The LP resource manager assigns a value of 0 if the LP resource does not exist or if the user did not update the CheckSum value after the LP resource was made available.
RunCmdName	The LP resource name that is used as a parameter with the <code>run1pcmd</code> command.
FilterScript	The path to the filter script.
FilterArg	The list of arguments to pass to <code>FilterScript</code> .

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

Displays information about one or more LP 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 `ls1pcmd` 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, `ls1pcmd -a` runs in the management domain. To run `ls1pcmd -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 listed. 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 `ls1pcmd` command runs once for the first valid scope that the LP resource manager finds.

-A

Displays all of the LP resources with their attributes and values.

-R

Display all attributes of the LP resources that have the same RunCmdName value.

-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 for which you want to display information.

Security

To run the `ls1pcmd` command, you need:

- read permission in the Class ACL of the IBM.LPCommands resource class.
- read 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 *RSCF 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 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

1. To list the names of all LP resources on the local node, enter:

```
lslpcmd
```

The output will look like this:

```
lpcommand1  
lpcommand2
```

2. To list the names and attributes of all LP resources on the local node, enter:

```
lslpcmd -A
```

The output will look like this:

```
Name=lpcommand1  
CommandPath=/tmp/my_command  
Description=  
Lock=1  
Checksum=112  
RunCmdName=lpcommand1  
FilterScript=  
FilterArg=  
-----  
Name=lpcommand2  
CommandPath=/tmp/cmds/this_command  
Description=  
Lock=0  
Checksum=0  
RunCmdName=lpcommand2  
FilterScript=  
FilterArg=  
-----
```

3. To list the attributes of the LP resource `lpcommand1` on the local node, enter:

```
lslpcmd lpcommand1
```

The output will look like this:

```
Name=lpcommand1  
CommandPath=/tmp/my_command  
Description=  
Lock=1  
Checksum=100  
RunCmdName=lpcommand1  
FilterScript=  
FilterArg=
```

4. To list the attributes of LP resources that have a `RunCmdName` value of `ipower` on the local node, enter:

```
lslpcmd -R ipower
```

The output will look like this:

```
Name=lpcommand1  
CommandPath=/opt/csm/bin/ipower  
Description=  
Lock=1  
Checksum=112  
RunCmdName=ipower  
FilterScript=/tmp/test1  
FilterArg=node1,node2,node3  
-----  
Name=lpcommand2  
CommandPath=/opt/csm/bin/ipower  
Description=  
Lock=0  
Checksum=112  
RunCmdName=ipower  
FilterScript=/tmp/test1
```



```
FilterArg=node4,node5,node6
```

```
:
```

Location

/opt/rsct/bin/lslpcmd

Contains the `lslpcmd` command

lslpp Command

Purpose

Lists installed software products.

Syntax

```
lslpp [-R { path | ALL } ] { -d | -E | -f | -h | -i | -l | -L | -p } [ -a ] [ -c ] [ -J ] [ -q ] [ -I ] [ -O { [ r ] [ s ] [ u ] } ]  
[ FilesetName ... | -b File | all ]
```

```
lslpp [-R { path | ALL } ] -w [ -c ] [ -q ] [ -O { [ r ] [ s ] [ u ] } ] [ FileName ... | all ]
```

```
lslpp [-R { path | ALL } ] -L -c [ -v ]
```

```
lslpp [-R { path | ALL } ] -S [A|O]
```

```
lslpp [-R { path | ALL } ] -e
```

Description

The **lslpp** command displays information about installed filesets or fileset updates. The *FilesetName* parameter is the name of a software product. The *File* parameter specifies a bundle file to use as a fileset list.

When only the **-l** (lowercase L) flag is entered, the **lslpp** command displays the latest installed level of the fileset specified for formatted filesets. The base level fileset is displayed for formatted filesets. When the **-a** flag is entered along with the **-l** flag, the **lslpp** command displays information about all installed filesets for the *FilesetName* specified. The **-I** (uppercase i) flag combined with the **-l** (lowercase L) flag specifies that the output from the **lslpp** command should be limited to base level filesets.

The `lslpp` command and the **compare_report** command both show information about interim fixes installed on the system. The `lslpp -L` or `lslpp -Lc` command and the `lslpp -e` command must be run by root. Any interim fix information returned is used by the **compare_report** command. The information includes an interim fix label and a level value. The interim fix label is the equivalent of a fileset name, and its level is based on the time (YY.MM.DD.HHMMSS, where YY is the year, MM is the month, DD is the day, HH is the hour, MM is the minute, and SS is the second) in which the interim fix was packaged. If a non-root user runs these commands, only software products and levels are returned, and interim fix information is not included. If a root user runs the `lslpp -e` command and the `lslpp -L` command, interim fix information can be shown.

The **-d**, **-f**, **-h**, **-i**, **-l** (lowercase L), **-L**, and **-p** flags request different types of output reports.

The **-a**, **-c**, **-J**, and **-q** flags specify the amount and format of the information that is displayed in the report.

The **-O** flag specifies that data is to come from a specified part of the fileset. The part may be the root part, **-Or**, the share part, **-Os**, or the **usr** part, **-Ou**.

The default value for the *FilesetName* parameter is **all**, which displays information about all installed software products. Pattern matching characters, such as * (asterisk) and ? (question mark), are valid in the *FilesetName* parameter. You don't have to enclose these characters in " (single quotation marks).

However, using single quotation marks prevents you from searching the contents of your present directory.

Output Values

Much of the output from the **lspp** command is understandable without an explanation. Other fields contain data that needs to be defined. The following sections define terms used in several of the output fields.

State Values

The **state** field in the **lspp** output gives the state of the fileset on your system. It can have the following values:

State	Definition
APPLIED	The specified fileset is installed on the system. The APPLIED state means that the fileset can be rejected with the installp command and the previous level of the fileset restored. This state is only valid for Version 4 fileset updates and 3.2 migrated filesets.
APPLYING	An attempt was made to apply the specified fileset, but it did not complete successfully, and cleanup was not performed.
BROKEN	The specified fileset or fileset update is broken and should be reinstalled before being used.
COMMITTED	The specified fileset is installed on the system. The COMMITTED state means that a commitment has been made to this level of the software. A committed fileset update cannot be rejected, but a committed fileset base level and its updates (regardless of state) can be removed or deinstalled by the installp command.
EFIXLOCKED	The specified fileset is installed on the system and is locked by the interim fix manager (the emgr command).
OBSOLETE	The specified fileset was installed with an earlier version of the operating system but has been replaced by a repackaged (renamed) newer version. Some of the files that belonged to this fileset have been replaced by versions from the repackaged fileset.
COMMITTING	An attempt was made to commit the specified fileset, but it did not complete successfully, and cleanup was not performed.
REJECTING	An attempt was made to reject the specified fileset, but it did not complete successfully, and cleanup was not performed.

Action Values

The **action** field in the **lspp** output identifies the installation action that was taken for the fileset. The following values may be found in this field:

Action	Definition
APPLY	An attempt was made to apply the specified fileset.
CLEANUP	An attempt was made to perform cleanup for the specified fileset.
COMMIT	An attempt was made to commit the specified fileset.
REJECT	An attempt was made to reject the specified fileset.

Status Values

The **status** field in the **lspp** output identifies the resultant status in the history of installation actions. The following values may be found in this field:

Status	Definition
BROKEN	The fileset was left in a broken state after the specified action.
CANCELED	The specified action was canceled before it completed.
COMPLETE	The commitment of the fileset has completed successfully.

Flags

Item	Description
-a	Displays all the information about filesets specified when combined with other flags. This flag shows all updates when combined with the -l flag and all history when combined with the -h flag. This flag cannot be specified with the -f flag.
-b File	Specifies a bundle file to search for fileset names. The filesets listed in the bundle are then listed as if they had been specified explicitly as <i>FilesetName</i> parameters. To mimic installp behavior, the installp image names are automatically wildcarded. For example, a bundle file entry of <code>I : bos . abc</code> will behave as if <code>bos . abc*</code> was specified as a <i>FilesetName</i> parameter. Note: This might also return results for <code>bos . abcdef</code> . If the file does not reside in one of the known bundle locations, the full path and file name, including extension, must be specified.
-c	Displays information as a list separated by colons. This flag cannot be specified with the -J flag.
-d	Displays filesets that are dependents of the specified software. A dependent fileset is one that has the specified software as a prerequisite, corequisite, iferequisite, or installed requisite.
-e	Displays every interim fix installed on the system.
-E	Lists license agreements.
-f	Displays the names of the files added to the system during installation of the specified fileset. This flag cannot be specified with the -a flag.
-h	Displays the installation and update history information for the specified fileset. You cannot use this flag with the -J flag.
-I	(uppercase i) Limits the inputs to software products.
-i	Displays the product information for the specified fileset.
-J	Generates output in a form suitable for the System Management Interface Tool (SMIT) command to list output. This flag can only be specified with the -l (lowercase L) and -L flags.
-l	(lowercase L) Displays the name, most recent level, state, and description of the specified fileset.

Item	Description
-L	<p>Displays the name, most recent level, state, type, and a description of the specified fileset. Part information (usr, root, and share) is consolidated into the same listing. For formatted filesets, it displays the most recent maintenance or technology level for the specified filesets. In addition, this flag lists any subsystem selective fixes that were installed on top of the maintenance or technology level. RPM and ISMP images are also listed.</p> <p>When combined with the -c flag, there is a difference in the Type field when used with an installp image. A blank value indicates an installp image without any updates. A value of F indicates an installp image with updates.</p> <p>When combined with the -c flag, the build date, which is specified by the year and the week in the form of yyww (for example, 0852), is displayed for the fileset, if there is one. Additional fields are displayed with the -Lc output, as indicated in the header of the output.</p>
-O	<p>Lists information for the specified part of the fileset. When the -O flag is not specified information is listed for all parts. This option is designed for use by the nim command to list software product information for diskless or dataless workstations. You can use the following flags with this flag:</p> <ul style="list-style-type: none"> -r Indicates to list information for the root part. -s Indicates to list information for the /usr/share part. -u Indicates to list information for the /usr part.
-p	Displays requisite information for the specified fileset.
-q	Suppresses the display of column headings.
-R { path ALL }	Indicates a user-specified installation location.
-S [A O]	Displays a list of automatically installed filesets and a list of optionally installed filesets. If the -S flag is followed by A , then only the automatically installed filesets are listed. If the -S flag is followed by O , then only the optionally installed filesets are listed.
-v	Displays only information from the vendor database, which contains ISMP product information. This flag is only valid when used with both the -L and the -c flags.
-w	Lists fileset that owns this file.

You must specify one of the mutually exclusive flags: **-d**, **-e**, **-E**, **-f**, **-h**, **-i**, **-l**, **-L**, **-p**, **-S**, and **-w**.

Examples

1. To list the installation state for the most recent level of installed filesets for all of the **bos.rte** filesets, type:

```
lslpp -l "bos.rte.*"
```

2. To list the installation state for the base level and updates for the fileset **bos.rte.filesystem**, type:

```
lslpp -La bos.rte.filesystem
```

3. To list the installation history information of all the filesets in the **bos.net** software package, type:

```
lslpp -ha 'bos.net.*'
```

4. To list the names of all the files of the **bos.rte.lvm** fileset, type:

```
lslpp -f bos.rte.lvm
```

5. To list the fileset that owns **installp**, type:

```
lslpp -w /usr/sbin/installp
```

Output similar to the following displays:

File Type	Fileset	
-----	-----	-----
/usr/sbin/installp	bos.rte.install	File

6. To list the fileset that owns all file names that contain **installp**, type:

```
lslpp -w "*installp*"
```

Output similar to the following displays:

File Type	Fileset	
-----	-----	-----
/usr/sbin/installp	bos.rte.install	File
/usr/clvm/sbin/linstallpv	prpq.clvm	File
/usr/lpp/bos.sysmgmt/nim/methods/c_installp	bos.sysmgmt.nim.client	File

7. To display all files in the inventory database, type:

```
lslpp -w
```

8. To display the installation state for the RPM **cdrecord** image , type:

```
lslpp -L cdrecord
```

9. To display the installation state for all the filesets contained in the Server bundle located at **/usr/sys/inst.data/sys_bundles/Server.bnd**, type:

```
lslpp -L -b Server
```

or:

```
lslpp -L -b /usr/sys/inst.data/sys_bundles/Server.bnd
```

Files

Item

/etc/objrepos/history

Description

Specifies installation and update history information of all software products on the root.

/usr/lib/objrepos/history

Specifies installation and update history information of all software products on the **/usr** file system.

/usr/share/lib/objrepos/history

Specifies installation and update history information of all software products on the **/usr/share** file system.

/etc/objrepos/lpp

Specifies installation information of all software products on the root.

/usr/lib/objrepos/lpp

Specifies installation information of all software products on the **/usr** file system.

Item	Description
/usr/share/lib/objrepos/lpp	Specifies installation information of all software products on the /usr/share file system.
/etc/objrepos/product	Specifies installation and update information of all software products on the root.
/usr/lib/objrepos/product	Specifies installation and update information of all software products on the /usr file system.
/usr/share/lib/objrepos/product	Specifies installation and update information of all the software products on the /usr/share file system.
/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.
<code>/usr/sys/inst.data/sys_bundles/</code> and <code>/usr/sys/inst.data/user_bundles/</code>	Known locations for bundle files. Bundle files should have a <code>.bnd</code> extension.

lslpracl Command

Purpose

Displays the access controls for a least-privilege (LP) resource.

Syntax

To display the access controls for an LP resource:

- On the local node:

```
lslpracl [-l | -i | -t | -d | -D delimiter] [-L] [-p] [-E] [-x] [-h] [-TV] [name]
```

- On all nodes in a domain:

```
lslpracl -a [-l | -i | -t | -d | -D delimiter] [-L] [-p] [-E] [-x] [-h] [-TV] [name]
```

- On a subset of nodes in a domain:

```
lslpracl { -n host1[, host2, ... ] } [-l | -i | -t | -d | -D delimiter] [-L] [-p] [-E] [-x] [-h] [-TV] [name]
```

Description

The `lslpracl` command displays the access control list (ACL) that is associated with a least-privilege (LP) resource. The accesses contained in the ACL entries are displayed. The Resource ACL controls access to the LP resources. If no LP resource name is specified, the Resource ACLs for all LP resources are listed. By default, this command displays information in table format (`-t`).

This command displays the following ACL information:

Field	Description
Name	The name of the LP resource. See “lpracl Information” on page 101 for a description of the network identity.

Field	Description
Identity	The network identity of the user.
Permissions	The permissions allowed for Identity. The valid values are: a Administrator permission r Read permission (consists of the e, l, q, and v permissions) w Write permission (consists of the c, d, o, and s permissions) x Execute permission c Refresh permission d Define and undefine permission e Event permission l Enumerate permission o Online, offline, and reset permission q Query permission s Set permission v Validate permission 0 No permission
NodeName	The location of the LP resource (for management domain scope or peer domain scope).
PeerDomain	The name of the RSCT peer domain in which the LP resource is defined. This field is displayed when the -p flag is specified.

If the Resource ACL indicates that the Resource Shared ACL controls access to the LP resource, the ID is displayed as `Uses Resource Shared ACL` and there is no permission value. Use the -L flag to display the Resource Shared ACL when it is used by the Resource ACLs that are being displayed.

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.

Parameters

name

Specifies the name of the LP resource.

Flags

-a

Displays the Resource ACLs on all nodes in the domain. The CT_MANAGEMENT_SCOPE environment variable 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 `ls1prac1` command runs once for the first valid scope that the LP resource manager finds. For example, suppose that a management domain and a peer domain exist and the CT_MANAGEMENT_SCOPE environment variable is not set. In this case, `ls1prac1 -a` runs in the management domain. To run `ls1prac1 -a` in the peer domain, you must set CT_MANAGEMENT_SCOPE to 2.

-i

Generates a template in a form that can be used, after appropriate editing, as file input to the `ch1prac1` command.

-l

Displays the information about separate lines (long format).

-t

Displays the information in separate columns (table format). It is the default.

-d

Displays the information using delimiters. The default delimiter is a pipe symbol (`|`). Use the `-D` flag if you want to change the default delimiter.

-D delimiter

Displays the information using the specified delimiter. Use this flag to specify a delimiter other than the default pipe symbol (`|`) when the information you want to display contains pipe symbols, for example. You can use this flag to specify a delimiter of one or more characters.

-n host1[,host2,...]

Specifies the node in the domain from which the Resource ACL is displayed. By default, the Resource ACL is displayed on the local node. This flag is valid only in a management domain or a peer domain. 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 runs once for the first valid scope found.

-L

Displays the accesses of the Resource Shared ACL if the Resource ACL indicates that access is controlled by the Resource Shared ACL.

-p

Displays the name of the RSCT peer domain in which the LP resource is defined.

-E

Displays read permission as `e1qv` instead of `r` and write permission as `cdos` instead of `w`.

-x

Excludes the header (suppresses header printing).

-h

Writes the command usage statement to standard output.

-T

Writes the command trace messages to standard error.

-v

Writes the command verbose messages to standard output.

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 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 in processing the resources of the least-privilege (LP) resource manager. The management scope determines the set of possible target nodes where 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, unless the -a flag or the -n flag is specified.

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 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** The resource was not found.

Security

To run the `lslpracl` command, you need:

- read permission in the Class ACL of the IBM.LPCCommands resource class.
- read 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 “[lpacl Information](#)” on page 101 for general information about LP ACLs and the *Administering RSCT* guide for information about modifying them.

Implementation specifics

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

Location

`/opt/rsct/bin/lslpracl`

Examples

1. To list the Resource ACLs for the LP resource `lpcommand1` on `nodeA` in table format, run this command on `nodeA`:

```
lslpracl lpcommand1
```

The following output is displayed:

```
Resource ACLs for LPRM
Name      Identity                Permissions  NodeName
lpcommand1  joe@LOCALHOST          rx          nodeA
lpcommand1  bill@0x374bdcbe384ed38a rx          nodeA
lpcommand1  jane@0x374bdcbe384ed38a rwax       nodeA
```

2. To list the Resource ACLs for the LP resource `lpcommand1` on `nodeA` in long format, run this command on `nodeA`:

```
lslpracl -l lpcommand1
```

The following output is displayed:

```
Resource ACLs for LPRM
Name lpcommand1, NodeName nodeA
  Identity =   joe@LOCALHOST
  Permissions =   rx

  Identity =   bill@0x374bdcbe384ed38a
  Permissions =   rx

  Identity =   jane@0x374bdcbe384ed38a
  Permissions =   rwax
```

3. To list the Resource ACLs for the LP resource `lpcommand1` on `nodeA` in delimited format, run this command on `nodeA`:

```
lslpracl -d lpcommand1
```

The following output is displayed:

```
Resource ACLs for LPRM
Name|Identity|Permissions|NodeName
lpcommand1|joe@LOCALHOST|rx|nodeA
```

```
lpcommand1|bill@0x374bdcbe384ed38a|rx|nodeA
lpcommand1|jane@0x374bdcbe384ed38a|rwax|nodeA
```

4. To list the Resource ACLs for the LP resource lpcommand1 in the active domain, run this command on nodeA:

```
lslpracl -a lpcommand1
```

The following output is displayed:

```
Resource ACLs for LPRM
Name      Identity      Permissions  NodeName
lpcommand1  joe@LOCALHOST  rx          nodeA.pok.ibm.com
lpcommand1  bill@0x374bdcbe384ed38a  rx          nodeA.pok.ibm.com
lpcommand1  jane@0x374bdcbe384ed38a  rwax       nodeA.pok.ibm.com
lpcommand1  joe@LOCALHOST  rx          nodeB.pok.ibm.com
lpcommand1  jane@0x374bdcbe384ed38a  rwax       nodeB.pok.ibm.com
```

5. To list the Resource ACLs for all LP resources on nodeA, run this command on nodeA:

```
lslpracl
```

The following output is displayed:

```
Resource ACLs for LPRM
Name      Identity      Permissions  NodeName
lpcommand1  joe@LOCALHOST  rx          nodeA
lpcommand1  bill@0x374bdcbe384ed38a  rx          nodeA
lpcommand1  jane@0x374bdcbe384ed38a  rwax       nodeA
lpcommand2  jim@LOCALHOST  rx          nodeA
lpcommand2  jane@0x374bdcbe384ed38a  rwax       nodeA
lpcommand3  mary          rwax       nodeA
lpcommand4  bob@LOCALHOST  rx          nodeA
lpcommand4  sam@0x374bdcbe384ed38a  rwax       nodeA
```

6. To list the Resource ACLs for the LP resource lpcommand1 in the active domain and list the peer domain name, run this command on nodeA:

```
lslpracl -ap lpcommand1
```

The following output is displayed:

```
Resource ACLs for LPRM
Name      Identity      Permission  NodeName      PeerDomain
lpcommand1  joe@LOCALHOST  rx          nodeA.pok.ibm.com  PD1
lpcommand1  bill@0x374bdcbe384ed38a  rx          nodeA.pok.ibm.com  PD1
lpcommand1  jane@0x374bdcbe384ed38a  rwax       nodeA.pok.ibm.com  PD1
lpcommand1  joe@LOCALHOST  rx          nodeB.pok.ibm.com  PD1
lpcommand1  jane@0x374bdcbe384ed38a  rwax       nodeB.pok.ibm.com  PD1
```

7. To list the Resource ACLs for the LP resource lpcommand2 on nodeA, run this command on nodeA:

```
lslpracl lpcommand2
```

The following output is displayed:

```
Resource ACLs for LPRM
Name      Identity      Permissions  NodeName
lpcommand2  Uses Resource Shared ACL  nodeA
```

8. To list the Resource ACLs for the LP resource lpcommand2 on nodeA, and show the Resource Shared ACL if it is used, run this command on nodeA:

```
lslpracl -L lpcommand2
```

The following output is displayed:

```
Resource ACLs for LPRM
Name      Identity      Permissions  NodeName
```

```
lpcommand2      bill@0x374bdcbe384ed38a  rx      nodeA
lpcommand2      jane@0x374bdcbe384ed38a  rwax    nodeA
```

lslpriac1 Command

Purpose

Displays the access controls for the least-privilege (LP) Resource Initial ACL.

Syntax

To display the access controls for the Resource Initial ACL:

- On the local node:

```
lslpriac1 [ -l | -i | -t | -d | -D delimiter ] [-p] [-E] [-x] [-h] [-TV]
```

- On all nodes in a domain:

```
lslpriac1 -a [ -l | -i | -t | -d | -D delimiter ] [-p] [-E] [-x] [-h] [-TV]
```

- On a subset of nodes in a domain:

```
lslpriac1 { -n host1 [, host2 , ... ] } [ -l | -i | -t | -d | -D delimiter ] [-p] [-E] [-x] [-h] [-TV]
```

Description

The `lslpriac1` command displays the access control list (ACL) that is associated with the least-privilege (LP) Resource Initial ACL. The accesses contained in the ACL entries are displayed. The Resource Initial ACL is used as the Initial ACL that gets copied to the Resource ACL when an LP resource is created. By default, this command displays information in table format (`-t`).

This command displays the following ACL information:

Field	Description
Identity	The network identity of the user. See “lpacl Information” on page 101 for a description of the network identity.

Field	Description
Permissions	The permissions allowed for Identity. The valid values are: a Administrator permission r Read permission (consists of the e, l, q, and v permissions) w Write permission (consists of the c, d, o, and s permissions) x Execute permission c Refresh permission d Define and undefine permission e Event permission l Enumerate permission o Online, offline, and reset permission q Query permission s Set permission v Validate permission 0 No permission
NodeName	The location of the IBM.LPCommands resource class (for management domain scope or peer domain scope).
PeerDomain	The name of the RSCT peer domain in which the IBM.LPCommands resource class is defined. This field is displayed when the -p flag is specified.

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

Displays the Resource Initial ACLs on all nodes in the domain. The CT_MANAGEMENT_SCOPE environment variable 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 `lslpriac1` command runs once for the first valid scope that the LP resource manager finds. For example, suppose that a management domain and a peer domain exist and the CT_MANAGEMENT_SCOPE environment variable is not set. In this case, `lslpriac1 -a` runs

in the management domain. To run `lslpriac1 -a` in the peer domain, you must set `CT_MANAGEMENT_SCOPE` to 2.

- i**
Generates a template in a form that can be used, after appropriate editing, as file input to the `chlpriac1` command.
- l**
Displays the information about separate lines (long format).
- t**
Displays the information in separate columns (table format). This is the default.
- d**
Displays the information using delimiters. The default delimiter is a pipe symbol (`|`). Use the `-D` flag if you want to change the default delimiter.
- D *delimiter***
Displays the information using the specified delimiter. Use this flag to specify a delimiter other than the default pipe symbol (`|`) when the information you want to display contains pipe symbols, for example. You can use this flag to specify a delimiter of one or more characters.
- n *host1[,host2,...]***
Specifies the node in the domain from which the Resource Initial ACL is displayed. By default, the Resource Initial ACL is displayed on the local node. This flag is valid only in a management domain or a peer domain. 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 runs once for the first valid scope found.
- p**
Displays the name of the RSCT peer domain in which the `IBM.LPCCommands` resource class is defined.
- E**
Displays read permission as `e1qv` instead of `r` and write permission as `cdos` instead of `w`.
- x**
Excludes the header (suppresses header printing).
- h**
Writes the command usage statement to standard output.
- T**
Writes the command trace messages to standard error.
- V**
Writes the command verbose messages to standard output.

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` 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 in processing the resources of the least-privilege (LP) resource manager. The management scope determines the set of possible target nodes where 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, unless the -a flag or the -n flag is specified.

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 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** The resource was not found.

Security

To run the `lslpriacl` command, you need read permission in the Class ACL of the `IBM.LPCommands` resource class. Permissions are specified in the LP ACLs on the contacted system. See [“lpacl Information” on page 101](#) for general information about LP ACLs and the *Administering RSCT* guide for information about modifying them.

Implementation specifics

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

Location

`/opt/rsct/bin/lslpriacl`

Examples

1. To list the Resource Initial ACLs on nodeA in table format, run this command on nodeA:

```
lslpriacl
```

The following output is displayed:

```
Resource Initial ACLs for LPRM
Identity      Permissions  NodeName
joe@LOCALHOST rx          nodeA
bill@0x374bdcbe384ed38a rwx        nodeA
jane@0x374bdcbe384ed38a rwax       nodeA
```

2. To list the Resource Initial ACLs on nodeA in long format, run this command on nodeA:

```
lslpriacl -l
```

The following output is displayed:

```
Resource Initial ACLs for LPRM
NodeName c175n06.ppd.pok.ibm.com
  Identity = joe@LOCALHOST
  Permissions = rx

  Identity = bill@0x374bdcbe384ed38a
  Permission = rwx

  Identity = jane@0x374bdcbe384ed38a
  Permissions = rwax
```

3. To list the Resource Initial ACLs on nodeA in delimited format, run this command on nodeA:

```
lslpriacl -d
```

The following output is displayed:

```
Resource Initial ACLs for LPRM
Identity|Permissions|NodeName
joe@LOCALHOST|rx|nodeA
bill@0x374bdcbe384ed38a|rwx|nodeA
jane@0x374bdcbe384ed38a|rwax|nodeA
```

4. To list the Resource Initial ACLs in the active domain, run this command:

```
lslpriacl -a
```

The following output is displayed:

```
Resource Initial ACLs for LPRM
Identity      Permissions  NodeName
joe@LOCALHOST rx          nodeA.pok.ibm.com
bill@0x374bdcbe384ed38a rwx        nodeA.pok.ibm.com
jane@0x374bdcbe384ed38a rwax       nodeA.pok.ibm.com
joe@LOCALHOST rx          nodeB.pok.ibm.com
jane@0x374bdcbe384ed38a rwax       nodeB.pok.ibm.com
```

5. To list the Resource Initial ACLs in the active domain and list the peer domain name, run this command:

```
lslpriacl -ap
```

The following output is displayed:

```
Resource Initial ACLs for LPRM
Identity      Permissions  NodeName      PeerDomain
joe@LOCALHOST rx          nodeA.pok.ibm.com PD1
bill@0x374bdcbe384ed38a rwx        nodeA.pok.ibm.com PD1
jane@0x374bdcbe384ed38a rwax       nodeA.pok.ibm.com PD1
joe@LOCALHOST rx          nodeB.pok.ibm.com PD1
jane@0x374bdcbe384ed38a rwax       nodeB.pok.ibm.com PD1
```


lslprsac1 Command

Purpose

Displays the access controls for the least-privilege (LP) Resource Shared ACL.

Syntax

To display the access controls for the Resource Shared ACL:

- On the local node:

```
lslprsac1 [-l | -i | -t | -d | -D delimiter] [-p] [-E] [-x] [-h] [-TV]
```

- On all nodes in a domain:

```
lslprsac1 -a [-l | -i | -t | -d | -D delimiter] [-p] [-E] [-x] [-h] [-TV]
```

- On a subset of nodes in a domain:

```
lslprsac1 { -n host1 [, host2 , ... ] } [-l | -i | -t | -d | -D delimiter] [-p] [-E] [-x] [-h] [-TV]
```

Description

The `lslprsac1` command displays the access control list (ACL) that is associated with the least-privilege (LP) Resource Shared ACL. The accesses contained in the ACL entries are displayed. The Resource Shared ACL controls access to LP resources in which the Resource ACL indicates that the Resource Shared ACL is used. By default, this command displays information in table format (-t).

This command displays the following ACL information:

Field	Description
Identity	The network identity of the user. See “lpacl Information” on page 101 for a description of the network identity.

Field	Description
Permissions	The permissions allowed for Identity. The valid values are: a Administrator permission r Read permission (consists of the e, l, q, and v permissions) w Write permission (consists of the c, d, o, and s permissions) x Execute permission c Refresh permission d Define and undefine permission e Event permission l Enumerate permission o Online, offline, and reset permission q Query permission s Set permission v Validate permission 0 No permission
NodeName	The location of the IBM.LPCommands resource class (for management domain scope or peer domain scope).
PeerDomain	The name of the RSCT peer domain in which the IBM.LPCommands resource class is defined. This field is displayed when the -p flag is specified.

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

Displays Resource Shared ACLs on all nodes in the domain. The CT_MANAGEMENT_SCOPE environment variable 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 `ls1prsacl` command runs once for the first valid scope that the LP resource manager finds. For example, suppose that a management domain and a peer domain exist and the CT_MANAGEMENT_SCOPE environment variable is not set. In this case, `ls1prsacl -a` runs

in the management domain. To run `ls1prsacl -a` in the peer domain, you must set `CT_MANAGEMENT_SCOPE` to 2.

- i**
Generates a template in a form that can be used, after appropriate editing, as file input to the `ch1prsacl` command.
- l**
Displays the information on separate lines (long format).
- t**
Displays the information in separate columns (table format). This is the default.
- d**
Displays the information using delimiters. The default delimiter is a pipe symbol (`|`). Use the `-D` flag if you want to change the default delimiter.
- D delimiter**
Displays the information using the specified delimiter. Use this flag to specify a delimiter other than the default pipe symbol (`|`) when the information you want to display contains pipe symbols, for example. You can use this flag to specify a delimiter of one or more characters.
- n host1[,host2,...]**
Specifies the node in the domain from which the Resource Shared ACL is displayed. By default, the Resource Shared ACL is displayed on the local node. This flag is valid only in a management domain or a peer domain. 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 runs once for the first valid scope found.
- p**
Displays the name of the RSCT peer domain in which the `IBM.LPCCommands` resource class is defined.
- E**
Displays read permission as `e1qv` instead of `r` and write permission as `cdos` instead of `w`.
- x**
Excludes the header (suppresses header printing).
- h**
Writes the command usage statement to standard output.
- T**
Writes the command trace messages to standard error.
- V**
Writes the command verbose messages to standard output.

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` 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 in processing the resources of the least-privilege (LP) resource manager. The management scope determines the set of possible target nodes where 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, unless the -a flag or the -n flag is specified.

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 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** The resource was not found.

Security

To run the `lslprsacl` command, you need read permission in the Class ACL of the `IBM.LPCommands` resource class. Permissions are specified in the LP ACLs on the contacted system. See [“lpacl Information” on page 101](#) for general information about LP ACLs and the *Administering RSCT* guide for information about modifying them.

Implementation specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) filesset for the AIX and Linux operating systems.

Location

`/opt/rsct/bin/lslprsacl`

Examples

1. To list the Resource Shared ACLs on nodeA in table format, run this command on nodeA:

```
lslprsacl
```

The following output is displayed:

```
Resource Shared ACLs for LPRM
Identity      Permissions  NodeName
joe@LOCALHOST  rx          nodeA
bill@0x374bdcbe384ed38a  rwx        nodeA
jane@0x374bdcbe384ed38a  rwax       nodeA
```

2. To list the Resource Shared ACLs on nodeA in long format, run this command on nodeA:

```
lslprsacl -l
```

The following output is displayed:

```
Resource Shared ACLs for LPRM
NodeName c175n06.ppd.pok.ibm.com
  Identity = joe@LOCALHOST
  Permissions = rx

  Identity = bill@0x374bdcbe384ed38a
  Permissions = rwx

  Identity = jane@0x374bdcbe384ed38a
  Permissions = rwax
```

3. To list the Resource Shared ACLs on nodeA in delimited format, run this command on nodeA:

```
lslprsacl -d
```

The following output is displayed:

```
Resource Shared ACLs for LPRM
Identity|Permissions|NodeName
joe@LOCALHOST|rx|nodeA
bill@0x374bdcbe384ed38a| rwx |nodeA
jane@0x374bdcbe384ed38a| rwax |nodeA
```

4. To list the Resource Shared ACLs in the active domain, run this command:

```
lslprsacl -a
```

The following output is displayed:

```
Identity      Permissions  NodeName
joe@LOCALHOST  rx          nodeA.pok.ibm.com
bill@0x374bdcbe384ed38a  rwx        nodeA.pok.ibm.com
jane@0x374bdcbe384ed38a  rwax       nodeA.pok.ibm.com
joe@LOCALHOST  rx          nodeB.pok.ibm.com
jane@0x374bdcbe384ed38a  rwax       nodeB.pok.ibm.com
```

5. To list the Resource Shared ACLs in the active domain and list the peer domain name, run this command:

```
lslprsacl -ap
```

The following output is displayed:

```
Resource Shared ACLs for LPRM
Identity      Permissions  NodeName      PeerDomain
joe@LOCALHOST  rx          nodeA.pok.ibm.com  PD1
bill@0x374bdcbe384ed38a  rwx        nodeA.pok.ibm.com  PD1
jane@0x374bdcbe384ed38a  rwax       nodeA.pok.ibm.com  PD1
joe@LOCALHOST  rx          nodeB.pok.ibm.com  PD1
jane@0x374bdcbe384ed38a  rwax       nodeB.pok.ibm.com  PD1
```

lslv Command

Purpose

Displays information about a logical volume.

Syntax

To Display Logical Volume Information

lslv [**-L**] [**-l** **-m**] [**-n** *PhysicalVolume*] *LogicalVolume*

To Display Logical Volume Allocation Map

lslv [**-L**] [**-n** *PhysicalVolume*] **-p** *PhysicalVolume* [*LogicalVolume*]

Description

The **lslv** command displays the characteristics and status of the *LogicalVolume* or lists the logical volume allocation map for the physical partitions on the *PhysicalVolume*. The logical volume can be a name or identifier.

Note: If the **lslv** command cannot find information for a field in the Device Configuration Database, it will insert a question mark (?) in the value field. As an example, if there is no information for the LABEL field, the following is displayed:

```
LABEL: ?
```

The command attempts to obtain as much information as possible from the description area when it is given a logical volume identifier.

You can use the System Management Interface Tool (SMIT) **smit lslv** fast path to run this command.

Flags

Item	Description
-L	Specifies no waiting to obtain a lock on the Volume group. Note: If the volume group is being changed, using the -L flag gives unreliable date.

Item	Description
-l	<p>Lists the following fields for each physical volume in the logical volume:</p> <p>PV Physical volume name.</p> <p>Copies The following three fields:</p> <ul style="list-style-type: none"> • The number of logical partitions containing at least one physical partition (no copies) on the physical volume • The number of logical partitions containing at least two physical partitions (one copy) on the physical volume • The number of logical partitions containing three physical partitions (two copies) on the physical volume <p>In band The percentage of physical partitions on the physical volume that belong to the logical volume and were allocated within the physical volume region specified by Intra-physical allocation policy.</p> <p>Distribution The number of physical partitions allocated within each section of the physical volume: outer edge, outer middle, center, inner middle, and inner edge of the physical volume.</p>
-m	<p>Lists the following fields for each logical partition:</p> <p>LPs Logical partition number.</p> <p>PV1 Physical volume name where the logical partition's first physical partition is located.</p> <p>PP1 First physical partition number allocated to the logical partition.</p> <p>PV2 Physical volume name where the logical partition's second physical partition (first copy) is located.</p> <p>PP2 Second physical partition number allocated to the logical partition.</p> <p>PV3 Physical volume name where the logical partition's third physical partition (second copy) is located.</p> <p>PP3 Third physical partition number allocated to the logical partition.</p>
-n <i>PhysicalVolume</i>	<p>Accesses information from the specific descriptor area of <i>PhysicalVolume</i> variable. The information may not be current since the information accessed with the -n flag has not been validated for the logical volumes. If you do not use the -n flag, the descriptor area from the physical volume that holds the validated information is accessed and therefore the information that is displayed is current. The volume group need not be active when you use this flag.</p>

Item	Description
-p <i>PhysicalVolume</i>	<p>Displays the logical volume allocation map for the <i>PhysicalVolume</i> variable. If you use the <i>LogicalVolume</i> parameter, any partition allocated to that logical volume is listed by logical partition number. Otherwise, the state of the partition is listed as one of the following:</p> <p>used Indicates that the partition is allocated to another logical volume.</p> <p>free Indicates that the specified partition is not being used on the system.</p> <p>stale Indicates that the specified partition is no longer consistent with other partitions. The computer lists the logical partitions number with a question mark if the partition is stale.</p>

If no flags are specified, the following status is displayed:

Item	Description
LOGICAL VOLUME	Name of the logical volume. Logical volume names must be unique system-wide and can range from 1 to 15 characters.
VOLUME GROUP	Name of the volume group. Volume group names must be unique system-wide and can range from 1 to 15 characters.
LOGICAL VOLUME IDENTIFIER (LV identifier)	Identifier of the logical volume.
PERMISSION	Access permission; read-only or read-write.
VOLUME GROUP STATE (VG state)	State of the volume group. If the volume group is activated with the varyonvg command, the state is either active/complete (indicating all physical volumes are active) or active/partial (indicating all physical volumes are not active). If the volume group is not activated with the varyonvg command, the state is inactive .
LOGICAL VOLUME STATE (LV state)	State of the logical volume. The Opened/stale status indicates the logical volume is open but contains physical partitions that are not current. Opened/syncd indicates the logical volume is open and synchronized. Closed indicates the logical volume has not been opened.
TYPE	Logical volume type.
WRITE VERIFY	Write verify state of on or off.
MIRROR WRITE CONSISTENCY	Mirror write consistency state of on or off.
MAX LPs	Maximum number of logical partitions the logical volume can hold.
PP size	Size of each physical partition.
COPIES	Number of physical partitions created for each logical partition when allocating.
SCHEDULE POLICY(Sched policy)	Sequential or parallel scheduling policy.
LPs	Number of logical partitions currently in the logical volume.
PPs	Number of physical partitions currently in the logical volume.

Item	Description
Stale partitions	Number of physical partitions in the logical volume that are not current.
BB POLICY	Bad block relocation policy.
INTER-POLICY	Inter-physical allocation policy.
INTRA-POLICY	Intra-physical allocation policy.
UPPER BOUND	If the logical volume is super strict, upper bound is the maximum number of disks in a mirror copy.
RELOCATABLE	Indicates whether the partitions can be relocated if a reorganization of partition allocation takes place.
MOUNT POINT	File system mount point for the logical volume, if applicable.
LABEL	Specifies the label field for the logical volume.
Each LP copy on a separate PV?	The strictness value. Current state of allocation, strict, nonstrict, or superstrict. A strict allocation states that no copies for a logical partition are allocated on the same physical volume. If the allocation does not follow the strict criteria, it is called nonstrict. A nonstrict allocation states that at least one occurrence of two physical partitions belong to the same logical partition. A superstrict allocation states that no partition from one mirror copy may reside the same disk as another mirror copy.
SERIALIZE IO?	Serialization of overlapping IOs state of yes or no. If serialization is turned on (yes), then overlapping IOs are not allowed on a block range, and only a single IO in a block range is processed at any one time. Most applications, such as file systems and databases, perform serialization; therefore, serialization should be turned off (no). The default setting for new logical volumes is no.
STRIPE WIDTH	The number of physical volumes being striped across.
STRIPE SIZE	The number of bytes per stripe.
INFINITE RETRY	Lists the infinite retry option of the logical volume.
PREFERRED READ	Lists the preferred logical volume copy for the read operation.
>> ENCRYPTION	Lists the data encryption option of the logical volume. This information is available in AIX 7 with 7200-05, or later. <<<

Examples

1. To display information about the `testlv` logical volume, enter:

```
lslv testlv
```

Information about the `testlv` logical volume, its logical and physical partitions, and the volume group to which it belongs is displayed as shown in the following example output:

```
LOGICAL VOLUME:    testlv                VOLUME GROUP:    testvg
LV IDENTIFIER:    000e8b6e0000d900000001476c303bc8.1 PERMISSION:      read/write
VG STATE:         active/complete    LV STATE:        closed/syncd
TYPE:             jfs                WRITE VERIFY:    off
MAX LPs:          512                PP SIZE:         128 megabyte(s)
COPIES:           3                  SCHED POLICY:    parallel
LPs:              10                PPs:             30
```

```

STALE PPs:          0          BB POLICY:          relocatable
INTER-POLICY:       minimum    RELOCATABLE:       yes
INTRA-POLICY:       middle     UPPER_BOUND:       32
MOUNT POINT:        N/A        LABEL:              None
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
INFINITE RETRY:     no          PREFERRED_READ:    3

```

2. To display the logical volume allocation map for `hdisk2`, enter:

```
lslv -p hdisk2
```

An allocation map for `hdisk2` is displayed, showing the state of each partition. Since no *LogicalVolume* parameter was included, the map does not contain logical partition numbers specific to any logical volume.

3. To display information about logical volume `lv03` by physical volume, enter:

```
lslv -l lv03
```

The characteristics and status of `lv03` are displayed, with the output arranged by physical volume.

4. To display information about physical volume `hdisk3` gathered from the descriptor area on `hdisk2`, enter:

```
lslv -n hdisk2 -p hdisk3 lv02
```

An allocation map, using the descriptor area on `hdisk2`, is displayed. Because the *LogicalVolume* parameter is included, the number of each logical partition allocated to that logical volume is displayed on the map.

5. To display information about a specific logical volume, using the identifier, enter:

```
lslv 00000256a81634bc.2
```

All available characteristics and status of this logical volume are displayed.

File

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

lsmaster Command

Purpose

Displays the characteristics for the configuration of an NIS controller server.

Syntax

```
/usr/sbin/lsmaster [ -c | -l ]
```

Description

The **lsmaster** command displays the characteristics of an NIS controller server. The host names of the worker servers are listed along with the currently served domains.

You can use the System Management Interface Tool (SMIT) **smit lsmaster** fast path to run this command.

Flags

Item	Description
------	-------------

- | | |
|-----------|--|
| -c | Specifies that the output should be in colon format. This flag is the default. |
| -l | Specifies that the output should be in list format. |

Examples

To list the NIS controller server characteristics in colon format, enter:

```
lsmaster -c
```

Files

Item	Description
<code>/var/yp/domainname</code> directory	Contains the NIS maps for the NIS domain.

lsmcode Command

Purpose

Displays microcode and firmware levels of the system and adapters and devices.

Syntax

```
lsmcode [ -A | -d Name ] [ -r | -c ] [ -t [ service | system | adapter | adapter-boot | raid-dasd | backplane ] ]
```

Description

The **lsmcode** command when run without any flags, displays the platform system firmware microcode level and the service processor microcode levels, if supported. Not all systems contain a service processor, nor do all systems support displaying the system processor level. Information on a specific device is displayed with the **-d** flag.

If you run the **lsmcode** command with the **-r** or **-c** flag, it displays the microcode levels in a **printf** format; that is, not a menu. This method is preferred if running **lsmcode** from a script.

Flags

Item	Description
-A	Displays microcode level information for all supported devices. Using this flag assumes the -r flag.
-c	Displays the microcode/firmware levels without using menus.
-d <i>Name</i>	Displays microcode level information for the named device.

Item	Description
-r	<p>Displays the microcode/firmware levels in a tabular format. The microcode level is preceded by a Type if supported or required.</p> <p>Current supported Types are as follows:</p> <p>system System Firmware</p> <p>service Service Processor</p> <p>adapter Adapter Functional microcode</p> <p>adapter-boot Adapter Boot Microcode</p> <p>raid-dasd DASD Microcode in a RAID array</p> <p>backplane Backplane Microcode in a RAID subsystem</p>
-t	<p>Specifies the microcode type. The microcode level information of the specified type is displayed. You can use the -t flag only when a device supports multiple types. You can use the -A flag to find any devices supporting multiple types.</p>

Examples

1. To display the system firmware level and service processor (if present), type:

```
lsmcode -c
```

The system displays a message similar to the following:

```
System Firmware level is TCP99256
```

2. To display the system firmware level and service processor (if present) in raw mode, type:

```
lsmcode -r
```

The system displays a message similar to the following:

```
system:TCP99256
```

3. To display the adapter microcode levels for a RAID adapter **scraid0**, type:

```
lsmcode -r -d scraid0
```

The system displays a message similar to the following:

```
adapter:4.20.18|adapter-boot:4.00.26
raid-dasd:22:FFC #:DDYS-T0.524D3031.53393446
raid-dasd:26:FFC #:DDYS-T0.524D3031.53393446
raid-dasd:2e:FFC #:DDYS-T0.525A3034.53393243
```

4. To display the microcode level for a tape drive **rmt0**, type:

```
lsmcode -r -d rmt0
```

The system displays a message similar to the following:

```
C009
```

5. To display the microcode level for all supported devices, type:

```
lsmcode -A
```

The system displays a message similar to the following:

```
sys0!system:TCP99256
rmt0!C009
scaid0!adapter:4.20.18|adapter-boot:4.00.26
raid-dasd:22:FFC #:DDYS-T0.524D3031.53393446
raid-dasd:26:FFC #:DDYS-T0.524D3031.53393446
raid-dasd:2e:FFC #:DDYS-T0.525A3034.53393243
.....
```

6. To display the microcode level of the **adapter** microcode type for a RAID adapter **scaid0**, type:

```
lsmcode -rd scaid0 -t adapter
```

The system displays a message similar to the following:

```
adapter:4.50.01
```

Files

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

Related Information

The **diag** command.

lsmksysb Command

Purpose

Lists or restores the contents of a volume group backup on a specified media.

Syntax

```
lsmksysb [ -b blocks ] [ -f device ] [ -a ] [ -c ] [ -l ] [ -n ] [ -r ] [ -s ] [ -d path ] [ -B ] [ -D ] [ -L ] [ -V ] [ file_list ]
```

Description

The **lsmksysb** command lists the contents of a volume group backup from tape, file, CD-ROM, or other source and can be used to restore files from a valid backup source. The **lsmksysb** command also works for multi-volume backups such as multiple CDs, DVDs, USB disks, or tapes.

The **lsmksysb -r** and **restorevgfiles** commands perform identical operations and must be considered interchangeable.

Flags

Item	Description
-a	Verifies the physical block size of the tape backup, as specified by the -b block flag. You might need to alter the block size if necessary to read the backup. The -a flag is valid only when a tape backup is used.
-b blocks	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 defaults to 100.

Item	Description
-B	<p>Prints the volume group backup log to stdout.</p> <p>This flag displays the past 256 backups (roughly). The log is in alog format and is kept in /var/adm/ras/vgbackuplog. Each line of the log is a semicolon-separated list of the file or device name, the command that is used to make backup, date, shrink size, full size of the backup, and recommended maintenance or technology level (if any).</p> <p>Note: The shrink size is the size of the data on all file systems. The full size is total size of each file system (unused + data).</p>
-c	Produces colon-separated output. This flag works only with the -l and -L flags.
-d path	Specifies the directory path to which the files are 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.
-f device	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> defaults to /dev/rmt0 .
-l	<p>Displays useful information about a volume group backup.</p> <p>This flag requires the -f device flag. This flag causes lsmksysb to display information such as volume group, date and time backup was made, uname output from backed up system, oslevel, recommended maintenance or technology level, backup size in megabytes, and backup shrink size in megabytes. The shrink size is the size of the data on all file systems. The full size is the total size of each file system (unused + data). The -l flag also displays the logical volume and file system information of the backed up volume group, equivalent to running "lsvg -l vgname".</p>
-L	<p>Displays lpp fileset information about a mksysb backup only.</p> <p>This flag requires the -f device flag and displays the equivalent information to that produced by invoking "lslpp -l" on the running backed up system. This flag does not produce output about any volume group backup other than that produced by mksysb.</p>
-n	Does not restore ACLs, PCLs, or extended attributes
-r	Specifies to restore the backup files, as defined by the <i>file-list</i> parameter. If the <i>file-list</i> parameter is not specified, then all files in the backup are restored. If the -r flag is not used, then executing the lsmksysb command lists only the files in the specified backup.
-s	Specifies that the backup source is a user volume group and not rootvg.
-V	<p>Verifies a tape backup.</p> <p>This flag requires the -f device flag and works for tape devices only. The -V flag causes lsmksysb to verify the readability of the header of each file on the volume group backup and print any errors that occur to stderr.</p>

Parameters

Item	Description
<i>file_list</i>	Identifies the list of files to be restored. This parameter is used only when the -r flag is specified. The full path of the files relative to the current directory must be specified in the space-separated list. All files in the specified directory are 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 list the contents of the system backup that is on the default device **/dev/rmt0**, enter the following command:

```
lsmksysb
```

2. To list the contents of the system backup that is on device **/dev/cd1**, enter the following command:

```
lsmksysb -f /dev/cd1
```

3. To list the contents of the system backup that is on device **/dev/cd1**, which is a user volume group that is not rootvg, enter the following command:

```
lsmksysb -f /dev/cd1 -s
```

4. To restore **/etc/filesystems** from the system backup that is on device **/dev/cd1**, enter the following command:

```
lsmksysb -f /dev/cd1 -r ./etc/filesystems
```

5. To restore all files in the **/myfs/test** directory of the non-rootvg backup, which is on device **/dev/cd1**, and write the restored files to **/data/myfiles**, enter the following command:

```
lsmksysb -f /dev/cd1 -r -s -d /data/myfiles ./myfs/test
```

6. To display colon separated lpp information about a **mksysb** backup tape that is on **/dev/rmt0**, enter the following command:

```
lsmksysb -Lc -f /dev/rmt0
```

7. To display the backup log of the volume group to **stdout**, enter the following command:

```
lsmksysb -B
```

8. To list volume group and general backup data about a backup that is on **/tmp/mybackup**, enter the following command:

```
lsmksysb -l -f /tmp/mybackup
```

9. To verify the readability of each header on a volume group backup tape in **/dev/rmt0**, enter the following command:

```
lsmksysb -V -f /dev/rmt0
```

10. To list the contents of the system backup that is on device **/dev/usbms0**, enter the following command:

```
lsmksysb -f /dev/usbms0
```

Files

Item	Description
<code>/usr/bin/lsmksysb</code>	Contains the lsmksysb command

lsmp Command

Purpose

Displays mirror pool information.

Syntax

lsmp [-L] [-A] [-n *descriptorphysicalvolume*] [-m *mirrorpoolname*] *vgname*

Description

The **lsmp** command displays mirror pool information for a volume group.

Note: The volume group must be varied on to list the mirror pool information. If the volume group is varied off, you must specify the source disk name using the **-n** flag to list the information.

Flags

Item	Description
-A	Displays information specific to asynchronous mirroring, if it exists.
-L	Specifies no waiting to obtain a lock on the volume group. Note: If the volume group is being changed, using the -L flag gives unreliable data.
-m <i>mirrorpoolname</i>	Displays mirror pool information only for the mirror pool that is specified by the <i>mirrorpoolname</i> parameter. If you do not specify the flag, information for all mirror pools that belong to the volume group is displayed.
-n <i>descriptorphysicalvolume</i>	Accesses information from the descriptor area that is specified by the <i>descriptorphysicalvolume</i> variable. The information might not be current because the information accessed with the -n flag has not been validated for the logical volumes. If you do not use the -n flag, the descriptor area from the physical volume that holds the validated information is accessed, and therefore the current information is displayed. The volume group does not need to be active when you use the -n flag.

Parameters

Item	Description
<i>vgname</i>	Specifies the volume group name.

Examples

1. To display all mirror pool information for a volume group, enter the following command:

```
lsmp vg1
```

The following output is displayed:


```
VOLUME GROUP:      vg1                Mirror Pool Super Strict: no
MIRROR POOL:       mp1                Mirroring Mode:      SYNC
MIRROR POOL:       mp2                Mirroring Mode:      SYNC
```

2. To display all mirror pool information for a volume group and include asynchronous mirroring information in the output, enter the following command:

```
lsmg -A glvm_vg
```

The following output is displayed:

```
VOLUME GROUP:      glvm_vg                Mirror Pool Super Strict: yes
MIRROR POOL:       mp_bvr                Mirroring Mode:      ASYNC
ASYNC MIRROR STATE: inactive            ASYNC CACHE LV:      mp_pok_lv
ASYNC CACHE VALID: yes                  ASYNC CACHE EMPTY:   yes
ASYNC CACHE HWM:   60                    ASYNC DATA DIVERGED: no
MIRROR POOL:       mp_pok                Mirroring Mode:      ASYNC
ASYNC MIRROR STATE: active              ASYNC CACHE LV:      bvr_pok_lv
ASYNC CACHE VALID: yes                  ASYNC CACHE EMPTY:   no
ASYNC CACHE HWM:   90                    ASYNC DATA DIVERGED: no
```

3. To display information for the mirror pool mp_pok from the glvm_vg volume group and include asynchronous mirroring information in the output, enter the following command:

```
lsmg -A -m mp_pok glvm_vg
```

The following output is displayed:

```
VOLUME GROUP:      glvm_vg                Mirror Pool Super Strict: yes
MIRROR POOL:       mp_pok                Mirroring Mode:      ASYNC
ASYNC MIRROR STATE: active              ASYNC CACHE LV:      bvr_pok_lv
ASYNC CACHE VALID: yes                  ASYNC CACHE EMPTY:   no
ASYNC CACHE HWM:   90                    ASYNC DATA DIVERGED: no
```

lsmg Command

Purpose

Displays information about the MultiPath I/O (MPIO) storage devices.

Syntax

```
lsmg [ -l device_name ] [ -o ]
```

```
lsmg -S [ -l device_name ] [ -d ]
```

```
lsmg -z [ -l device_name ]
```

```
lsmg -q [ -l device_name ]
```

```
lsmg -a [ -r ] [ -e ] [ -z ] [ -f ]
```

```
lsmg -h
```

Description

The **lsmg** command displays information about AIX MPIO storage devices. This command can be used only for AIX MPIO storage devices that are controlled by path-control modules (PCMs) that are enabled for the **lsmg** command support. Some AIX® MPIO storage devices do not support the **lsmg** command queries. However, all AIX MPIO storage devices support Path operational status queries.

The **lsmg** command displays the following types of information:

- [Path operational status](#)
- [Additional NVMe extended status](#)
- [Path statistics](#)
- [Device inquiry data](#)
- [Parent adapter information](#)

Path operational status

You can run the **lsmPIO** command without any flags or with the **-l** flag to display the path operational status.

```
# lsmPIO
name path_id status path_status parent connection
=====
hdisk1234 0 Enabled Opt,Rsv fscsi0 500a098186a7d4ca,0008000000000000
hdisk1234 1 Enabled Non fscsi0 500a098196a7d4ca,0008000000000000
hdisk1234 2 Enabled Opt,Se1 fscsi1 500a098186a7d4ca,0008000000000000
hdisk1234 3 Enabled Non fscsi1 500a098196a7d4ca,0008000000000000
```

The output is similar to the output that is displayed by running the **lspath** command as shown in the following command:

```
lspath -F "path_id status parent connection"
```

The valid values of the status column are Enabled, Disabled, Failed, or Missing. The extended path_status field might contain one or more three-letter status abbreviations to provide more detailed path status.

Note: >| Not all extended path_status fields are applicable to all MPIO storage devices. Some path_status values appear only if the storage area network (SAN) fabric supports notification about SAN congestion. |<

Following are the possible values for the path_status field:

Opt

Indicates that the MPIO disk path is an optimized path. This value indicates an MPIO disk path that attaches to a preferred controller in a device that has multiple controllers. The PCM selects one of the preferred MPIO disk paths for I/O operations, whenever possible.

Non

Indicates that the MPIO disk path is a nonoptimized path. On a device with preferred MPIO disk paths, this path is not considered as preferred path. The PCM avoids the selection of this path for I/O operations, unless all preferred paths fail.

>|Chg

Indicates that the access characteristics of the MPIO disk are changing on the storage target. This state is temporary until the path transitions to a new state. |<

>|Los

Indicates that the MPIO disk path lost access to data permanently and is no longer usable in the host. The storage target must be evaluated or diagnosed for possible problems. |<

>|Sby

Indicates that the MPIO disk path is a standby path. If an MPIO disk path is in the Sby state, the disk path cannot be used for I/O operations. A device-dependent operation on the storage device can change the state of the MPIO disk path to either Opt or Non state and then the MPIO disk path can be used for I/O operations. |<

>|NAv

Indicates that the MPIO disk path is not available. If an MPIO disk path is in the NAv state, the disk path cannot be used for I/O operations. A device-dependent operation on the storage device can

change the state of the MPIO disk path to either Opt or Non state and then the MPIO disk path can be used for I/O operations. |<

>|Trn

Indicates that the MPIO disk path is switching between two valid states. If an MPIO disk path is in the Trn state, the disk path cannot be used for I/O operations. The state of the disk path changes from Trn state to another state automatically. An MPIO disk path cannot be in the Trn state for a long duration. However, the AIX operating system might not detect the change in the path status immediately. |<

Act

Indicates that the MPIO disk path is an active path on a device that has active and passive controllers. The PCM selects active MPIO disk paths for I/O operations on such a device.

Pas

Indicates that the MPIO disk path is a passive path on a device that has active and passive controllers. The PCM avoids the selection of passive paths.

Sel

Indicates that the MPIO disk path is being selected for I/O operations, for the time when the **lsmpio** command is to be run.

Rsv

Indicates that the MPIO disk path experienced an unexpected reservation conflict. This value might indicate a usage or configuration error, with multiple hosts that access the same disk.

Fai

Indicates that the MPIO disk path experienced a failure. It is possible for a path to have a path status value of Enabled and still have an extended path status value of Fai. This scenario indicates that operations that are sent on this MPIO disk path are failing, but AIX MPIO did not mark the path as Failed. Sometimes, AIX MPIO leaves one path to the device in Enabled state, even when all MPIO disk paths are experiencing errors.

Deg

Indicates that the MPIO disk path is in a degraded state. This scenario indicates that the MPIO disk path was being used for I/O operations. Those operations experienced errors, thus causing the PCM to temporarily avoid the use of the path. Any additional errors might cause the MPIO disk path to fail.

>|EEH

Indicates that the Fibre Channel adapter that is used by the MPIO disk path is experiencing an Enhanced Error Handling (EEH) event. When the Fibre Channel adapter, which is using the MPIO disk path for I/O operations, experiences an EEH error, the EEH event causes the PCM to temporarily avoid the MPIO disk path. MPIO disk paths with a status of EEH are considered as degraded because the I/O operation is blocked on the MPIO disk path during the EEH event. The path-selection algorithms prefer the Deg path status over the EEH path status and prefer EEH path status over an unusable path status, such as RSV. After the Fibre Channel adapter is back online and resumes I/O operations, the path status is changed from EEH to Deg, and the I/O operations resume on the MPIO disk path. |<

Clo

Indicates that the MPIO disk path is closed. A device is considered as closed if all MPIO disk paths that are connected to the device are closed. If only some MPIO disk paths are closed, then those paths might have experienced errors during the last time the device was opened. The AIX MPIO periodically attempts to recover closed paths until the device path is open.

>|PFa

Indicates that the remote port failed. When the PCM receives an event that indicates that the remote port that is used by the MPIO disk path is no longer part of the SAN fabric, the PCM marks the path as a failed path. The PCM clears the PFa state when the remote port rejoins the SAN fabric. |<

> **PCn**

Indicates that the SAN fabric reported network traffic congestion to the remote port that is used by the MPIO disk path. The PCM does not use this MPIO disk path if other MPIO disk paths without the network traffic congestion are available for the MPIO device. The PCM automatically clears the PCn state if the SAN fabric does not report any recent network traffic congestion events. |<

> **PDg**

Indicates a degraded remote port. The degraded port might experience many errors even though the MPIO disk path to the remote port remains active. Such degraded port cannot process data. The PCM avoids such disk path. To clear the degraded status of the port, you must first disable the degraded port on the switch that the port belongs to, and then enable that port. If the PDg state occurs repeatedly on a path, investigate the ports and fibers that are used by the path to identify any issues with the SAN hardware. |<

> **LCn**

Indicates that the link that is associated with the AIX adapter is congested because large amount of data is being sent to or from the AIX adapter. The PCM avoids such MPIO disk paths if other MPIO disk paths without the link congestion are available. The PCM automatically clears the LCn state if the SAN fabric does not report any recent link congestion events. |<

> **Deferred**

> | Indicates a newly added path to a currently open disk. The MPIO disk path cannot be opened and used immediately because of internal restrictions. However, the disk path can be opened and used when the disk is reopened. |<|<

Additional NVMe extended status

When the status of the MPIO disk path is Failed, the following extended status can be seen as an extra detail as to why the path failed:

<i>Table 4. NVMe extended status</i>	
Status	Description
Offline	Indicates that the disk path is manually set to offline. You must not manually set the path to offline unless advised by technical support.
FCloginErr, LinkDwnErr, CassErr, CiocErr, EndpntMiss	Indicates various NVMe over Fibre Channel error conditions. The most likely condition is CassErr that means the Create Association service to the target controller failed. The CassErr condition normally means the host (NQN) is not properly mapped to the target device.
CtlrRstErr, CtlrRdyErr, AdminCmdErr, ConnectErr, AuthErr, FabCmdErr, HbaErr, CrqErr	Indicates various NVMe error conditions that are useful to technical support personnel.
FCDDErr, HostDDErr	Indicates driver errors such as exceeding internal resource limits.

Path statistics

The -S flag, along with the optional -d flag, causes the lsmpio command to display normal or detailed path statistics. The optional -l flag restricts the display to contain statistics for just one MPIO storage device. The statistics include how many times the MPIO disk path is selected for an I/O operation, how many errors occurred on the MPIO disk path, and how many times the MPIO disk path failed. The detailed statistics information breaks down the failure counts, into counts of different types of failures.

Device inquiry data


The `-q` flag of the `lsmPIO` command causes the AIX MPIO to query the device, by using Small Computer System Interface (SCSI) commands to retrieve and display information about the attached device. Because each queried device is opened and queried by using SCSI commands, this operation might take time to run for many devices.

Parent adapter information

The `-a` flag, along with the optional `-r` flag, causes the `lsmPIO` command to display information about the Fibre Channel adapters that are used by the AIX MPIO storage devices. The information includes details about the local adapter identifier, such as the worldwide name of Fibre Channel adapters and the current state of the MPIO disk path, if available. The `-r` flag adds information about remote ports, which are accessed by the Fibre Channel adapter.

You can use the `-z` flag to reset all statistics to zero.

Flags

Item	Description
<code>-a</code>	Lists parent Fibre Channel adapter information.
<code>-d</code>	Displays detailed statistics. This flag is only valid with the <code>-S</code> flag.
<code>-e</code>	When this flag is used with the <code>-a</code> flag, error counts for the local adapter ports are displayed. When this flag is used with the <code>-a</code> and <code>-r</code> flags, error counts for local adapter ports and remote ports are displayed. These error counts indicate an issue with the physical connectivity between the local adapter and the remote storage port. The error counts indicate recent errors for different time ranges.
<code>> -f <</code>	<code>> </code> When the <code>-f</code> flag is used with the <code>-a</code> flag, the Fabric Performance Impact Notification (<code>fpin</code>) error counts for the local adapter ports are displayed. When the <code>-f</code> flag is used with the <code>-a</code> and <code>-r</code> flags, <code>fpin</code> error counts for both the local adapter ports and remote storage ports are displayed. These <code>fpin</code> error counts indicate that either the local adapter ports or the remote storage ports are congested. For different predefined time ranges that are displayed in the output (for example, last 10 minutes, last 60 minutes, and last 24 hours), the <code>fpin</code> error counts indicate recent errors. <code> <</code>
<code>-h</code>	Displays command usage information.
<code>-l disk_name</code>	Specifies a device. If this flag is included, the command operates on a single device. If this flag is omitted, the command operates on all AIX MPIO devices. This flag can be used by itself for the summary path status, or with the <code>-q</code> , <code>-S</code> , or <code>-z</code> flags.
<code>-o</code>	Indicates that the AIX disk driver attempts to access all multipath I/O (MPIO) that are associated with the specified MPIO disks. The attempt includes accessing the MPIO paths that were marked as failed when the MPIO disk was last closed. This flag provides an up-to-date status about the MPIO path.  Attention: If you attempt to access failed MPIO disk paths, the response time for the <code>lsmPIO</code> command can be slower. The response time can be delayed by a few seconds or several minutes depending on several factors such as the number of MPIO disks that are impacted and the scope of the <code>lsmPIO</code> command.
<code>-q</code>	Queries the device information. This command uses standard SCSI commands to query the device for information. The precise information that is returned varies, depending on the device type.

Item	Description
-r	Displays the remote port information. This flag is used along with the -a flag to display information about remote ports that are accessed by an adapter. The information that is returned might depend on the network protocol that is used by the adapter.
-S	Displays statistics for one or all devices. This flag displays basic counters for path use and path errors. If the -d flag is used along with this flag, it displays more detailed statistics.
-z	Resets all statistics. If this flag is used, it causes the PCM to reset all statistical counters back to zero.

Note: You can use the **-z** flag with the **-a** flag to reset the adapter error counts and **fpin** error counts.

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 the summary information for the **hdisk1234** device, enter the following command:

```
lsmpio -l hdisk1234
```

The system displays an output similar to one of the following outputs:

```

• name path_id status path_status parent connection
=====
hdisk1234 0 Enabled Opt,Rsv fscsi0 500a098186a7d4ca,0008000000000000
hdisk1234 1 Enabled Non fscsi0 500a098196a7d4ca,0008000000000000
hdisk1234 2 Enabled Opt,Sel,EEH fscsi1 500a098186a7d4ca,0008000000000000
hdisk1234 3 Enabled Non fscsi1 500a098196a7d4ca,0008000000000000

```

In this example, path 0 is optimized but is marked as reserved (**Rsv**) that makes the path unusable. In this case, path 2 is selected for the I/O operations because the **EEH** state is preferred than the unusable **Rsv** state. Moreover, the path 0 might never be in the **Deg** state because it is already being avoided due to the **Rsv** state. The I/O requests might not be fulfilled because the adapter that is used by path 2 is rebooting because of the **EEH** error.

```

• name path_id status path_status parent connection
=====
hdisk1234 0 Enabled Opt,Rsv fscsi0 500a098186a7d4ca,0008000000000000
hdisk1234 1 Enabled Non fscsi0 500a098196a7d4ca,0008000000000000
hdisk1234 2 Enabled Opt,EEH fscsi1 500a098186a7d4ca,0008000000000000
hdisk1234 3 Enabled Non fscsi1 500a098196a7d4ca,0008000000000000
hdisk1234 4 Enabled Opt,Sel,Deg fscsi2 500a098186b7d4ca,0008000000000000
hdisk1234 5 Enabled Non fscsi2 500a098196b7d4ca,0008000000000000

```

In this example, path 4 with the **Deg** state is preferred over path 2 with the **EEH** state. If you run **iostat -m hdisk1234**, the **iostat** command output shows that all the I/O operations are performed at path 4.

```

• name path_id status path_status parent connection
=====
hdisk1234 0 Enabled Opt,Rsv fscsi0 500a098186a7d4ca,0008000000000000
hdisk1234 1 Enabled Non fscsi0 500a098196a7d4ca,0008000000000000
hdisk1234 2 Enabled Opt,Sel fscsi1 500a098186a7d4ca,0008000000000000
hdisk1234 3 Enabled Non fscsi1 500a098196a7d4ca,0008000000000000
hdisk1234 4 Enabled Opt,Sel fscsi2 500a098186b7d4ca,0008000000000000
hdisk1234 5 Enabled Non fscsi2 500a098196b7d4ca,0008000000000000

```

In this example output, none of the paths have **Deg** or **EEH** state. In this case, path 2 or path 4 is selected based on shortest queue or round-robin algorithms.

- | name | path_id | status | path_status | parent connection |
|-----------|---------|---------|-------------|--|
| hdisk1234 | 0 | Enabled | Opt,Rsv | fscsi0 500a098186a7d4ca,0008000000000000 |
| hdisk1234 | 1 | Enabled | Non | fscsi0 500a098196a7d4ca,0008000000000000 |
| hdisk1234 | 2 | Enabled | Opt,Sel | fscsi1 500a098186a7d4ca,0008000000000000 |
| hdisk1234 | 3 | Enabled | Non | fscsi1 500a098196a7d4ca,0008000000000000 |
| hdisk1234 | 4 | Enabled | Opt | fscsi2 500a098186b7d4ca,0008000000000000 |
| hdisk1234 | 5 | Enabled | Non | fscsi2 500a098196b7d4ca,0008000000000000 |

In this example, path 2 is selected. The device is using the `fail_over` algorithm. If path 2 fails, the I/O request fails over to path 4.

- To display detailed device statistics for the `hdisk10` device, enter the following command:

```
lsmpio -Sd1 hdisk10
```

The system displays an output similar to the following output:

```
Disk: hdisk10
Path statistics since Tue May 21 17:38:43 CDT 2013
Path 0: (fscsi0:500a098186a7d4ca,8000000000000000)
  Path Selections:                                0
  Adapter Errors:                                  0
    Software:                                       0
    Hardware:                                       0
    Transport Dead:                                 0
    Transport Busy:                                 0
    Transport Fault:                                0
    No Device Response:                             0
    Target Port ID Changed:                         0
    Enhanced Error Handling (EEH):                  0
  Command Timeouts:                                0
  Reservation Conflicts:                           0
  SCSI Queue Full:                                  0
  SCSI Busy:                                         0
  SCSI ACA Active:                                  0
  SCSI Task Aborted:                                0
  SCSI Aborted Command:                             0
  SCSI Check Condition:                             0
    Medium Error:                                   0
    Hardware Error:                                 0
    Not Ready:                                       0
    Other:                                           0
  Last Error:                                       N/A
  Last Error Time:                                  N/A
  Path Failure Count:                                0
    Due to Adapter Error:                           0
    Due to I/O Error:                                0
    Due to Health Check:                             0
    Due to SCSI Sense:                               0
    Due to Qualifier Bit:                             0
    Due to Opening Error:                             0
    Due to PG SN Mismatch:                           0
  Last Path Failure:                                N/A
  Last Path Failure Time:                           N/A
```

Note: If some SCSI error counts are reported, it does not indicate a problem or indicate that I/O operations failed. During regular processing, temporary, recoverable errors might be reported frequently. These errors might cause the I/O operation to be attempted again.

- To display MPIO adapter information with the remote port information, enter the following command:

```
lsmpio -ar
```

The system displays an output similar to the following output:

```
Adapter Driver: fscsi0 - AIX PCM
Adapter WWPN: 10000000c94c7bd6
Link State: Up
Remote Ports          Paths          Paths          Paths          Paths          ID
500a098186a7d4ca     Enabled        Disabled        Failed         Missing
                    31            0              0             0             0x20a00
```

```

500a098196a7d4ca      31      0      0      0      0x20b00
500507630a18016b     19      0      0      0      0x31200
500507630a18416b     19      0      0      0      0x31300
500507630a18816b     19      0      0      0      0x31400
500507630a18c16b     19      0      0      0      0x31500

```

```

Adapter Driver: fscsi1 - AIX PCM
Adapter WWPN: 10000000c94c7bd7
Link State: Up

```

Remote Ports	Paths Enabled	Paths Disabled	Paths Failed	Paths Missing	ID
500a098186a7d4ca	31	0	0	0	0x20a00
500a098196a7d4ca	31	0	0	0	0x20b00
500507630a18016b	19	0	0	0	0x31200
500507630a18416b	19	0	0	0	0x31300
500507630a18816b	19	0	0	0	0x31400
500507630a18c16b	19	0	0	0	0x31500
5001738000330150	1	0	0	0	0x10100
5001738000330162	1	0	0	0	0x10200

4. To query an MPIO storage device and display information about it, enter the following command:

```
lsmPIO -ql hdisk48
```

The system displays an output similar to the following output:

```

Device: hdisk48
Vendor Id: IBM
Product Id: 2107900
Revision: .160
Capacity: 10G
Volume Serial: 600507630AFFC16B0000000000001505 (Page 83 NAA)

```

Note: The output that is displayed is derived from the standard inquiry data and the device identification vital product data (VPD). If the device represents a Peer-to-Peer Remote Copy (PPRC) pair (the **san_rep_device** attribute has a value of yes), the display includes the volume serial number for each logical unit number (LUN) in the PPRC pair and the vendor-specific ID that is shared by the two LUNs of the PPRC pair, as shown in the following output:

```

Device: hdisk33
Vendor Id: IBM
Product Id: 2107900
Revision: .160
Capacity: 10G
Volume Serial: 600507630AFFC16B0000000000000113 (Page 83 NAA)
Volume Serial: 600507630AFFC16B000000000000031F (Page 83 NAA)
Vendor LUN Id: 3735544C37373130313133005022AD6A

```

5. To display local and remote port error counts, enter the following command:

```
lsmPIO -are
```

The output might be similar to the following sample:

```

Adapter Driver: fscsi3 -> AIX PCM
Adapter WWPN: 21000024ff6aee7d
Link State: Up
Connectivity Errors:
  Last 10 Minutes: 74
  Last 60 Minutes: 222
  Last 24 Hours: 12345

```

Remote Ports	Connectivity Errors		
	Last 10 Minutes	Last 60 Minutes	Last 24 Hours
5001738000330171	0	0	0
5001738000330173	0	0	0
500a098286a7d4ca	2	9	45
500a098196a7d4ca	72	213	12300

6. To display local and remote port fpin error counts, enter the following command:

```
lsmPIO -arf
```

The output might be similar to the following sample:

```
# lsmPIO -arf
Adapter Driver: fscsi1 -> AIX PCM
Adapter WWPN: 100000109b1708ae
FPIN STATE: CLEAR
FPIN Errors
Last 10 Minutes: 0
Last 60 Minutes: 0
Last 24 Hours: 0
Total Adapter Errs: 1

          FPIN Errors
          Last 10 Last 60 Last 24 Total Peer FPIN STATE
          Minutes Minutes Hours Errs
5005076802164af3 0 0 0 4 CLEAR
5005076802364af4 0 0 0 0 CLEAR
5005076802464af4 0 0 0 0 CLEAR
5005076802264af3 1 1 1 2 PCn

Adapter Driver: fscsi0 -> AIX PCM
Adapter WWPN: 100000109b1708ad
FPIN STATE: LCn,LI
FPIN Errors
Last 10 Minutes: 1
Last 60 Minutes: 1
Last 24 Hours: 1
Total Adapter Errs: 2

          FPIN Errors
          Last 10 Last 60 Last 24 Total Peer FPIN STATE
          Minutes Minutes Hours Errs
5005076802164af3 0 0 0 4 LI
5005076802364af4 0 0 0 0 CLEAR
5005076802464af4 0 0 0 0 CLEAR
5005076802264af3 1 1 1 2 PCn
```

lsnamsv Command

Purpose

Shows name service information stored in the database.

Syntax

```
lsnamsv { -C | -S"AttributeList ..." } [ -Z ]
```

Description

The **lsnamsv** high-level command shows customized, TCP/IP-based name service information from the **/etc/resolv.conf** file only. No information from the name server database is shown. The command can extract all customized name service information or selected name service attribute information from the configuration database.

You can use the System Management Interface Tool (SMIT) **smit lsnamerslv** fast path to run this command.

Flags

Item	Description
-C	Extracts all customized name service configuration information.

Item	Description
-S " <i>AttributeList ...</i> "	Specifies a selected set of attributes to be extracted from the system configuration database. Attributes can be the following: domain Domain name nameserver Internet address of name server in dotted decimal format
-Z	Specifies that the output be in colon format. This flag is used when the lsnamsv command is invoked from the SMIT usability interface.

Examples

1. To list all customized name service configuration information in dotted decimal format, enter the following command:

```
lsnamsv -C
```

2. To list selected attributes, enter the following command:

```
lsnamsv -S "domain nameserver"
```

The **-S** flag indicates that the quoted list that follows contains a list of attributes to display.

lsnfsexp Command

Purpose

Displays the characteristics of directories that are exported with the Network File System (NFS).

Syntax

```
/usr/sbin/lsnfsexp [ -c | -l ] [ Directory ] [ -V Exported Version ] [ -f Exports_file ]
```

Description

The **lsnfsexp** command displays the characteristics of NFS-exported directories. The *Directory* parameter specifies the directory to be displayed. If no directory is specified, all directories exported with NFS will be displayed.

Flags

Item	Description
-c	Specifies that the output should be in colon format.
-l	(Lowercase L) Specifies that the output should be in list format. This flag is the default.
<i>Directory</i>	Specifies the directory to be displayed. If no directory is specified, all directories exported with NFS will be displayed.
-f <i>Exports_file</i>	Specifies the full path name of the export file to use if other than /etc/exports .

Item	Description
-V <i>Exported Version</i>	Specifies the version of the directory to be displayed. Valid version numbers are 2, 3 and 4.

Examples

1. To list all of the directories currently exported with NFS in the colon format, enter:

```
lsnfsexp -c
```

2. To list all of the directories currently exported with NFS in the colon format and use a specified path name other than **/etc/exports** enter:

```
lsnfsexp -c -f /etc/exports.other
```

3. To list the entry for the **/common/documents** directory that is exported as version 4, enter the following command:

```
lsnfsexp /common/documents -V 4
```

File

Item	Description
/etc/exports	Lists the directories the server can export.

lsnfsmnt Command

Purpose

Displays the characteristics of NFS mountable file systems.

Syntax

```
/usr/sbin/lsnfsmnt [ -c | -l | -p ] [ FileSystem ]
```

Description

The **lsnfsmnt** command displays the current characteristics of NFS mountable file systems. The *FileSystem* parameter specifies the file system to be displayed in the output. If no file system is specified, all of the file systems that are NFS mountable will be displayed.

Flags

Item	Description
-c	Specifies that the output should be in colon format.
-l	(Lowercase L) Specifies that the output should be in list format. This flag is the default.
-p	Specifies that the output should be in pipe format.

Examples

To list all of the NFS mounted file systems in the colon format, enter:

```
lsnfsmnt -c
```

Files

Item	Description
<code>/etc/filesystems</code>	Centralizes file system characteristics.

lsnim Command

Purpose

Displays information about the Network Installation Management (NIM) environment.

Syntax

To Display a List of Supported NIM Classes, Subclasses, or Types

```
lsnim { -p | -P } [ -cClass | -S ]
```

To Display Predefined NIM Information

```
lsnim { -p | -P } [ -cClass | -sSubclass | -tType ] [ -l | [ -o ] | -O ] [ -Z ]
```

OR

```
lsnim { -p | -P } [ -a Attribute ] . . . [ -Z ]
```

To Display Attributes Required for an Operation

```
lsnim -tType -qOperation
```

To Display Information about All Customized NIM Objects

```
lsnim [ -cClass | -sSubclass | -tType ] [ -l | [ -o ] | -O ] [ -Z ]
```

OR

```
lsnim [ -aAttribute ] . . . [ -Z ]
```

To Display Information about a Specific NIM Object

```
lsnim [ -l | -O | -a Attribute . . . ] [ -Z ] ObjectName
```

OR

```
lsnim [ -qOperation ] ObjectName
```

To Display Information about Resources Available to a Specific NIM Machine

```
lsnim -L [ -sSubclass | -tType ] ObjectName
```

To Display Information about NIM Groups

```
lsnim -g | -m [ -a Attribute | -cClass | -L | -l | -sSubclass | -tType ] GroupObjectName
```

Description

The **lsnim** command displays information about the NIM environment. This information is divided into two basic categories: predefined and customized.

Predefined information consists of values that are preset by NIM and cannot be modified by the user. Examples of predefined information include:

- The types of objects supported by NIM
- The classes and subclasses into which NIM organizes objects

- The operations that can be performed on NIM objects
- The attributes that can be entered by the user

In general, NIM uses this information to make decisions during operations. Predefined information can be displayed by using the **-p** or **-P** flag. The **-p** flag displays default values while the **-P** flag displays help information.

Customized information consists of values that you enter or modify. This information represents the physical environment in which NIM operates. Related pieces of customized information are grouped together to form *objects*, which are organized in the NIM database by object type and class. Some examples of object types include *diskless*, *paging*, and *standalone*. Two examples of object classes are *machines* and *network*.

For example, a standalone workstation that is part of the NIM environment is represented by a unique object. This object is classified by NIM as a *standalonemachines* object, where *standalone* represents the object type and *machines* represents the object class. Entering the **lsnim** command on the command line without any flags displays information on all customized objects.

You can also use the **lsnim** command to display relationships between customized objects. Choose an object to *anchor* on (specified by the *Objectname* parameter) and then select the desired relationship with the **-c**, **-s**, or **-t** flag. The information displayed then depends upon the type and class of the anchored object. For example, if you select an object of type **spot**, the type of relationships that can be displayed are:

- Machines that use the Shared Product Object Tree (SPOT) resource.
- Networks that can access the SPOT resource.

When not displaying relationships, the **lsnim** command provides flags that can be used to filter the output that it would normally display. The **-a**, **-c**, **-O**, **-s**, or **-t** flag can be used to restrict the amount of information which is displayed.

Flags

Item	Description
-a <i>Attribute</i>	Filters displayed information based on the specified attribute name. The possible attributes are: Operation subclass type class
-c <i>Class</i>	Specifies a NIM object class. When this flag is used without the <i>Objectname</i> parameter, it filters the displayed information so only information about objects in that class is displayed.
-g	Displays long listing of group object with state information for individual members.
-l	Displays detailed information.
-L	Displays information about resources that can be accessed by a client machine.
-m	Applies other flags specified to group members.
-o	Is used by the SMIT interface of a NIM environment.
-O	Lists the operations NIM supports.
-p	Displays predefined information using default values.
-P	Displays help information for predefined data.
-q <i>Operation</i>	Lists the attributes required for the specified operation.

Item	Description
-S	Displays a list of NIM subclasses.
-s Subclass	Specifies a NIM subclass. When this flag is used without the <i>ObjectName</i> parameter, it filters the displayed information so only information about objects in that subclass is displayed.
-t Type	Specifies a NIM object type. When this flag is used without the <i>Objectname</i> parameter, it filters the displayed information so only information about objects of that type is displayed.
-Z	Displays information in colon-separated format.

Security

Access Control: You must have root authority to run the **lsnim** command.

Examples

1. To display a list of NIM object classes, enter:

```
lsnim -p
```

2. To display a list of NIM subclasses, enter:

```
lsnim -p -S
```

3. To display the list of NIM object types for the `machines` object class, enter:

```
lsnim -p -c machines
```

4. To display help information about NIM object types for the `machines` object class, enter:

```
lsnim -P -c machines
```

5. To display detailed information about the NIM attributes named `lpp_source` and `Rstate`, enter:

```
lsnim -p -a lpp_source -a Rstate
```

6. To display the operations which can be performed on the `paging` object type, enter:

```
lsnim -p -t paging -0
```

7. To display the information required to perform a `bos_inst` operation on an object of the `standalone` object type, enter:

```
lsnim -t standalone -q bos_inst
```

8. To display information about all customized objects of the `diskless` object type, enter:

```
lsnim -t diskless
```

9. To display all customized objects in the `networks` object class, enter:

```
lsnim -c networks
```

10. To display detailed information about a NIM object named `altoid`, enter:

```
lsnim -l altoid
```

11. To display the relationship between an object named `altoid` and all NIM resources, enter:

```
lsnim -c resources altoid
```

12. To display a list of operations that can be applied to altoid, enter:

```
lsnim -O altoid
```

13. To display a list of resources available to altoid, enter:

```
lsnim -L altoid
```

14. To display the members of the machine group MacGrp1 with state and group exclusion status, enter:

```
lsnim -g MacGrp1
```

15. To display basic information about the members of the resource group ResGrp1, enter:

```
lsnim -m ResGrp1
```

16. To display a long listing of members of the machine group MacGrp1, with any hidden NIM internal information, enter:

```
lsnim -m -Fl MacGrp1
```

17. To display all members of machine group MacGrp1 which has a spot allocated, enter:

```
lsnim -ma spot MacGrp1
```

Files

Item	Description
/etc/niminfo	Contains variables used by NIM.

lsnlspath Command

Purpose

Shows the value of the secure NLSPATH system configuration variable.

Syntax

```
lsnlspath
```

Description

The **lsnlspath** command outputs the current value of the secure NLSPATH variable.

lsparent Command

Purpose

Displays the possible parent devices that accept a specified connection type or device.

Syntax

```
lsparent { -C | -P } { -k ChildConnectionKey | -l ChildName } [ -f File ] [ -F Format ] [ -h ] [ -H ]
```

Description

The **lsparent** command lists devices from the Device Configuration database that can accept a given device as a child device, specified by the **-l** *ChildName* flag, or a given type of child device connection, specified by the **-k** *ChildConnectionKey* flag.

You can display the default output one of the following ways.

- Use the **-C** flag to display the default output information for a device from the Customized Devices object class, which is name, state, location, and description.
- Use the **-P** flag to display the default output information for a device from the Predefined Devices object class, which is class, type, subclass, and description.

To override these two default outputs, you can use the **-F** *Format* flag to display the output as designated by a user-formatted string. The *Format* parameter is a quoted list of column names separated and possibly terminated by nonalphanumeric characters.

You can supply the flags either on the command line or from the specified *File* parameter using the **-f** flag.

Flags

Item	Description
-C	Lists information about a device that is in the Customized Devices object class. The information displayed can be from both the Customized and Predefined Devices object classes. This flag cannot be used with the -P flag.
-f <i>File</i>	Reads the necessary flags from the <i>File</i> variable.
-F <i>Format</i>	Displays the output in a user-specified format, where the <i>Format</i> variable is a quoted list of column names from the Predefined Devices object class or the Customized Devices object class separated and possibly terminated by non-alphanumeric characters. If white space is used as the separator, the lsparent command displays the output in aligned columns. In addition to the column names in the two object classes, the special name <i>description</i> can be used to display a text description of the device.
-H	Displays headers above the column output.
-h	Displays the command usage message.
-k <i>ChildConnectionKey</i>	Specifies the connection key that identifies the device subclass name of the child device. This flag cannot be used with the -l flag.
-l <i>ChildName</i>	Specifies the logical name of a possible child device. This flag cannot be used with the -k flag.
-P	Lists information about a device that is in the Predefined Devices object class. The information displayed can be from both the Customized and Predefined Devices object classes. This flag cannot be used with the -C flag.

Examples

1. To list possible parent devices in the Customized Devices object class that accept an RS-232 device, type the following:

```
lsparent -C -k rs232
```

The system displays a message similar to the following:


```
sa0 Available 01-S1 Standard I/O Serial Port
sa1 Available 01-S2 Standard I/O Serial Port
sa2 Available 10-68 IBM 8-Port EIA-232/RS-422A (PCI) Adapter
sa3 Available 10-70 IBM 8-Port EIA-232/RS-422A (PCI) Adapter
sa4 Available 10-78 IBM 8-Port EIA-232/RS-422A (PCI) Adapter
sa5 Available 20-58 IBM 8-Port EIA-232/RS-422A (PCI) Adapter
```

- To list possible types of parent devices in the Predefined Devices object class that accept an RS-232 device, type the following:

```
lsparent -P -k rs232
```

The system displays a message similar to the following:

```
adapter      pnp501   isa_sio   Standard I/O Serial Port
adapter      4f111100 pci       IBM 8-Port EIA-232/RS-422A (PCI) Adapter
concentrator 16c232   sync_pci 16-Port RAN EIA-232 for 128-Port Adapter
concentrator 16e232   sync_pci 16-Port Enhanced RAN EIA-232 for 128-Port Adapter
```

- To list possible parent devices in the Customized Devices object class that accept the `rmt0` tape device as a child device, type the following:

```
lsparent -C -l rmt0
```

The system displays a message similar to the following:

```
scsi2 Available 20-60 Wide/Ultra-2 SCSI I/O Controller
scsi3 Available 20-61 Wide/Ultra-2 SCSI I/O Controller
scsi1 Available 10-88 Wide/Ultra-2 SCSI I/O Controller
scsi0 Available 10-60 Wide/Fast-20 SCSI I/O Controller
```

- To list possible types of parent devices in the Predefined Devices object class that accept the `rmt0` tape device as a child device, type the following:

```
lsparent -P -l rmt0
```

The system displays a message similar to the following:

```
adapter sym896 pci Wide/Ultra-2 SCSI I/O Controller
adapter sym895 pci Wide/Ultra-2 SCSI I/O Controller
adapter sym875 pci Wide/Fast-20 SCSI I/O Controller
```

Files

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

lspath Command

Purpose

Displays information about paths to an MultiPath I/O (MPIO) capable device.

Syntax

```
lspath [ -F Format | -t ] [ -H ] [ -l Name ] [ -p Parent ] [ -s Status ] [ -w Connection ] [ -i PathID ]
```

```
lspath -A -l Name -p Parent [ -w Connection ] [ -i PathID ] { -D [ -O ] | -E [ -O ] | -F Format [ -Z character ] } [ -a Attribute ] ... [ -f File ] [ -H ]
```

```
lspath -A -l Name -p Parent [ -w Connection ] [ -i PathID ] -R -a Attribute [ -f File ] [ -H ]
```

```
lspath -h
```

Description

The **lspath** command displays one of two types of information about paths to an MPIO capable device. It either displays the operational status for one or more paths to a single device, or it displays one or more attributes for a single path to a single MPIO capable device. The first syntax shown above displays the operational status for one or more paths to a particular MPIO capable device. In this instance, the **lspath** command is similar to the **lsdev** command. The second syntax (keyed by the presence of the **-A** flag) displays one or more attributes for a single path to a particular MPIO capable device. In this instance, the **lspath** command is similar to the **lsattr** command. In fact, all of the flags for the **lsattr** command are supported on the **lspath** command when displaying path attributes.

Displaying Path Status with the lspath Command

When displaying path status, the set of paths to display is obtained by searching the device configuration database for paths that match the following criteria:

- The target device name matches the device specified with the **-l** flag. If the **-l** flag is not present, then the target device is not used in the criteria.
- The parent device name matches the device specified with the **-p** flag. If the **-p** flag is not present, then parent is not used in the criteria.
- The connection matches the connection specified with the **-w** flag. If the **-w** flag is not present, then connection is not used in the criteria.
- The path status matches status specified with the **-s** flag. If the **-s** flag is not present, the path status is not used in the criteria.

If none of the **-l**, **-p**, **-w**, **-s** flags are specified, then all paths known to the system are displayed.

By default, this command will display the information in columnar form. When no flags are specified that qualify the paths to display, the format of the output is:

```
status device parent
```

The default display format can be overridden by using the **-F Format** flag. The **-F Format** flag displays the output in a user-specified format where the *Format* parameter is a quoted list of field names separated by, and possibly ended by, non-alphanumeric characters or white space. The field names are the fields defined in the **CuPath** class or one of the column heading defined above.

Note: The column names above are not translated into other languages (either when output as column headings or when input as part of the *Format* of the **-F** flag).

Possible values that can appear for the status column are:

enabled

Indicates that the path is configured and operational. It will be considered when paths are selected for IO.

Note: The AIX MPIO does not fail the last path. Even though all paths to the storage are lost, the last path displays the status as **enabled**. In such a case, an IO error indicates the actual loss of the last path to the disk.

disabled

Indicates that the path is configured, but not currently operational. It has been manually disabled and will not be considered when paths are selected for IO.

failed

Indicates that the path is configured, but it has had IO failures that have rendered it unusable. It will not be considered when paths are selected for IO.

defined

Indicates that the path has not been configured into the device driver.

missing

Indicates that the path was defined in a previous boot, but it was not detected in the most recent boot of the system.

detected

Indicates that the path was detected in the most recent boot of the system, but for some reason it was not configured. A path should only have this status during boot and so this status should never appear as a result of the **lspath** command.

Displaying Path Attributes with the lspath Command

When displaying attributes for a path, the path must be fully qualified. Multiple attributes for a path can be displayed, but attributes belonging to multiple paths cannot be displayed in a single invocation of the **lspath** command. Therefore, in addition to the **-A** flag, the **-l**, **-p**, or **-w** flags are required to uniquely identify a single path. For example:

- if only one path exists to a device, the **-l** flag is required
- if only one path between a device and a specific parent, the **-l** and **-p** flags are required
- if there are multiple paths between a device and a specific parent, the **-l**, **-p**, and **-w** flags are required

Furthermore, the **-s** flag is not allowed.

The same rules used by the **lsattr** command for displaying device attributes applies to the **lspath** command for displaying path attributes.

By default, this command will display the information in columnar form. The format of the output is the same as the **lsattr** command:

attribute	value	description	user_settable
-----------	-------	-------------	---------------

All fields are shown by default. The default display format can be overridden by using the **-F Format** flag. The **-F Format** flag displays the output in a user-specified format where the *Format* parameter is a quoted list of column names separated by, and possibly ended by, non-alphanumeric characters or white space. The column names allowed are the field names from the **CuPathAt**, **PdPathAt**, and **PdAtXtd** object classes plus the columns listed above.

Note: The column names above are not translated into other languages (either when output as column headings or when input as part of the *Format* of the **-F** flag).

Flags

Item	Description
-a <i>Attribute</i>	Identifies the specific attribute to list. The ' <i>Attribute</i> ' is the name of a path specific attribute. When this flag is provided, only the identified attribute is displayed. Multiple instances of this flag may be used to list multiple attributes. If this flag is not specified at all, all attributes associated with the identified path will be listed.
-A	Indicates that attributes for a specific path are to be displayed. When the -A flag is present, the -s <i>Status</i> flag is not allowed. However, the -l <i>Name</i> , -p <i>Parent</i> , and -w <i>Connection</i> flags must be present to fully qualify the path.
-D	Displays the attribute names, default values, descriptions, and user-settable flag values for a specific path when not used with the -O flag. The -D flag displays only the attribute name and default value in colon format when used with the -O flag. This flag is only valid when displaying path attributes and it cannot be used with the -E , -F , or -R flag.
-E	Displays the attribute names, current values, descriptions, and user-settable flag values for a specific path when not used with the -O flag. The -E flag displays only the attribute name and current value in colon format when used with the -O flag. This flag is only valid when displaying path attributes and it cannot be used with the -D , -F , or -R flag.
-f <i>File</i>	Reads the needed flags from the <i>File</i> parameter.

Item	Description
-F <i>Format</i>	Displays the output in a user-specified format, where the <i>Format</i> parameter is a quoted list of column names separated by non-alphanumeric characters or white space. Using white space as the separator, the <code>lspath</code> command displays the output in aligned columns. Valid column names depends upon the type of information requested. For path display, column names from the CuPath object class can be specified. For path attribute display (the -A flag is specified), column names from the PdPathAt and CuPathAt object classes can be specified. In addition to the column names, there are two special purpose names that can be used. The <code>name</code> description can be used to obtain a display of attribute descriptions and <code>user-settable</code> can be used to obtain an indication as to whether or not an attribute can be changed. This flag cannot be used with the -E , -D , -O or -R flag.
-h	Displays the command usage message.
-H	Displays headers above the column output. To use the -H flag with the -O flag is meaningless, the -O flag prevails. To use the -H flag with the -R flag is meaningless; the -R flag prevails.
-i <i>PathID</i>	Indicates the path ID associated with the path to be displayed.
-l <i>Name</i>	Specifies the logical device name of the target device whose path information is to be displayed. This flag is optional for displaying path status, but is required for displaying path attributes.
-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 is only valid when displaying path attributes and it cannot be used with the -F and -R flags.
-p <i>Parent</i>	Indicates the logical device name of the parent device, whose paths are to be displayed. This flag is optional for displaying path status, but it is required for displaying path attributes.
-R	Displays the legal values for an attribute name. The -R flag cannot be used with the -D , -E , -F and -O flags. The -R flag displays the list attribute values in a vertical column as follows:

```
Value1
Value2
.
.
ValueN
```

The **-R** flag displays the range attribute values as `x...n(+i)` where `x` is the start of the range, `n` is the end of the range, and `i` is the increment.

Item	Description
-s Status	<p>The -s Status flag indicates the status to use in qualifying the paths to be displayed. When displaying path information, the allowable values for this flag are:</p> <p>enabled Display paths that are enabled for MPIO path selection.</p> <p>disabled Display paths that are disabled from MPIO path selection.</p> <p>failed Display paths that are failed due to IO errors.</p> <p>available Display paths whose path_status is PATH_AVAILABLE (that is, paths that are configured in the system, includes enabled, disabled, and failed paths).</p> <p>defined Display paths whose path_status is PATH_DEFINED.</p> <p>missing Display paths whose path_status is PATH_MISSING.</p> <p>You can use alternative forms of the values. Strings beginning with 0, d, or D are treated in the same way as defined. The only exception is that strings starting with di, Di, dI, or DI are treated in the same way as disabled. Strings beginning with 1, a, or A are treated in the same way as available. Strings beginning with 3, m, or M are treated in the same way as missing. Strings beginning with e or E are treated in the same way as enabled. Strings beginning with f or F are treated in the same way as failed.</p>
-t	Displays the path ID in addition to the current default output. The -t flag cannot be used with the -F or the -A flags.
-w Connection	Indicates the connection information to use in qualifying the paths to be displayed. This flag is optional for displaying path status, but is required for displaying path attributes.
-Z Character	The -Z Character flag is meant to be used for programs that need to deal with ODM fields that may have embedded new line characters. The -Z Character flag is used to change the record separator character for each record (line) of output generated. The new record separator is the ' <i>Character</i> ' argument to this flag. The -Z Character flag is only relevant when the -A and the -F Format flags are specified. The -Z Character flag cannot be used with the -D , -E , -O , or the -R flags.

Security

Privilege Control: All users can execute this command.

Auditing Events: None.

Examples

Examples of displaying path status:

1. To display the status of all paths to hdisk1 with column headers, enter the following command:

```
lspath -H -l hdisk1
```

The system will display a message similar to the following:

```
status  device  parent
enabled hdisk1  scsi0
disabled hdisk1  scsi1
missing hdisk1  scsi2
```

2. To display, without column headers, the set of paths whose operational status is disabled, enter the following command:

```
lspath -s disabled
```

The system will display a message similar to the following:

```
disabled hdisk1 scsi1
disabled hdisk2 scsi1
disabled hdisk23 scsi8
disabled hdisk25 scsi8
```

3. To display the set of paths whose operational status is failed, enter the following command:

```
lspath -s failed
```

The system will display a message similar to the following:

```
failed hdisk1 scsi1
failed hdisk2 scsi1
failed hdisk23 scsi8
failed hdisk25 scsi8
```

4. To display in a user-specified format, without column headers, the set of paths to hdisk1 whose path status is available enter the following command:

```
lspath -l hdisk1 -s available -F"connection:parent:path_status:status"
```

The system will display a message similar to the following:

```
5,0:scsi0:available:enabled
6,0:scsi1:available:disabled
```

Note that this output shows both the path status and the operational status of the device. The path status simply indicates whether the path is configured or not. The operational status indicates how the path is being used with respect to path selection processing in the device driver. Only paths with a path status of available also have an operational status. If a path is not currently configured into the device driver, it does not have an operational status.

Examples of displaying path attributes:

1. If the target device is a SCSI disk, to display all attributes for the path to parent scsi0 at connection 5,0, enter the following command:

```
lspath -AHE -l hdisk10 -p scsi0 -w "5,0"
```

The system will display a message similar to the following:

attribute	value	description	user_settable
weight	1	Order of path failover selection	true

lspriv Command

Purpose

Displays the privileges available on the system.

Syntax

```
lspriv [ -v ]
```

Description

The **lspriv** command displays privileges available to the system. If run within a workload partition (WPAR), the **lspriv** command displays only the privileges available to the partition. If the **-v** flag is specified, the **lspriv** command also displays privilege descriptions.

Flags

Item	Description
-v	Specifies to display the privilege description for each privilege.

Security

Any user can run the **lspriv** command on the system.

Examples

1. To display all privileges available on the system, use the following command:

```
lspriv
```

The system displays an output similar to the following example:

```
PV_ROOT
PV_AU
PV_AU_ADD
PV_AU_ADMIN
PV_AU_READ
...
```

2. To display all privileges available on the system and their textual description, use the following command:

```
lspriv -v
```

The system displays an output similar to the following example:

```
PV_ROOT Allows a process to pass any non-SU privilege check.
PV_AU Equivalent to all Auditing privileges (PV_AU_*) combined.
PV_AU_ADD Allows a process to record/add an audit record.
PV_AU_ADMIN Allows a process to configure and query the audit system.
PV_AU_READ Allows a process to read a file marked as an audit file.
...
```

lsprtsv Command

Purpose

Shows print service information stored in the database.

Syntax

```
lsprtsv { -c | -p } [ -h ] [ -q Entry ... ] [ -Z ]
```

Description

The **lsprtsv** high-level command shows predefined and customized TCP/IP-based print service information. Use the **lsprtsv** command to extract customized or predefined print service information.

The **lsprtsv** command can show the following information:

- A list of host names that have access rights to the print server

- Logical print queue information only

Flags

Item	Description
-c	Extracts customized configuration information.
-h	Shows a list of host names that can use the print server.
-p	Extracts predefined configuration information.
-q <i>QEntry...</i>	Shows the logical print queues specified and their attributes available on a host. The <i>QEntry</i> variable specifies the names of the queues to display.
-Z	Specifies that the output be produced in colon format. This flag is used if the lsprtsv command is invoked from the SMIT usability interface.

Examples

1. To show all host names who have access rights to a print server, enter:

```
$ lsprtsv -c -h
```

2. To show which logical printers are available on a given client machine, enter:

```
lsprtsv -c -q sahara
```

lsps Command

Purpose

Displays the characteristics of a paging space.

Syntax

```
lsps { -s | [ -c | -l ] { -a | -t { lv | nfs | ps_helper } | PagingSpace } }
```

Description

The **lsps** command displays the characteristics of a paging space. The **lsps** command displays characteristics such as the paging-space name, physical-volume name, volume-group name, size, percentage of the paging space used, whether the space is active or inactive, and whether the paging space is set to automatic. The *PagingSpace* parameter specifies the paging space whose characteristics are to be shown.

For NFS paging spaces, the physical-volume name and volume-group name will be replaced by the host name of the NFS server and the path name of the file that is used for paging.

If the **-t** flag is specified, the argument will be assumed to be a third-party helper executable. If the helper executable is present in the */sbin/helpers/pagespace* path then it will be spawned passing all the arguments and with the **-l** flag to specify the **lsps** command. The helper executable must take care of displaying the characteristics of the page space. If the helper program doesn't exist in */sbin/helpers/pagespace* directory, the **lsps** command will display the usage error. The helper executable must exit with a 0 if successful and a non-zero if it fails.

You can use the System Management Interface Tool (SMIT) **smit lsps** fast path to run this command.

Flags

Item Description

- a** Specifies that the characteristics of all paging spaces are to be given. The size is given in megabytes.
- c** Specifies that the output should be in colon format. The colon format gives the paging space size in logical partitions.
- l** Specifies that the output should be in list format.
- s** Specifies that the summary characteristics of all paging spaces are to be given. This information consists of the total paging space in megabytes and the percentage of paging space currently assigned (used). If the **-s** flag is specified, all other flags are ignored.

Note: There is a paging space limit of 64 GB per device.

Note: Setting the environment variable **PSALLOC=early** causes the use of early paging space algorithm. In this case, the value the **-s** flag specifies is different from the value returned for a single paging space or when using the **-a** flag for all the paging spaces. The value the **-s** flag displays is the percentage of paging space allocated (reserved), whether the paging space has been assigned (used) or not. Therefore, the percentage reported by the **-s** flag is usually larger than that reported by the **-a** flag when **PSALLOC** is set to early.

- t** Specifies the characteristics of the paging space. One of the following variables is required:

lv

Specifies that the characteristics of only logical volume paging spaces are to be given.

nfs

Specifies that the characteristics of only NFS paging spaces are to be given. The heading of the output will be changed to display the host name of the NFS server and the path name of the file that resides on the server that is being used for NFS paging.

ps_helper

Name of the helper program for a third party device.

Examples

1. To list the characteristics of all paging spaces, enter:

```
lsps -a
```

This displays the characteristics for all paging spaces and provides a listing similar to the following listing:

Page Space	PhysicalVolume	Volume Group	Size	%Used	Active	Auto	Type	Chksum
hd6	hdisk0	rootvg	512MB	1	yes	yes	lv	8

2. To display the characteristics of paging space myps using the helper program foo enter the following command:

```
lsps -t foo myps
```

This displays the characteristics for all paging spaces and provides a listing similar to the following listing:

Page Space	Physical Volume	Volume Group	Size	%Used	Active	Auto	Type
myps	mydisk	myvg	512MB	1	yes	yes	lv

Files

Item	Description
<code>/etc/swapspace</code> s	Specifies the paging space devices and their attributes.

lspv Command

Purpose

Displays information about a physical volume within a volume group.

Syntax

lspv

OR

lspv [**-L**] [**-P**] [**-l** | **-p** | **-M**] [**-n** *descriptorphysicalvolume*] [**-v** *volumegroupid*] *physicalvolume*

Description

The **lspv** command displays the information about the physical volume if the specific physical volume name is specified. If you do not add flags to the **lspv** command, by default all the available physical volumes are printed along with the following information:

- Physical disk name.
- Physical volume identifiers (PVIDs).
- The volume group, if any, that the physical volume belongs to or the label, if any, locked with the **lkdev** command.
- The state of the volume group.

Active

When the volume group is varied on.

Concurrent

When the volume group is varied on in the concurrent mode.

Locked

When the physical volume is locked with the **lkdev** command.

Note: If the **lspv** command cannot find the information for a field in the Device Configuration Database, it will insert a question mark (?) in the value field. As an example, if there is no information for the **PP RANGE** field, the following value might be displayed:

```
PP RANGE: ?
```

Note: The **lspv** command, without any flags, can display the General Parallel File System (GPFS) volume groups that are located on the disks. However, the **lspv** command must initially be run with a root authority so that the command has permissions to query the GPFS nodes for information. After the GPFS volume group names are cached locally, non-root users running the **lspv** command can see the GPFS volume group names.

The **lspv** command attempts to obtain as much information as possible from the description area when it is given a logical volume identifier.

When the *physicalvolume* parameter is used, the following characteristics of the specified physical volume are displayed:

Item	Description
<code>Physical volume</code>	The name of the physical volume.

Item	Description
Volume group	The name of volume group. Volume group names must be unique systemwide names and can be from 1 to 15 characters long.
PV Identifier	The physical volume identifier for this physical disk.
VG Identifier	The volume group identifier of which this physical disk is a member.
PVstate	The state of the physical volume. If the volume group that contains the physical volume is varied on with the varyonvg command, the state is <code>active</code> , <code>missing</code> , or <code>removed</code> . If the physical volume is varied off with the varyoffvg command, the state is <code>varied off</code> .
Allocatable	The allocation permission for this physical volume.
Logical volumes	The number of logical volumes using the physical volume.
Stale PPs	The number of physical partitions on the physical volume that are not current.
VG descriptors	The number of volume group descriptors on the physical volume.
PP size	The size of physical partitions on the volume.
Total PPs	The total number of physical partitions on the physical volume.
Free PPs	The number of free physical partitions on the physical volume.
Used PPs	The number of used physical partitions on the physical volume.
Max Request	The max transfer size of the physical volume.
Free distribution	The number of free partitions available in each intra-physical volume section.
Used distribution	The number of used partitions in each intra-physical volume section.
Mirror Pool	The mirror pool that the physical volume has been assigned to.

You can use the System Management Interface Tool (SMIT) **smit lspv** fast path to run this command.

Flags

Item	Description
-L	Specifies no waiting to obtain a lock on the Volume group. Note: If the volume group is being changed, using the -L flag gives unreliable date.

Item	Description
-l	<p>Lists the following fields for each logical volume on the physical volume:</p> <p>LVname Name of the logical volume to which the physical partitions are allocated.</p> <p>LPs The number of logical partitions within the logical volume that are contained on this physical volume.</p> <p>PPs The number of physical partitions within the logical volume that are contained on this physical volume.</p> <p>Distribution The number of physical partitions, belonging to the logical volume, that are allocated within each of the following sections of the physical volume: outer edge, outer middle, center, inner middle and inner edge of the physical volume.</p> <p>Mount Point File system mount point for the logical volume, if applicable.</p>
-M	<p>Lists the following fields for each logical volume on the physical volume:</p> <pre style="background-color: #f0f0f0; padding: 5px;">PVname:PPnum [LVname: LPnum [:Copynum] [PPstate]]</pre> <p>Where:</p> <p>PVname Name of the physical volume as specified by the system.</p> <p>PPnum Physical partition number.</p> <p>LVname Name of the logical volume to which the physical partitions are allocated. Logical volume names must be system-wide unique names, and can range from 1 to 64 characters.</p> <p>LPnum Logical partition number. Logical partition numbers can range from 1 to 64,000.</p> <p>Copynum Mirror number.</p> <p>PPstate Only the physical partitions on the physical volume that are not current are shown as stale.</p>
-n <i>descriptorphysicalvolume</i>	<p>Accesses information from the variable descriptor area specified by the <i>descriptorphysicalvolume</i> variable. The information may not be current, since the information accessed with the -n flag has not been validated for the logical volumes. If you do not use the -n flag, the descriptor area from the physical volume that holds the validated information is accessed, and therefore the information displayed is current. The volume group need not be active when you use this flag.</p>

Item	Description
-p	<p>Lists the following fields for each physical partition on the physical volume:</p> <p>Range A range of consecutive physical partitions contained on a single region of the physical volume.</p> <p>State The current state of the physical partitions: <code>free</code>, <code>used</code>, <code>stale</code>, or <code>vgda</code>.</p> <p>Note: If a <i>volume group</i> is converted to a big vg format, it may be necessary to use some data partitions for <i>volume group</i> descriptor area. These partitions will be marked <code>vgda</code>.</p> <p>Region The intra-physical volume region in which the partitions are located.</p> <p>LVname The name of the logical volume to which the physical partitions are allocated.</p> <p>Type The type of the logical volume to which the partitions are allocated.</p> <p>Mount point File system mount point for the logical volume, if applicable.</p>
-P	Lists the mirror pool that each physical volume belongs to.
-u	<p>Lists all the physical volumes in the system along with the following information:</p> <ul style="list-style-type: none"> • Physical disk name. • Physical volume identifiers (PVIDs). • The volume group (if any), or label (if any), that the physical volume belongs to and that is locked with the lkdev command. • The state of the volume group. <p>Active When the volume group is varied on.</p> <p>Concurrent When the volume group is varied on in the concurrent mode.</p> <p>Locked When the physical volume is locked with the lkdev command.</p> <ul style="list-style-type: none"> • Unique device identifier (UDID). • Universally Unique Identifier (UUID).
-v <i>volumegroupid</i>	Accesses information based on the <i>volumegroupid</i> variable. This flag is needed only when the <code>lspv</code> command does not function due to incorrect information in the Device Configuration Database. The <i>volumegroupid</i> variable is the hexadecimal representation of the volume group identifier, which is generated by the mkvg command.

Examples

1. To display the status and characteristics of physical volume `hdisk3`, enter the following command:

```
lspv hdisk3
```

2. To display the status and characteristics of physical volume `hdisk5` by physical partition number, enter the following command:

```
lspv -p hdisk5
```

Files

Item	Description
/usr/sbin	Contains the lspv command.

Ispprc Command

Purpose

Displays information about PPRC (Peer-to-Peer Remote Copy) disks.

Syntax

```
ispprc -A [-o ]
```

```
ispprc -c pprc_disk
```

```
ispprc [-h ]
```

```
ispprc -p pprc_disk
```

```
ispprc -v pprc_disk
```

Description

The **ispprc** command displays information that is related to PPRC disk like vital product data (VPD) information of the individual LUNs that are part of the PPRC disk, path group information of a PPRC disk, replication path information of a PPRC disk, and a list of all PPRC disks available on the system.

Flags

Item	Description
-A [-o]	Displays the information of all PPRC disks in the system, such as PPRC state and path groups IDs. The optional -o flag briefly opens all potential PPRC disks before displaying the status, ensuring that the status is current. Use of the -o flag might increase the length of time that is required to run the command, depending on the number of disks that are attached to the system.

Example:

hdisk# Storage	PPRC state	Primary path group ID	Secondary path group ID	Primary Storage WWNN	Secondary WWNN
hdisk4 500507630affc16b	Active	0(s)	1	500507630affc16b	
hdisk5 500507630affc16b	Active	0(s)	1	500507630affc16b	
hdisk6 500507630affc16b	Active	0(s)	1	500507630affc16b	

Note: For explanation of the path group IDs and the selected path group that is identified as s, see the description of the **-p** flag.

Item**Description****-c**

Displays information about the replication path connection, which is related to a PPRC disk. The output displays information about the replication paths between the two storage subsystems that contain the LUNs in the PPRC pair. The output displays the worldwide node names, Subsystem IDs (SSID), Logical Subsystems (LSS), and ports for the endpoints of the paths and the current path state. The contents of the output depends on the current state of the PPRC disk.

- If the disk is not part of a PPRC pair, the output shows all paths that originate from the LSS that the disk resides on, to any other LSS.
- If the disk is part of a PPRC pair, but has only one path group, the output displays paths that originate from the LSS on which the disk resides and lead to the LSS on which the partner disk resides.
- If the disk is part of a PPRC pair and has two path groups, the output displays all paths between the two LSSs on which the members of the PPRC pair reside.

Example output of the replication paths:

```
lspprc -c hdisk33

Displays all paths between LSS 01 and LSS 03
```

Source WWNN	SSID	LSS	Port	Target WWNN	SSID	LSS	Port	State
500507630AFFC16B	FF03	03	0301	500507630AFFC16B	FF01	01	0302	Up
500507630AFFC16B	FF03	03	0302	500507630AFFC16B	FF01	01	0303	Up
500507630AFFC16B	FF01	01	0300	500507630AFFC16B	FF03	03	0302	Up

-p

Displays path group information that is part of the specified PPRC disk.

Example output of path group information:

```
lspprc -p hdisk55
```

Path group id	WWNN	LSS	VOL	Path group status
0(s)	500507630affc16b	0xf	0x1c	
PRIMARY				
1	5005076303ffd2ea	0xc1	0x0	SECONDARY

path group id	path id	path status	parent	connection
0	0	Available	fscsi0	500507630a08016b,400f401c00000000
0	1	Available	fscsi0	500507630a08416b,400f401c00000000
1	2	Available	fscsi1	50050763030812ea,40c1400000000000
1	3	Available	fscsi1	50050763030852ea,40c1400000000000

When you use PPRC, the paths are grouped based on which LUN is accessed by the path, in the PPRC pair. The path group ID indicates how the paths are grouped, with all paths that have the same path group ID accessing the same LUN in the PPRC pair. A path group ID of -1 indicates that there are no paths that are configured from this initiator to the indicated LUN in the PPRC pair.

At any time, only one of the two path groups is selected for I/O operations to the hdisk. The selected path group is identified in the output by "(s)".

Item	Description
-v	Displays the VPD information of individual LUNs that are part of given PPRC disk. Example output of VPD information:

```
# lspprc -v hdisk0

Hyperswap lun unique identifier.....35203735544c3737313
037303000502a14ae07210790003IBMfcp

hdisk0 Primary          MPIO IBM 2107 FC Disk

Manufacturer.....IBM
Machine Type and Model.....2107900
ROS Level and ID.....2E313630
Serial Number.....75TL7710
Device Specific.(Z7).....0700
Device Specific.(Z0).....000005329F101002
Device Specific.(Z1).....700
Device Specific.(Z2).....075
Unique Device Identifier.....200B75TL771070007210790003IBMfcp
Logical Subsystem ID.....0x07
Volume Identifier.....0x00
Subsystem Identifier(SS ID)..0xFF07
Control Unit Sequence Number..00000TL771
Storage Subsystem WWNN.....500507630affc16b
Logical Unit Number ID.....4007400000000000

hdisk0 Secondary       MPIO IBM 2107 FC Disk

Manufacturer.....IBM
Machine Type and Model.....2107900
ROS Level and ID.....2E313630
Serial Number.....75TL7710
Device Specific.(Z7).....0900
Device Specific.(Z0).....000005329F101002
Device Specific.(Z1).....900
Device Specific.(Z2).....075
Unique Device Identifier.....200B75TL771090007210790003IBMfcp
Logical Subsystem ID.....0x09
Volume Identifier.....0x00
Subsystem Identifier(SS ID)..0xFF09
Control Unit Sequence Number..00000TL771
Storage Subsystem WWNN.....500507630affc16b
Logical Unit Number ID.....4009400000000000
```

Files

Item	Description
/usr/bin/lspprc	Contains the lspprc command.

lsque Command

Purpose

Displays the queue stanza name.

Syntax

lsque [**-c**] **-q***Name*

Description

The **lsque** command uses the **printf** subroutine to display the name of the queue stanza and associated attributes from the **/etc/qconfig** file.

Flags

Item	Description
-c	Causes colon output format for use by SMIT.
-q <i>Name</i>	Specifies the <i>Name</i> of the queue stanza that is sent to 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

1. To display the name of queue stanza lp0, enter:

```
lsque -qlp0
```

A list similar to the following is displayed:

```
lp0:  
  device = lpd0  
  host = neptune  
  rq = nlp0
```

2. To display the name of queue stanza lp0 in colon format, enter:

```
lsque -c -q lp0
```

A list similar to the following is displayed:

```
device:discipline:up:acctfile:host:s_statfilter:l_statfilter:rq  
lpd0:fcfs:true:false:neptune:::nlp0
```

Files

Item	Description
/usr/bin/lsque	Contains the lsque command.
/etc/qconfig	Contains the configuration file.

lsquedev Command

Purpose

Displays the device stanza name.

Syntax

```
lsquedev [ -c ] -qName -d Name
```

Description

The **lsquedev** command displays the name of the queue stanza and associated attributes from the **/etc/qconfig** file.

Flags

Item	Description
-c	Specifies colon output format for use by SMIT.
-d <i>Name</i>	Specifies the <i>Name</i> variable of the device stanza that is displayed.
-q <i>Name</i>	Specifies the <i>Name</i> variable of the queue containing the device stanza that is displayed.

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 device stanza `d1p0` on the `lp0` queue, type:

```
lsqueuedev -q lp0 -d d1p0
```

A listing similar to the following is displayed:

```
d1p0:
FILE = /dev/lp0
BACKEND = /usr/lib/lpd/piobe
```

2. To display device stanza `d1p0` on the `lp0` queue in colon format, type:

```
lsqueuedev -c -qlp0 -d d1p0
```

A listing similar to the following is displayed:

```
file:access:feed:header:trailer:backend:align
d1p0:/dev/lp0:read:never:never:never:/usr/lib/lpd/piobe:TRUE
```

Files

Item	Description
<code>/usr/bin/lsqueuedev</code>	Contains the lsqueuedev command.
<code>/etc/qconfig</code>	Contains the configuration file.

lsresource Command

Purpose

Displays bus resources for available devices in the system and recommends attribute values for bus resource resolution.

Syntax

```
lsresource [ -a | -r ] [ -d ] -l Name
```

Description

The **lsresource** command writes a list of assigned bus resources to standard out, or determines if the bus resources for devices resolve.

The **lsresource** command lets you display the currently assigned values for the bus resource attributes for the device specified by the given device logical name (**-l Name**). Use the **-a** flag to display the currently assigned values for all bus resource attributes for all devices under the same parent bus as the specified device.

Use the **-r** flag to determine if the bus resources for the specified device are resolvable. In this case, the **lsresource** command checks all devices under the same parent bus as the specified device, including defined but not available devices, to see if their bus resource attributes are resolvable. The **lsresource** command produces no output if all attributes resolve. The **lsresource** command provides information depending on the type of conflict detected if any device's bus resources are unresolvable. In some cases, the **lsresource** command can provide you with information that leads to the resolution of the conflict.

The **lsresource** command identifies the device name, attribute name, and a suggested value for the attribute if a conflict results from an attribute that only a user can change. Setting the attribute to the suggested value should resolve the conflict. This may aid in the configuration of devices with attributes that can only a user can change. Such devices include adapter cards which use jumpers or switches on the card to select values.

In some cases, a conflict may be due to an attribute which the system can normally adjust at boot time but is prevented from doing so at run time because the device is in the Available state. In these situations, the **lsresource** command will indicate that the configuration will be resolved by rebooting the system.

It is possible that multiple user changeable attributes will be identified when unresolvable conflicts occur. These may be for the device specified by the given device logical name (**-l Name**) or for other devices in the system. All of the identified attributes will need to be changed to resolve the conflict. It may even be the case where user changeable attributes are identified and a reboot is indicated. In this case, all of the identified attributes will need to be changed and the system rebooted to resolve the conflicts.

Finally, **lsresource** may determine that the set of devices currently defined in the devices configuration database can not be resolved regardless of attributes being changed or the system rebooted. In this case, a list of the devices which could not be resolved is written to standard out. If the problem has resulted from a new device just being defined, that device should be removed, or the devices listed by **lsresource** should be removed. If the problem is not resolved by removing devices, there could be additional problems on the next reboot. This is because the order in which devices are resolved at boot time may differ from the order they are resolved by **lsresource**, resulting in a different set of unresolvable devices at boot time. If the set of unresolvable devices at boot time should now include a device needed for booting, problems such as no console being configured or the system failing to boot could occur.

The following applies when **lsresource** is used to list currently assigned bus resource values (the **-r** flag is not specified).

The **TYPE** field in the output listing contains the following symbols:

Ite	Description
m	
B	Bus Memory Address Values
M	Bus Memory Address Values
O	I/O Address Values
I	Bus Interrupt Levels
N	Non-sharable Bus Interrupt Levels
A	DMA Arbitration Level

The **S** column denotes shared attributes. These are attributes which are required to be set to the same value. They are grouped by the number specified in the column. All attributes with a 1 by them must be set to the same value, all attributes with a 2 by them must be set to the same value, and so on. In some cases, two or more interrupt attributes may be set to the same value but have no numbers in the **S** column indicating that they are shared. This is because the values are not required to be the same but just happen to be set to the same value because they could not be assigned their own unique values.

The **G** column denotes attributes in a group. These are a set of attributes whose values depend on each other. If one is changed to the next possible value, the rest of the attributes in the group must also be changed to the next possible value. Their groupings are indicated by the number specified in the column. All attributes with a 1 by them are in the same group, all attributes with a 2 by them are same group, and so on.

On some models, the interrupt value displayed may be followed by a value enclosed in parenthesis. This is not part of the interrupt value but serves to identify the interrupt controller to which the interrupt is associated. The identifier consists of a letter followed by a number, such as A0. The letter indicates the type of interrupt controller and the number distinguishes between multiple instances of that type of controller. There are two types of interrupt controllers that may be identified:

Ite Description

m

- A Indicates an AT interrupt controller.
- B Indicates a non-AT interrupt controller.

Flags

Item Description

- a** Specifies that all allocated bus resource attributes for all devices connected to the same top parent bus as the device specified with the **-l** flag are to be displayed. This flag cannot be used with the **-r** flag.
- d** Specifies that the attribute text descriptions are to be included in the output.
- l Name** (Lowercase L) Specifies the logical name of the device attributes to display.
- r** Specifies to attempt to resolve all bus resources of all devices connected to the same top parent bus as the device specified with the **-l** flag. This will include all devices that are in the DEFINED state. The **lsresource** command will display any conflicts and advise the user on changeable values. No changes to the ODM database are made. This flag cannot be used with the **-a** flag.

Security

Access Control: Any User

Auditing Events: N/A

Examples

1. To list bus attributes for the token ring device, enter:

```
lsresource -l tok0
```

The system displays a message similar to the following:

TYPE	DEVICE	ATTRIBUTE	S	G	CURRENT VALUE
M	tok0	dma_bus_mem			0x003b2000 - 0x003f1fff
O	tok0	bus_io_addr			0x000086a0 - 0x000086af
N	tok0	bus_infr_lvl			3
A	tok0	dma_lvl			7

2. To list bus attributes for all devices, enter:

```
lsresource -a -l tok0
```

The system displays a message similar to the following:

TYPE	DEVICE	ATTRIBUTE	S	G	CURRENT	VALUE
M	bus0	bus_iocc_mem			0x00ffffff0	- 0x00ffffff
M	gda0	vram_start	1		0x00400000	- 0x007fffff
M	gda0	bus_mem_start			0x000c0000	- 0x000c1fff
M	gda0	dma1_start			0x00800000	- 0x009fffff
M	gda0	dma2_start			0x00a00000	- 0x00bfffff
M	gda0	dma3_start			0x00c00000	- 0x00dfffff
M	gda0	dma4_start			0x01000000	- 0x011fffff
M	scsi0	bus_mem_addr			0x000e0000	- 0x000e0fff
M	scsi0	dma_bus_mem			0x00100000	- 0x00301fff
M	tok0	dma_bus_mem			0x003b2000	- 0x003f1fff
O	da0	bus_io_addr			0x00000060	- 0x0000006f
O	siokta0	bus_io_addr			0x00000050	- 0x0000005f
O	sioma0	bus_io_addr			0x00000048	- 0x0000004f
O	ppa0	bus_io_addr			0x00000078	- 0x0000007f
O	gda0	bus_addr_start	1		0x00002110	- 0x0000211f
O	tok0	bus_io_addr			0x000086a0	- 0x000086af
I	siokta0	bus_intr_lvl			1	(A0)
I	sioma0	bus_intr_lvl			1	(A0)
I	ppa0	bus_intr_lvl			13	(A0)
I	gda0	int_level			9	(A0)
I	scsi0	bus_intr_lvl			14	(A0)
N	fda0	bus_intr_lvl			6	(A0)
N	tok0	bus_intr_lvl			3	(A0)
A	fda0	dma_lvl			0	
A	gda0	dma_channel			3	
A	scsi0	dma_lvl			4	
A	tok0	dma_lvl			7	

3. To report the outcome of a resolution of device attributes, enter:

```
lsresource -r -d -l tok0
```

Depending on the outcome of the resolution, different messages may be displayed. The output below signifies to a user that the resolution can be successful if changes are made, i.e., the attributes are changed to the suggested values.

```
lsresource: The attribute(s) for some device(s) in the system could
not be resolved. To resolve conflicts, attribute(s) need to be
modified. A suggested value for each attribute is provided.
```

DEVICE	ATTRIBUTE	CURRENT	SUGGESTED	DESCRIPTION
ent1	bus_intr_lvl	11	5	Bus interrupt level
ent1	bus_mem_addr	0xc0000	0xc4000	Bus memory address
ent1	bus_io_addr	0x300	0x320	Bus I/O address
ent2	bus_intr_lvl	11	7	Bus interrupt level
ent2	bus_mem_addr	0xc0000	0xc8000	Bus memory address

Files

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

lsresponse Command

Purpose

Lists information about one or more responses.

Syntax

```
lsresponse [-a] [-C | -l | -t | -d | -D delimiter] [-A] [-q] [-U] [-x] [-b] [-h] [-TV]
[response1[, response2, ...] :node_name]
```

Description

The `lsresponse` command lists the following information about defined responses:

Field	Description
ResponseName	The name of the response.
Node	The location of the response.
Action	The name of an action.
DaysOfWeek	<p>The days of the week when the action can be run. DaysOfWeek and TimeOfDay together define the interval when the action can be run.</p> <p>The values for the days can be separated by plus signs (+) or displayed as a range of days separated by a hyphen (-). Multiple DaysOfWeek values are separated by commas (,). The number of DaysOfWeek values must match the number of TimeOfDay values. The values for each day follow:</p> <p>1 Sunday</p> <p>2 Monday</p> <p>3 Tuesday</p> <p>4 Wednesday</p> <p>5 Thursday</p> <p>6 Friday</p> <p>7 Saturday</p>
TimeOfDay	<p>The time range when Action can be run, consisting of the start time followed by the end time separated by a hyphen. DaysOfWeek and TimeOfDay together define the interval when the action can be run.</p> <p>The time is in 24-hour format (HHMM), where the first two digits represent the hour and the last two digits represent the minutes. Multiple TimeOfDay values are separated by commas (,). The number of DaysOfWeek values must match the number of TimeOfDay values.</p>
ActionScript	The script or command to run for the action.
ReturnCode	The expected return code for ActionScript.
CheckReturnCode	Indicates whether the actual return code for ActionScript is compared to its expected return code. The values are: y (yes) and n (no).
EventType	The type of event that causes the action to be run: event, rearm event, or both.
StandardOut	Indicates whether standard output is directed to the audit log. The values are: y (yes) and n (no).
EnvironmentVariables	Indicates any environment variables that will be set before the action is run.
UndefRes	Indicates whether the action is to be run if a monitored resource becomes undefined. The values are: y (yes) and n (no).

Field	Description
Locked	Indicates whether the resource is locked or unlocked.
EventBatching	Indicates whether the response action supports event batching.

To get a list of all response names, run the `lsresponse` command alone without any response names specified. A list of all response names is returned. The default format in this case is tabular.

Specifying a node name after the response names limits the display to the responses defined on that node. List all of the responses on a node by specifying a colon (:) followed by the node name. The node name is a node within the management scope determined by the `CT_MANAGEMENT_SCOPE` environment variable. The management scope determines the list of nodes from which the responses are listed. For local scope, only responses on the local node are listed. Otherwise, the responses from all nodes within the domain are listed.

To see all the information about all response names, specify the `-A` flag with the `lsresponse` command. The `-A` flag causes all information about a response to be listed when no response names are specified. When all of the information about all responses is listed, the long format is the default.

When more than one response is specified, the response information is listed in the order in which the responses are entered.

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, `lsresponse -a` with `CT_MANAGEMENT_SCOPE` not set will list the management domain. In this case, to list the peer domain, set `CT_MANAGEMENT_SCOPE` to 2.

-A

Displays all of the attributes of the response.

-b

Displays only the responses that support event batching.

-C

Displays the **mkresponse** command that can be used to create the response and one of its actions. If more than one response is specified, each **mkresponse** command appears on a separate line. This flag is ignored when no responses are specified. This flag overrides the **-l** flag.

-d

Specifies delimiter-formatted output. The default delimiter is a colon (:). Use the `-D` flag if you wish to change the default delimiter.

-D delimiter

Specifies delimiter-formatted output that uses the specified delimiter. Use this flag to specify something other than the default, colon (:). For example, when the data to be displayed contains colons, use this flag to specify another delimiter of one or more characters.

-l

Displays the response information on separate lines (long form).

-q

Does not return an error when response does not exist.

- t**
Displays the response information in separate columns (table form).
- U**
Indicates whether the resource is locked.
- x**
Suppresses headers when 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

response1[, response2, ...]

This parameter can be a response name or a substring of a response name. You can specify more than one response name. When it is a substring, any defined response name that contains the substring is listed.

node_name

Specifies the node 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 read permission for the IBM.EventResponse resource class to run `lsresponse`. Permissions are specified in the access control list (ACL) file on the contacted system. See the *Administering RSCT* 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

1. To list all of the responses, run this command:

```
lsresponse
```

The output will look like this:

```
ResponseName
"E-mail root anytime"
"E-mail root first shift"
"Critical notifications"
"Generate SNMP trap"
```

2. To see which resources are locked, run this command:

```
lsresponse -U
```

The output will look like this:

ResponseName	Node	Locked
"Broadcast event on-shift"	"nodeA"	"No"
"E-mail root off-shift"	"nodeA"	"No"
"E-mail root anytime"	"nodeA"	"No"
"Log event anytime"	"nodeA"	"No"
"Informational notifications"	"nodeA"	"No"
"Warning notifications"	"nodeA"	"No"

```
"Critical notifications"      "nodeA"  "No"
"Generate SNMP trap"         "nodeA"  "No"
```

3. To list general information about the response "Critical notifications", run this command:

```
lsresponse "Critical notifications"
```

The output will look like this:

```
ResponseName = "Critical notifications"
Node         = "nodeA"
Action       = "Log Critical Event"
DaysOfWeek   = 1+2+7
TimeOfDay    = 0000-2400
ActionScript = "/opt/rsct/bin/logevent /tmp/
criticalEvents"
ReturnCode   = 0
CheckReturnCode = "y"
EventType    = "b"
StandardOut  = "y"
EnvironmentVars = "'Env1=5','Env=10'"
UndefRes     = "n"

ResponseName = "Critical notifications"
Node         = "nodeA"
Action       = "E-mail root"
DaysOfWeek   = 6+2,6+2,6+5
TimeOfDay    = 1700-2400,0000-0800,0000-2400
ActionScript = "/opt/rsct/bin/notifyscript root"
ReturnCode   = 0
CheckReturnCode = "y"
EventType    = "b"
StandardOut  = "y"
EnvironmentVars = ""
UndefRes     = "n"
```

4. To list the command that would create the response "Critical notifications" along with one of its actions, run this command:

```
lsresponse -C "Critical notifications"
```

The output will look like this:

```
mkresponse -n "Log Critical Event" -d 1+2+7 -t 0000-2400 \
-s "usr/sbin/rsct/bin/logevent /tmp/criticalEvents" \
-e b -r 0 "Critical notifications"
```

5. To list all responses that have the string E-mail in their names, run this command:

```
lsresponse "E-mail"
```

The output will look like this:

```
ResponseName = "E-mail root anytime"
Action       = "E-mail root"
:
ResponseName = "E-mail root first shift"
Action       = "E-mail root"
```

Location

`/opt/rsct/bin/lsresponse`

lsrole Command

Purpose

Displays role attributes.

Syntax

```
lsrole [-R load_module] [-c | -f | -C] [-a List] { ALL | Name [ ,Name ] ... }
```

Description

The **lsrole** command displays the role attributes. You can use this command to list all attributes of all the roles or all the attributes of specific roles. Since there is no default parameter, you must enter the **ALL** keyword to see the attributes of all the roles. By default, the **lsrole** command displays all role attributes. To view selected attributes, use the **-a List** flag. If one or more attributes cannot be read, the **lsrole** command lists as much information as possible.

By default, the **lsrole** command lists each role's attributes on one line. It displays attribute information as *Attribute=Value* definitions, each separated by a blank space. To list the role attributes in stanza format, use the **-f** flag. To list the information as colon-separated records, use the **-c** flag.

You can use the System Management Interface Tool (SMIT) **smit lsrole** fast path to run this command.

If the system is configured to use multiple domains for the role database, the roles, as specified by the *Name* parameter, are searched from the domains in the order specified by the **secorder** attribute of the roles stanza in the **/etc/nscontrol.conf** file. If duplicate entries exist in multiple domains, only the first entry instance is listed. Use the **-R** flag to list the roles from a specific domain.

The **lsrole** command only lists the role definitions available in the roles database. If the system is operating in enhanced Role Based Access Control (RBAC) mode, the information in the roles database might differ from what is used for security considerations on the system in the kernel security tables (KST). To view the state of the roles database in the KST, use the **lskst** command.

Flags

Item	Description
-a List	<p>Lists the attributes to display. The <i>List</i> variable can include any attribute that is defined in the chrole command. Specify more than one attributes with a blank space between attribute names. If an empty list is specified, only the role names are displayed. In addition to the attributes defined in the chrole command, the following attributes can also be listed with the -a flag:</p> <p>all_auths Traverses the role hierarchy of the specified roles and gathers all the authorizations. The all_auths attribute differs from the authorizations attribute because the lsrole command only lists the explicit authorizations of the specified roles for that attribute.</p> <p>users Displays the users that are granted the specified roles.</p> <p>description Displays the text description of the role as indicated by the dfltmsg, msgcat, msgset and msgnum attributes for the role.</p>
-c	Displays the role attributes in colon-separated records, as follows:

```
# role: attribute1: attribute2: ...  
Role: value1: value2: ...
```

Item	Description
-C	<p>Displays the role attributes in colon-separated records that are easier to parse than the output of the -c flag:</p> <pre>#role:attribute1:attribute2: ... role:value1:value2: ... role2:value1:value2: ...</pre> <p>The output is preceded by a comment line that has details about the attribute represented in each colon-separated field. If you specified the -a flag, the order of the attributes matches the order specified in the -a flag. If a role does not have a value for a given attribute, the field is still displayed but is empty. The last field in each entry is ended by a newline character rather than a colon.</p>
-f	<p>Displays the output in stanzas, with each stanza identified by a role name. Each <i>Attribute=Value</i> pair is listed on a separate line:</p> <pre>Role: attribute1=value attribute2=value attribute3=value</pre>
-R <i>load_module</i>	Specifies the loadable module to list roles from.

Security

The **lsrole** command is a privileged command. You must assume a role that has the following authorization to run the command successfully.

Item	Description
aix.security.role.list	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 that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Files Accessed:

Mode	File
r	<i>/etc/security/roles</i>

Examples

- To display the role `rolelist` and groups of the role `ManageAllUsers` in a colon format, use the following command:

```
lsrole -c -a rolelist groups ManageAllUsers
```

Information similar to the following appears:

```
# role: rolelist:groups
ManageAllUsers: ManagerBasicUser:security
```

- To list all attributes of the `ManageAllUsers` role from LDAP, use the following command:

```
lsrole -R LDAP ManageAllUsers
```

All the attribute information appears, with each attribute separated by a blank space.

Files

Item	Description
<u>/etc/security/roles</u>	Contains the attributes of roles.

lsrpdomain Command

Purpose

Displays peer domain information for the node.

Syntax

```
lsrpdomain [-o | -O] [-l | -t | -d | -D delimiter] [-x] [-h] [-TV] [peer_domain]
```

Description

The `lsrpdomain` command displays information about the peer domains that the node where the command runs belongs to. Use the command's flags and parameters to specify which information you want to display and how you want to display it. When you specify the name of a peer domain, the command displays information about that peer domain only. The `-o` and `-O` flags also limit the information this command displays. The `-o` flag displays information only about the online peer domain. The `-O` flag displays information only about peer domains that are offline.

By default, the `lsrpdomain` command displays information in table format (`-t`).

Some of the peer domain information that is displayed follows:

Field	Description
Name	The name of the peer domain.
RSCTActiveVersion	The version of RSCT that is active in the peer domain.
MixedVersions	Indicates whether more than one version of RSCT is active in the peer domain.
TSPort	The topology services port number.
GSPort	The group services port number.
OpState	The current state of the peer domain.

Flags

- o** Displays information about the node's online peer domain.
- O** Displays information about peer domains that are offline for the node.
- l** Displays the information on separate lines (long format).
- t** Displays the information in separate columns (table format). This is the default.
- d** Displays the information using delimiters. The default delimiter is a colon (:). Use the `-D` flag if you want to change the default delimiter.

-D delimiter

Displays the information using the specified delimiter. Use this flag to specify a delimiter other than the default colon (:) — when the information you want to display contains colons, for example. You can use this flag to specify a delimiter of one or more characters.

-x

Excludes the header (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***peer_domain***

Specifies the name of the peer domain about which you want to display information. You can specify a peer domain name or a substring of a peer domain name for this parameter. If you specify a substring, the command displays information about any defined peer domain with a name that contains the substring.

Security

The user of the `lsrpdomain` command needs read permission for the `IBM.PeerDomain` resource class on the node on which the command runs. 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 the node for which the peer domain information is requested.

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

1. To display general information about the peer domains to which nodeA belongs, run this command on nodeA:

```
lsrpdomain
```

The output will look like this:

Name	OpState	RSCTActiveVersion	MixedVersions	TSPort	GSPort
ApplDomain	Online	2.5.0.0	No	12347	12348

2. To display general information about the peer domains to which nodeA belongs, with the default delimiter (but without the heading), run this command on nodeA:

```
lsrpdomain -xd
```

The output will look like this:

```
ApplDomain:Online:2.5.0.0:No:12347:12348:
```

3. To display general information about the peer domains to which nodeA belongs, in long format, run this command on nodeA:

```
lsrpdomain -l
```

The output will look like this:

```
Name           = ApplDomain
OpState        = Online
RSCTActiveVersion = 2.5.0.0
MixedVersions  = No
```

TSPort = 12347
GSPort = 12348

Location

/opt/rsct/bin/lsrpdomain

lsrpnod Command

Purpose

Displays information about one or more of the nodes that are defined in the online peer domain.

Syntax

lsrpnod [**-i**] [**-l** | **-t** | **-d** | **-D delimiter**] **-o** | **-O** | **-L** | **-P** | **-Q**] [**-B**] [**-x**] [**-h**] [**-TV**] [*node_name*]

lsrpnod -p *peer_domain* [**-l** | **-t** | **-d** | **-D delimiter**] [**-x**] [**-h**] [**-TV**]

Description

The **lsrpnod** command displays information about one or more of the nodes that are defined in the online peer domain. Use the command's flags and parameters to specify which information you want to display and how you want to display it. When you specify a node name, the command displays information about that node only.

The **-o**, **-O**, and **-L** flags also limit the information this command displays. The **-o** flag displays information about nodes that are online. The **-O** flag displays information about nodes that are offline. The **-L** flag displays information about the local node, which is the node the command runs on.

The **-P** flag displays additional node configuration information related to group services group leader selection. The **-Q** flag displays additional node configuration information related to quorum decisions. The **-B** flag displays additional node configuration information related to the tiebreaker mechanism.

By default, the **lsrpnod** command displays information in table format (**-t**).

Some of the node information that is displayed follows:

Field	Description
Name	The name of the node in the peer domain.
OpState	The operational state of the node.
RSCTVersion	The version of RSCT that is active in the node.

The following fields are displayed when you specify the **-i** flag:

Field	Description
NodeNum	The node number used by topology services and group services. This number is unique within the cluster.
NodeID	The unique node identifier.

Along with other fields (depending on the flags specified), this field is displayed when you specify the **-P** flag:

Field	Description
Preferred	Indicates whether the node is a group services group leader candidate.

Along with other fields (depending on the flags specified), this field is displayed when you specify the **-Q** flag:

Field	Description
Quorum	Indicates whether the node participates in quorum decisions.

Along with other fields (depending on the flags specified), this field is displayed when you specify the **-B** flag:

Field	Description
Tiebreaker	Indicates whether the node has access to the peer domain's tiebreaker mechanism.

See the *Administering RSCT* guide for information about group services group leader selection, quorum decisions, and the tiebreaker mechanism.

Flags

-d

Displays the information using delimiters. The default delimiter is a colon (:). Use the **-D** flag if you want to change the default delimiter.

-D delimiter

Displays the information using the specified delimiter. Use this flag to specify a delimiter other than the default colon (:). — when the information you want to display contains colons, for example. You can use this flag to specify a delimiter of one or more characters.

-i

Displays the node number and node ID for the node. The node number is used by topology services and group services and is unique within the cluster. The node ID is the unique node identifier.

-l

Displays the information on separate lines (long format).

-L

Displays information about the local node only, which is the node that the command runs on.

-o

Displays information about the nodes that are online in the peer domain.

-O

Displays information about the nodes that are offline in the peer domain.

-p peer_domain

Displays information about nodes defined in an *offline* peer domain that the local node belongs to. (By default, the `lsnpnode` command displays information about the nodes that are defined in the domain where you are currently *online*.) However, this information might not reflect changes that are made to the domain after the local node is taken offline, because an offline node might not have the latest configuration.

The **-p** flag ignores the `CT_CONTACT` environment variable. You must have root access to use the **-p** flag.

-P

Indicates whether the node is a group services group leader candidate. **yes** is displayed if the node can be a group services group leader. **no** is displayed if the node cannot be a group services group leader. See the *Administering RSCT* for more information about group services group leader selection.

-Q

Indicates whether the node participates in quorum decisions. **yes** is displayed if the node participates in quorum decisions. **no** is displayed if the node does not participate in quorum decisions. See the *Administering RSCT* for more information on quorum decisions.

-B

Indicates whether the node has access to the peer domain's tiebreaker mechanism. `yes` is displayed if the node has access to the peer domain's tiebreaker mechanism. `no` is displayed if the node does not have access to the peer domain's tiebreaker mechanism. See the *Administering RSCT* for more information on the tiebreaker mechanism.

-t

Displays the information in separate columns (table format). This is the default format.

-x

Excludes the header (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

node_name

Specifies the name of the node about which you want to display information. You can specify a node name or a substring of a node name for this parameter. If you specify a substring, the command displays information about any defined node with a name that contains the substring.

Security

The user of the `lsrnode` command needs read permission for the `IBM.PeerNode` resource class on the node this command runs on. 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 in the peer domain.

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

1. To display general information about the nodes in the online peer domain that nodeA belongs to, run this command on nodeA:

```
lsrpnode
```

The output will look like this:

```
Name      OpState  RSCTVersion
nodeA     Online   3.1.4.0
nodeB     Online   3.1.4.0
nodeC     Offline  3.1.4.0
```

2. To display general information about the nodes in the online peer domain that nodeA belongs to, with the default delimiter (but without the heading), run this command on nodeA:

```
lsrpnode -xd
```

The output will look like this:

```
nodeA:Online:3.1.4.0:
nodeB:Online:3.1.4.0:
nodeC:Offline:3.1.4.0:
```

3. To display general information about the nodes in the online peer domain that nodeA belongs to, in long format, run this command on nodeA:

```
lsrpnode -l
```

The output will look like this:

```
Name      = nodeA
OpState    = Online
RSCTVersion = 3.1.4.0
```

```
Name      = nodeB
OpState   = Online
RSCTVersion = 3.1.4.0

Name      = nodeC
OpState   = Offline
RSCTVersion = 3.1.4.0
```

4. To display general information about the nodes in the online peer domain that nodeA belongs to, including the node number and node ID, run this command on nodeA:

```
lsrpnode -i
```

The output will look like this:

Name	OpState	RSCTVersion	NodeNum	NodeID
nodeA	Online	3.1.4.0	2	40a514bed9d82412
nodeB	Online	3.1.4.0	1	47fe57098f4ec4d9

5. To display general information about the nodes in the online peer domain to which **nodeA** belongs, including the preferred group services group leader information, run this command on **nodeA**:

```
lsrpnode -P
```

The output will look like this:

Name	OpState	RSCTVersion	Preferred
nodeA	Online	3.1.4.0	yes
nodeB	Online	3.1.4.0	no

6. To display general information about the nodes in the online peer domain to which **nodeA** belongs, including the quorum information, run this command on **nodeA**:

```
lsrpnode -Q
```

The output will look like this:

Name	OpState	RSCTVersion	Quorum
nodeA	Online	3.1.4.0	no
nodeB	Online	3.1.4.0	yes
nodeC	Online	3.1.4.0	yes

7. To display general information about the nodes in the online peer domain to which **nodeA** belongs, including quorum and tiebreaker information, run this command on **nodeA**:

```
lsrpnode -QB
```

The output will look like this:

Name	OpState	RSCTVersion	Quorum	Tiebreaker
nodeA	Online	3.1.4.0	no	no
nodeB	Online	3.1.4.0	yes	yes
nodeC	Online	3.1.4.0	yes	yes

Location

`/opt/rsct/bin/lsrpnode`

lsrset Command

Purpose

Displays system rset contents.

Syntax

```
lsrset [ -X ] [ -f ] [ -v | -o ] [ [ -S ] -r rsetname | -n namespace | -a ]
```

or

```
lsrset [ -X ] [ -P ] [ -v | -o ] -p pid
```

Description

The **lsrset** command displays information contained in rsets stored in the system registry or rsets attached to a process.

Flags

Item	Description
-f	Displays rset owner, group, and mode data.
-v	Verbose mode. Displays resources contained in the rset, rset owner, group and mode data.
-o	Displays only the online resources contained in the rset. The default is to display all resources.
-p <i>pid</i>	Displays the effective rset attached to this process.
-r <i>rsetname</i>	Displays the rset with this name in the system registry. The name consists of a namespace and an <i>rsname</i> separated by a "/" (slash). Both the <i>namespace</i> and <i>rsname</i> may contain up to 255 characters. See the rs_registername() service for additional information about character set limits of rset names.
-n <i>namespace</i>	Displays all rsets in this <i>namespace</i> in the system registry.
-a	Displays all rsets in the system registry.
-P	Displays the partition rset attached to the specified process.
-S	Displays the resources contained in this rset if it were to be scheduled with the -S hint with either the execrset or the attachrset command. The rset does not need to be an exclusive rset. This is to be contrasted with the attachrset and execrset commands, which require exclusive rsets to be specified with the -S flag.
-X	Prints all available characters of each user and group name instead of truncating to the first 8 characters.

Examples

1. To display all resources for all rsets in the system registry, type:

```
lsrset -v -a
```

2. To display a summary of the effective rset attached to pid 28026, type:

```
lsrset -p 28026
```

3. To display the online resources in the effective rset attached to pid 28026, type:

```
lsrset -o -p 28026
```

4. To display all the resources in the effective rset attached to pid 28026, type:

```
lsrset -v -p 28026
```

5. To display online resources for all rsets in the system registry, type:

```
lsrset -a -o
```

6. To display all resources for all rsets in the system registry with expanded user and group name, type:

```
lsrset -X -v -a
```

Files

Item	Description
<code>/etc/bin/lsrset</code>	Contains the lsrset command

lsrsrc Command

Purpose

Displays attributes and values for a resource or a resource class.

Syntax

To display the attributes and values for a *resource*:

```
lsrsrc [-s "selection_string"] [-a | -N { node_file | "-" }][ -A p | d | b ][ -p property ][ -l | -i | -t |  
-d | -D delimiter ] [-x] [-h] [-TV] [resource_class] [attr...]
```

```
lsrsrc -r [-s "selection_string"] [-a | -N { node_file | "-" }][ -l | -i | -t | -d | -D delimiter ] [-x]  
[-h] [-TV] [resource_class]
```

To display the attributes and values for a *resource class*:

```
lsrsrc -c [ -A p | d | b ][ -p property ][ -l | -i | -t | -d | -D delimiter ] [-x] [-a] [-h] [-TV]  
resource_class [attr...]
```

```
lsrsrc -C domain_name... [ -A p | d | b ][ -p property ][ -l | -i | -t | -d | -D delimiter ] [-x] [-h]  
[-TV] resource_class [attr...]
```

To display a list of all of the resource classes:

```
lsrsrc
```

Description

The **lsrsrc** command displays the persistent and dynamic attributes and their values for a resource or a 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.

When one or more attribute names are specified, these names and their values are displayed in the order specified, provided that each of the specified attribute names is valid. When no attribute names are specified:

- the `-A p | d | b` flag controls whether persistent attributes or dynamic attributes or both — and their values — are displayed.
- only attributes that are defined as `public` are displayed. Use the `-p` flag to override this default.

For best performance, specify either the `-A p` flag or only persistent attributes as parameters.

Specify the `-r` flag to display only the resource handles associated with the resources for the specified resource class.

To display a list of the attributes and values for a resource class, specify the `-c` flag.

By default, the resource attributes and values are displayed in long format. Use the `-t`, `-d`, or `-D` flag to display the resources in table format or delimiter-formatted output.

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 `lsrsrc` command does not list any attributes that have a datatype defined as `ct_none` (Quantum, for example). RMC does not return attribute values for attributes that are defined as Quantum. To list attribute definitions, use the `lsrsrcdef` command.

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, `lsrsrc -a` with `CT_MANAGEMENT_SCOPE` not set will list the management domain. In this case, to list the peer domain, set `CT_MANAGEMENT_SCOPE` to 2.

-A p | d | b

Specifies an attribute type. By default only persistent attributes are displayed. This flag can be used only when no attribute names are specified on the command line.

p

Displays only persistent attributes.

d

Displays only dynamic attributes.

b

Displays both persistent and dynamic attributes.

For best performance, specify the `-A p` flag.

-c

Displays the attributes for the resource class. This flag overrides the `-r` flag.

-C domain_name...

Displays the class attributes of 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. To display class attributes of a globalized resource class on all peer domains defined on the management server, use the `-c` flag with the `-a` flag instead of `-C`. The command returns the name of the peer domain in the form of an attribute `ActivePeerDomain`. This is not an actual attribute, but is presented as such to indicate which peer domain is being displayed.

-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 something 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.

-i

Generates a template of `resource_data_input_file` that can then, after appropriate editing, be used as input to the `mkrsrc` command. The output is displayed in long (stanza) format. All required and optional attributes that can be used to define a resource are displayed. The attribute data type is displayed as the value in the `attr=value` pairs. It is suggested that when you use this flag, the output of the `lsrsrc` command be directed to a file. This flag overrides the `-s` and `-A d` flags.

-l

Specifies long formatted output. Each attribute is displayed on a separate line. This is the default display format. If the `lsrsrc` command is issued with the `-l` flag, but without a resource class name, the `-l` flag is ignored when the command returns the list of defined resource class names.

-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.

-p *property*

Displays attributes with the specified *property*. By default, only public attributes are displayed. To display all of the attributes regardless of the property, use the `-p 0` flag. Use this flag in conjunction with the `-A` flag when no attributes are specified on the command line.

Persistent attribute properties:

0x0001

read_only

0x0002

reqd_for_define (required)

0x0004

inval_for_define (not valid)

0x0008

option_for_define (optional)

0x0010

selectable

0x0020

public

Dynamic attribute properties:

0x0020

public

A decimal or hexadecimal value can be specified for the property. To display attributes and their values for all attributes that have one or more properties, "OR" the properties of interest together and then specify the "OR"ed value with the `-p` flag. For example, to display attributes and their values for all persistent attributes that are either `reqd_for_define` or `option_for_define`, enter:

```
lsrsrc -p 0x0a
```

-r

Displays the resource handles for the resources that match the specified selection string or all resources when no selection string is specified.

-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 may be listed in a selection string. For information on how to specify selection strings, see the *RSCT: Administration Guide*.

-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

resource_class

Specifies the name of the resource class with the resources that you want to display.

attr...

Specifies one or more attribute names. Both persistent and dynamic attribute names can be specified to control which attributes are displayed and their order. Zero or more attributes can be specified. Attributes must be separated by spaces.

Security

The user needs read permission for the *resource_class* specified in `lsrsrc` to run `lsrsrc`. 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. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

1. To list the names of all of the resource classes, enter:

```
lsrsrc
```

The output will look like this:

```
class_name
"IBM.Association"
"IBM.Condition"
"IBM.EventResponse"
"IBM.Host"
"IBM.Ethernet"
"IBM.TokenRing"
...
```

2. To list the persistent attributes for resource IBM.Host that have 4 processors, enter:

```
lsrsrc -s "NumProcessors == 4" -A p -p 0 IBM.Host
```

The output will look like this:

```
Resource Persistent Attributes for: IBM.Host
resource 1:
  Name           = "c175n05.ppd.pok.ibm.com"
  ResourceHandle = "0x4008 0x0001 0x00000000 0x0069684c 0x0d7f55d5
0x0c32fde3"
  Variety        = 1
  NodeList       = {1}
  NumProcessors  = 4
  RealMemSize    = 1073696768
```

3. To list the public dynamic attributes for resource IBM.Host on node 1, enter:

```
lsrsrc -s 'Name == "c175n05.ppd.pok.ibm.com"' -A d IBM.Host
```

The output will look like this:

```
Resource Dynamic Attributes for: IBM.Host
resource 1:
  ProcRunQueue      = 1.03347987093142
  ProcSwapQueue     = 1.00548852941929
  TotalPgSpSize     = 65536
  TotalPgSpFree     = 65131
  PctTotalPgSpUsed  = 0.61798095703125
  PctTotalPgSpFree  = 99.3820190429688
  PctTotalTimeIdle  = 0
  PctTotalTimeWait  = 51.5244382399734
  PctTotalTimeUser  = 12.8246006482343
  PctTotalTimeKernel = 35.6509611117922
  PctRealMemFree    = 66
  PctRealMemPinned  = 4
  RealMemFramesFree = 173361
  VMPgInRate        = 0
  VMPgOutRate       = 0
  VMPgFaultRate     = 0
  ...
```

4. To list the Name, Variety, and ProcessorType attributes for the IBM.Processor resource on all the online nodes, enter:

```
lsrsrc IBM.Processor Name Variety ProcessorType
```

The output will look like this:

```
Resource Persistent Attributes for: IBM.Processor
resource 1:
  Name           = "proc3"
  Variety        = 1
  ProcessorType  = "PowerPC_604"
resource 2:
  Name           = "proc2"
  Variety        = 1
  ProcessorType  = "PowerPC_604"
resource 3:
  Name           = "proc1"
  Variety        = 1
  ProcessorType  = "PowerPC_604"
resource 4:
  Name           = "proc0"
  Variety        = 1
  ProcessorType  = "PowerPC_604"
```

5. To list both the persistent and dynamic attributes for the resource class IBM.Condition, enter:

```
lsrsrc -c -A b -p 0 IBM.Condition
```

The output will look like this:

```
Resource Class Persistent and Dynamic Attributes for: IBM.Condition
resource 1:
    ResourceType = 0
    Variety      =
0
```

6. To list the nodes in the cluster that have at least four processors, 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:

```
lsrsrc -s "NumProcessors >= 4" -N /tmp/common/node_file -t IBM.Host \
Name NumProcessors
```

The output will look like this:

```
Resource Persistent Attributes for IBM.Host
Name          NumProcessors
"node1.ibm.com" 4
"node2.ibm.com" 4
```

Location

`/opt/rsct/bin/lsrsrc`

lsrsrcassoc Command

Purpose

Retrieves a list of resources that are associated with a class using an association provider.

Syntax

```
lsrsrcassoc [-s "source_selection_string"] [-c association_class] [-d association_endpoint_class] [-S "destination_selection_string"] [-o role] [-R result_role] [-h] [-TV] source_class_name [property_list...]
```

Description

You can use the `lsrsrcassoc` command to learn about the relationships among CIM resources.

This command is an interface into the association query mechanism of the Common Information Model (CIM) resource manager. Association providers that are registered with the CIM resource manager are called to retrieve association data. Before using `lsrsrcassoc`, it might be helpful to run the `lsassocmap` command to find out which association classes are known to the resource monitoring and control (RMC) subsystem.

You must specify a source class name with the `lsrsrcassoc` command. With no flags specified, `lsrsrcassoc` retrieves all resources associated with every resource of this class. Flags can be used to filter which associated resources are displayed.

The command output is similar to that of `lsrsrc`. Resources associated with a source resource are displayed with their class name and one attribute per line to facilitate searching and filtering the output.

Parameters

source_class_name

Specifies the source class in the association.

property_list

Specifies one or more property names. Only these properties (or attributes, in RMC terminology) of associated resources are displayed. If you do not specify this parameter, all property names are displayed.

Flags

-s *source_selection_string*

Specifies that only resources of the source class that match the selection string are used in the search for associated resources.

-S *destination_selection_string*

Specifies that only resources of the associated classes that match this selection string are displayed.

-c *association_class*

Limits the association search to only those resources tied to the source class through *association_class*.

-d *association_endpoint*

Limits the search of associated resources to just the members of this class.

-o *role*

The CIM association interface defines the *role* parameter as the name of the property referring to the class on the source side of the association. Typical values for this parameter are "GroupComponent" or "PartComponent", though the specific name must come from the association class definition.

-R *result_role*

Used like the -o flag, except this is the name of the property that refers to the destination side of the association.

-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 to standard output.

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 are written to standard output.

Standard error

When the -T flag is specified, this command trace messages are written to standard error.

Exit status

0

The command ran successfully.

1

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

2

An incorrect flag was specified on the command line.

- 3 An incorrect parameter was specified on the command line.
- 4 The source endpoint class was not found.
- 5 The destination endpoint class was not found.
- 6 The association class was not found.

Implementation specifics

This command is part of the `rsct.exp.cimim` fileset, in the `rsct.exp` package on the AIX Expansion Pack and Reliable Scalable Cluster Technology (RSCT) package for the Linux operating system.

Location

Item	Description
<code>/opt/rsct/bin/lrsrscassoc</code>	

Examples

To view instances of `cimv2.IBMAIX_UnixProcess` (for AIX) and `cimv2.Linux_UnixProcess` (for Linux) that are associated with `cimv2.IBMAIX_OperatingSystem` and `cimv2.Linux_OperatingSystem` respectively on the specified node, enter:

For AIX:

```
lrsrscassoc -c cimv2.IBMAIX_OSProcess -s 'Name=~"c175nf14"' -S \
'Name=~"emacs"' cimv2.IBMAIX_OperatingSystem Handle Parameters
```

For Linux:

```
lrsrscassoc -c
cimv2.Linux_OSProcess -s 'Name=~"c175nf14"' -S \
'Name=~"emacs"' cimv2.Linux_OperatingSystem Handle Parameters
```

In these examples:

- `-c cimv2.IBMAIX_OSProcess` and `-c cimv2.Linux_OSProcess` are the association classes whose provider is used.
- `-s 'Name=~"c175nf14"'` is the selection string against the `cimv2.IBMAIX_OperatingSystem` and `cimv2.Linux_OperatingSystem` instances (we only want objects associated with the OS instance representing the node `c175nf14`).
- `-S 'Name=~"emacs"'` is the selection string against `cimv2.IBMAIX_UnixProcess` and `cimv2.Linux_UnixProcess` objects; only those with `Name` attributes that contain the pattern `emacs` are returned.
- `cimv2.IBMAIX_OperatingSystem` and `cimv2.Linux_OperatingSystem`, which are the "source object" parameter, are one of the classes in the association.
- `Handle Parameters` are properties that the provider is asked to return. `Handle` is the PID of the process; `Parameters` is a list of arguments to the process.

The following output is displayed:

```
Resource Persistent Attributes for cimv2.IBMAIX_UnixProcess (or cimv2.Linux_UnixProcess)
resource 1:
Handle = "2781"
Parameters = {"emacs", "-u", "foo.C"}
resource 2:
Handle = "2782"
Parameters = {"emacs", "bar.C"}
```

```

resource 3:
Handle = "2783"
Parameters = {"emacs", "foo_bar.C"}
resource 4:
Handle = "2784"
Parameters = {"emacs", "bar_foo.C"}
resource 5:
Handle = "2785"
Parameters = {"emacs", "CIMRC.C"}
resource 6:
Handle = "26994"
Parameters = {"emacs", "lsassocmap.pl"}

```

lsrsrcdef Command

Purpose

Displays definition information for a resource or a resource class.

Syntax

For a *resource*...

To display the definition:

```
lsrsrcdef [-p property] [-e] [-s] [-l | -i | -t | -d | -D delimiter] [-x] [-h] [-TV] resource_class
[attr...]
```

To display the persistent attribute definitions:

```
lsrsrcdef -A p [-p property] [-e] [-s] [-l | -i | -t | -d | -D delimiter] [-x] [-h] [-TV]
resource_class [attr...]
```

To display the dynamic attribute definitions:

```
lsrsrcdef -A d [-p property] [-e] [-s] [-l | -i | -t | -d | -D delimiter] [-x] [-h] [-TV]
resource_class [attr...]
```

For a *resource class*...

To display the definition:

```
lsrsrcdef -c [-p property] [-e] [-s] [-l | -i | -t | -d | -D delimiter] [-x] [-h] [-TV] resource_class
[attr...]
```

To display the persistent attribute definitions:

```
lsrsrcdef -c -A p [-p property] [-e] [-s] [-l | -i | -t | -d | -D delimiter] [-x] [-h] [-TV]
resource_class [attr...]
```

To display the dynamic attribute definitions:

```
lsrsrcdef -c -A d [-p property] [-e] [-s] [-l | -i | -t | -d | -D delimiter] [-x] [-h] [-TV]
resource_class [attr...]
```

To display a list of all of the resource class names:

```
lsrsrcdef
```

Description

The `lsrsrcdef` command displays the definition of a resource or a resource class or the persistent or dynamic attribute definitions of a resource or a resource class. By default:

- if no *attr* parameters are specified on the command line, this command displays the definitions for public attributes. To override this default, use the `-p` flag or specify the name of the attribute you want to display.

- this command does not display attribute descriptions. To display attribute definitions and descriptions, specify the `-e` flag.

Flags

-A p | d

Specifies the attribute type. You can display either persistent or dynamic attribute definitions. Use this flag with the `-c` flag to display the persistent or dynamic attribute definitions of a resource class.

p

Displays only persistent attributes

d

Displays only dynamic attributes

-c

Displays the definition of a resource class definition. To display the persistent attribute definitions for a resource class, specify this flag with the `-A p` flag. To display the dynamic attribute definitions for a resource class, specify this flag with the `-A d` flag.

-d

Specifies delimiter-formatted output. The default delimiter is a colon (:). Use the `-D` flag to change the default delimiter.

-D delimiter

Specifies delimiter-formatted output that uses the specified delimiter. Use this flag to specify something 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.

-e

Specifies expanded format. By default, the descriptions of the definitions are not displayed. Specify this flag to display the definitions and the descriptions.

-i

Generates a template of *resource_data_input_file* that can then, after appropriate editing, be used as input to the `mkrsrc` command. The output is displayed in long (stanza) format. All required and optional attributes that can be used to define a resource are displayed. The attribute data type is displayed as the value in the *attr=value* pairs. It is suggested that when you use this flag, the output of the `lsrsrcdef` command be directed to a file. This flag overrides the `-s` and `-A d` flags.

-l

Specifies "long" format — one entry per line. This is the default display format. If the `lsrsrcdef -l` command is issued without a resource class name, this flag is ignored when the command returns the list of defined resource class names.

-p property

Displays attribute definitions for attributes with the specified *property*. By default, only the definitions for public attributes are displayed. To display all attribute definitions regardless of the property, use the `-p 0` flag.

Persistent attribute properties:

0x0001

read_only

0x0002

reqd_for_define (required)

0x0004

inval_for_define (not valid)

0x0008

option_for_define (optional)

0x0010

selectable

0x0020
public

Dynamic attribute properties:

0x0020
public

A decimal or hexadecimal value can be specified for the property. To request the attribute definitions for all attributes that have one or more properties, "OR" the properties of interest together and then specify the "OR"ed value with the -p flag. For example, to request the attribute definitions for all persistent attributes that are either reqd_for_define or option_for_define, enter:

```
lsrsrdef -p 0x0a
```

- s**
Displays the structured data definition. Specify this flag for the structured data definition to be expanded so that each element definition of the structured data attributes is displayed.
- 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

resource_class

Specifies the name of the resource class with the attribute definitions you want to display.

attr

If a *resource_class* parameter is specified, zero or more attribute names can be specified. If no *attr* parameter is specified, the definition for all of the attributes for the resource are displayed. Specify individual attribute names to control which attributes are displayed and their order. Specify only persistent attribute names when the -A p flag is used. Specify only dynamic attribute names when the -A d flag is used. Attributes must be separated by spaces.

Security

The user needs write permission for the *resource_class* specified in `lsrsrdef` to run `lsrsrdef`. Permissions are specified in the access control list (ACL) file on the contacted system. See *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. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

1. To display the names of all of the resource classes defined on the system, enter:

```
lsrsrcdef
```

The output will look like this:

```
class_name  
"IBM.ATMDevice"
```

```
"IBM.Association"  
"IBM.AuditLog"  
"IBM.AuditLogTemplate"  
"IBM.Condition"  
"IBM.EthernetDevice"  
"IBM.EventResponse"  
...
```

2. To display the resource class definitions for resource IBM.Host, enter:

```
lsrsrcdef -c IBM.Host
```

The output will look like this:

```
Resource Class Definition for: IBM.Host  
resource class 1:  
  class_name      = "IBM.Host"  
  class_id       = 8  
  properties     = {"has_rsrc_insts", "mtype_subdivided"}  
  display_name   = ""  
  description    = ""  
  locator       = "NodeList"  
  class_pattr_count = 1  
  class_dattr_count = 3  
  class_action_count = 0  
  pattr_count    = 6  
  dattr_count   = 47  
  action_count  = 0  
  error_count   = 0  
  rsrc_mgr_count = 1  
rsrc_mgrs 1:  
  mgr_name = "IBM.HostRM"  
  first_key = 1  
  last_key = 1
```

3. To display the resource class persistent attribute definitions for resource IBM.Host, enter:

```
lsrsrcdef -c -A p -p 0 IBM.Host
```

The output will look like this:

```
Resource Class Persistent Attribute Definitions for: IBM.Host  
attribute 1:  
  program_name      = "Variety"  
  display_name     = ""  
  group_name       = ""  
  properties       = {"read_only", "inval_for_define"}  
  description      = ""  
  attribute_id     = 0  
  group_id        = 255  
  data_type       = "uint32"  
  variety_list    = {{1..1}}  
  variety_count   = 1  
  default_value   = 0
```

4. To display the resource persistent attribute definitions and descriptions for resource IBM.Host, enter:

```
lsrsrcdef -A p -p 0 -e IBM.Host
```

The output will look like this:

```
Resource Persistent Attribute Definitions for: IBM.Host  
attribute 1:  
  program_name      = "Name"  
  display_name     = "Name"  
  group_name       = "General"  
  properties       = {"reqd_for_define", "public", "selectable"}  
  description      = "Identifies the current name of the host  
as returned by command."  
  attribute_id     = 0  
  group_id        = 0  
  data_type       = "char_ptr"  
  variety_list    = {{1..1}}  
  variety_count   = 1  
  default_value   = ""  
attribute 2:
```

```

program_name      = "ResourceHandle"
display_name     = "Resource Handle"
group_name       = "Internal"
properties       = {"read_only","inval_for_define","selectable"}
description      = "A globally unique handle that identifies the host.
                  Every resource is assigned a resource handle,
                  which is used internally for identifying and
                  locating each resource. The resource handle
                  is fixed in size and avoids the problems of
                  name space collisions across different types
                  of resources."

attribute_id      = 1
group_id         = 255
data_type        = "rsrc_handle_ptr"
variety_list     = {{1..1}}
variety_count    = 1
default_value    = "0x0000 0x0000 0x00000000 0x00000000 0x00000000 0x00000000"
attribute 3:
program_name     = "Variety"
display_name     = "Variety"
group_name       = "Internal"
...

```

5. To display the public dynamic attributes for resource IBM.Host, enter:

```
lsrsrcdef -A d IBM.Host
```

The output will look like this:

```

Resource Dynamic Attribute Definitions for: IBM.Host
attribute 1:
program_name      = "ProcRunQueue"
display_name     = ""
group_name       = ""
properties       = {"public"}
description      = ""
attribute_id     = 1
group_id        = 1
data_type        = "float64"
variable_type    = 0
variety_list     = {{1..1}}
variety_count    = 1
init_value       = 0
min_value        = 0
max_value        = 100
expression       = "(ProcRunQueue - ProcRunQueue@P) >= (ProcRunQueue@P * 0.5)"
expression_description = ""
rearm_expression = "ProcRunQueue < 50"
rearm_description = ""
PTX_name         = ""
attribute 2:
...

```

Location

`/opt/rsct/bin/lsrsrcdef`

lssavevg Command

Purpose

Lists or restores the contents of a volume group backup on a specified media.

Syntax

```
lssavevg [ -b blocks ] [ -f device ] [ -a ] [ -c ] [ -l ] [ -n ] [ -r ] [ -s ] [ -d path ] [ -B ] [ -D ] [ -L ] [ -V ] [ file_list ]
```

Description

The **lssavevg** command lists the contents of a volume group backup from tape, file, CD-ROM, or other source and can be used to restore files from a valid backup source. The **lssavevg** command also works for multi-volume backups such as multiple CDs, DVDs, USB disks, or tapes.

The **lssavevg -r** and **restorevgfiles** commands perform identical operations and must be considered interchangeable.

Flags

Item	Description
-a	Verifies the physical block size of the tape backup, as specified by the -b block flag. You might need to alter the block size if necessary to read the backup. The -a flag is valid only when a tape backup is used.
-b blocks	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 defaults to 100.
-B	Prints the volume group backup log to stdout . This flag displays the past 256 backups (roughly). The log is in alog format and is kept in /var/adm/ras/vgbackuplog . Each line of the log is a semicolon-separated list of the file or device name, the command that is used to make backup, date, shrink size, full size of the backup, and recommended maintenance or technology level (if any). Note: The shrink size is the size of the data on all file systems. The full size is total size of each file system (unused + data).
-c	Produces colon-separated output. This flag works only with the -l and -L flags.
-d path	Specifies the directory path to which the files are 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.
-f device	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 .
-l	Displays useful information about a volume group backup. This flag requires the -f device flag. This flag causes lssavevg to display information such as volume group, date and time backup was made, uname output from backed up system, oslevel, recommended maintenance or technology level, backup size in megabytes, and backup shrink size in megabytes. The shrink size is the size of the data on all file systems. The full size is the total size of each file system (unused + data). The -l flag also displays the logical volume and file system information of the backed up volume group, equivalent to running " lsvg -l vgname ".
-L	Displays lpp fileset information about a mksysb backup only. This flag requires the -f device flag and displays the equivalent information to that produced by invoking " lslpp -l " on the running backed up system. This flag does not produce output about any volume group backup other than that produced by mksysb .
-n	Does not restore ACLs, PCLs, or extended attributes

Item	Description
-r	Specifies to restore the backup files, as defined by the <i>file-list</i> parameter. If the <i>file-list</i> parameter is not specified, then all files in the backup are restored. If the -r flag is not used, then executing the lssavevg command lists only the files in the specified backup.
-s	Specifies that the backup source is a user volume group and not rootvg.
-V	Verifies a tape backup. This flag requires the -f device flag and works for tape devices only. The -V flag causes lssavevg to verify the readability of the header of each file on the volume group backup and print any errors that occur to stderr .

Parameters

Item	Description
<i>file_list</i>	Identifies the list of files to be restored. This parameter is used only when the -r flag is specified. The full path of the files relative to the current directory must be specified in the space-separated list. All files in the specified directory are 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 list the contents of the system backup that is on the default device **/dev/rmt0**, enter the following command:

```
lssavevg
```

2. To list the contents of the system backup that is on device **/dev/cd1**, enter the following command:

```
lssavevg -f /dev/cd1
```

3. To list the contents of the system backup that is on device **/dev/cd1**, which is a user volume group that is not rootvg, enter the following command:

```
lssavevg -f /dev/cd1 -s
```

4. To restore **/etc/filesystems** from the system backup that is on device **/dev/cd1**, enter the following command:

```
lssavevg -f /dev/cd1 -r ./etc/filesystems
```

5. To restore all files in the **/myfs/test** directory of the non-rootvg backup, which is on device **/dev/cd1**, and write the restored files to **/data/myfiles**, enter the following command:

```
lssavevg -f /dev/cd1 -r -s -d /data/myfiles ./myfs/test
```

6. To display colon-separated lpp information about a **mksysb** backup tape that is on device **/dev/rmt0**, enter the following command:

```
lssavevg -Lc -f /dev/rmt0
```

7. To display the volume group backup log to **stdout**, enter the following command:

```
lssavevg -B
```

8. To list volume group and general backup data about a backup that is on **/tmp/mybackup**, enter the following command:

```
lssavevg -l -f /tmp/mybackup
```

9. To verify the readability of each header on a volume group backup tape in **/dev/rmt0**, enter the following command:

```
lssavevg -V -f /dev/rmt0
```

10. To list the contents of the system backup that is on device **/dev/usbms0**, enter the following command:

```
lssavevg -f /dev/usbms0
```

Files

Item	Description
/usr/bin/lssavevg	Contains the lssavevg command

lssavewpar Command

Purpose

Lists the contents of a workload partition backup on a specified media.

Syntax

```
lssavewpar [ -b blocks ] [ -f device ] [ -a ] [ -c ] [ -D ] [ -l | -L | -M | -N ] [ -V ]
```

Description

The **lssavewpar** command lists the contents of a workload partition backup from tape, file, CD, USB flash drive, or DVD.

Flags

Item	Description
-a	Verifies the physical block size of the tape backup, as specified by the -b flag. You might need to alter the block size if necessary to read the backup. The -a flag is valid only when a tape backup is used.
-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 to read is 100, which is the default. The -b flag is valid only when a tape backup is used.
-c	Produces colon-separated output. Specify the -c flag only with the -l , -L , -M , and -N flags.
-D	Produces the output of debugging.
-f <i>device</i>	Specifies the device that contains the backup (file, tape, CD, or other sources) as defined by the <i>device</i> parameter. If you do not specify the -f flag, the default device name is /dev/rmt0 .

Item	Description
-l	<p>Displays information about a workload partition backup.</p> <p>Displays information including the volume group, the date and time that the backup was made, the uname output from the backed up system, the operating system level, the maintenance or technology level, the backup size in megabytes, and the backup-shrink size in megabytes. The shrink size is the size of the data on all file systems. The full size is the total size of each file system (unused and data). The -l flag also displays the logical volume and file system information of the backed up workload partition. You must specify the -f flag when you specify the -l flag. The -l flag is mutually exclusive with the -L, -M, and -N flags.</p>
-L	<p>Displays lpp file-set information about a workload partition backup only.</p> <p>When you specify the -L flag, you must also specify the -f device flag. The -L flag is mutually exclusive with the -l, -M, and -N flags.</p>
-M	<p>Lists information about any writable namefs-mounted file systems included in the backup. The information is displayed in the following form:</p> <pre>Mount_Device Blocks Blocks_Free Blocks_Used Mount_Point</pre> <p>The Blocks and Blocks_Free fields describe the number of 512-byte blocks and the free 512-byte blocks that are present in the mounted file system. The Blocks_Used describes the number of 512-byte blocks used in the portion of the mounted file system mounted from the WPAR. The -M flag is mutually exclusive with the -l, -L, and -N flags.</p>
-N	<p>Lists information about any NFS-mounted file systems included in the backup. The information is of the following form:</p> <pre>RemoteHost HostFilesystem Blocks Blocks_Free Blocks_Used</pre> <p>The Blocks and Blocks_Free describe the number of 512-byte blocks and the free 512-byte blocks in the remote file system. The Blocks_Used describes the number of 512-byte blocks used in the portion of the remote file system mounted from the WPAR. The -N flag is mutually exclusive with the -l, -L, and -M flags.</p>
-v	<p>Verifies a tape backup.</p> <p>You must specify the -f flag with the -v flag. The flag is valid only for tape devices. The -v flag verifies the readability of the header of each file on the volume group backup and prints any errors that occur to the stderr file.</p>

Examples

1. To list the contents of the workload partition backup that is located on the default device **/dev/rmt0**, use the following command:

```
lssavewpar
```

2. To list the contents of the system backup that is located on device **/dev/cd1**, use the following command:

```
lssavewpar -f /dev/cd1
```


3. To display colon-separated lpp information about a workload partition backup tape that is located on **/dev/rmt0**, use the following command:

```
lssavewpar -Lc -f /dev/rmt0
```

4. To list volume group and general backup data about a backup located at `/tmp/mybackup`, use the following command:

```
lssavewpar -l -f /tmp/mybackup
```

5. To verify the readability of each header on a workload partition backup tape in **/dev/rmt0**, use the following command:

```
lssavewpar -V -f /dev/rmt0
```

6. To list the contents of the system backup located on device **/dev/usbms0**, use the following command:

```
lssavewpar -f /dev/usbms0
```

Issec Command

Purpose

Lists attributes in the security stanza files.

Syntax

```
Issec [ -c ] [ -f File ] [ -s Stanza ] [ -a Attribute ... ]
```

Description

The **Issec** command lists attributes stored in the security configuration stanza files. The following security configuration files contain attributes that you can specify with the *Attribute* parameter:

- **/etc/security/envIRON**
- **/etc/security/gROUp**
- **/etc/security/audit/hosts**
- **/etc/security/lastlog**
- **/etc/security/limits**
- **/etc/security/login.cfg**
- **/usr/lib/security/mkuser.default**
- **/etc/nscontrol.conf**
- **/etc/security/passwd**
- **/etc/security/portlog**
- **/etc/security/pwdalg.cfg**
- **/etc/security/roles**
- **/etc/security/smitacl.user**
- **/etc/security/smitacl.group**
- **/etc/security/user**
- **/etc/security/user.roles**
- **/etc/security/rtc/rtcd_policy.conf**
- **>/etc/security/uattr|<**

When listing attributes in the **/etc/security/environ**, **/etc/security/lastlog**, **/etc/security/limits**, **/etc/security/passwd**, **/etc/security/uattr**, and **/etc/security/user** files, the stanza name specified by the *Stanza* parameter must be either a valid user name or default. When listing attributes in the **/etc/security/group** file, the stanza name specified by the *Stanza* parameter must be either a valid group name or default. When listing attributes in the **/usr/lib/security/mkuser.default** file, the *Stanza* parameter must be either admin or user. When listing attributes in the **/etc/security/portlog** file, the *Stanza* parameter must be a valid port name. When listing attributes in the **/etc/security/login.cfg** file, the *Stanza* parameter must be either a valid port name, a method name, or the **usw** attribute.

You cannot list the **password** attribute of the **/etc/security/passwd** file with the **lssec** command.

Only the root user or a user with PasswdAdmin authorization can list the lastupdate and flags attributes for administrative users.

Flags

Item	Description
-c	Specifies that the output should be in colon-separated format.
-f File	Specifies the name of the stanza file to list.
-s Stanza	Specifies the name of the stanza to list.
-a Attribute	Specifies the attribute to list.

Security

Access Control: This command grants execute access only to the root user and the security group. The command has the trusted computing base attribute and runs the **setuid** subroutine for the root user to access the security databases.

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** authorization.

Examples

1. To list the number of unsuccessful login attempts by the root user since the last successful login of the root user, enter:

```
lssec -f /etc/security/lastlog -s root -a unsuccessful_login_count
```

The system displays the result as follows:

```
root unsuccessful_login_count=15
```

2. To list the times that logins are allowed on the **/dev/tty2** port, enter:

```
lssec -f /etc/security/login.cfg -s /dev/tty2 -a logintimes
```

The system displays the result as follows:

```
/dev/tty0 logintimes=!january1,!july4,!december25
```

3. To list the default setting for the **tpath** attribute and the **ttys** attribute in colon format,
4. enter:

```
lssec -c -f /etc/security/user -s default -a tpath -a ttys
```

The system displays the result as follows:

```
#name:tpath:ttys
default:nosak:ALL
```

Files

MItem	Description
o	
d	
e	
 /usr/bin/lissec	Specifies the path to the lissec command.
r /etc/security/environ	Contains the environment attributes of users.
r /etc/security/group	Contains extended attributes of groups.
r /etc/security/audit/hosts	Contains host and processor IDs.
r /etc/security/lastlog	Defines the last login attributes for users.
r /etc/security/limits	Defines resource quotas and limits for each user.
r /etc/security/login.cfg	Contains port configuration information.
r /usr/lib/security/ mkuser.default	Contains the defaults values for new users.
r /etc/nscontrol.conf	Contains configuration information of some name services.
r /etc/security/passwd	Contains password information.
r /etc/security/portlog	Contains unsuccessful login attempt information for each port.
r /etc/security/pwdalg.cfg	Contains configuration information for loadable password algorithms (LPA).
r /etc/security/roles	Contains a list of valid roles.
r /etc/security/smitacl.user	Contains user ACL definitions.
r /etc/security/smitacl.group	Contains group ACL definitions.
r /etc/security/user	Contains the extended attributes of users.
r /etc/security/user.roles	Contains a list of roles for each user.
r /etc/security/domains	Contains the valid domain definitions for the system.
r /etc/security/rtc/ wrtcd_policy.conf	Contains configuration information for the rtcd daemon
r /etc/security/uattr	Contains further user attributes. <<<
r	

Issecattr Command

Purpose

Displays the security attributes of a command, a device, a privileged file, a process or, a domain-assigned object.

Syntax

```
Issecattr [-R load_module] { -c | -d | -p [-h] [-A] | -f | -o } [-C | -F] [-a List] { ALL | Name [,Name] ... }
```

Description

The **lssecattr** command lists the security attributes of one or more commands, devices, or processes. The command interprets the *Name* parameter as either a command, a device, a privileged file, a process, or a domain-assigned object based on whether the **-c** (command), **-d** (device), **-f** (privileged file), **-p** (process), or **-o** (domain-assigned object) flag is specified. If the **-c** flag is specified, the *Name* parameter must include the full path to the commands. If the **-d** flag is specified, the *Name* parameter must include the full path to the devices. If the **-f** flag is specified, the *Name* parameter must include the full path to the file. If the **-p** flag is specified, the *Name* parameter must be the numeric process identifier (PID) of an active process on the system. If the **-o** flag is specified, the *Name* parameter must be the full path if it is a file or device and for port or port ranges it must be prefixed with TCP_ or UDP_. Use the **ALL** keyword to list the security attributes for all commands, devices, files, or processes. By default, the **lssecattr** command displays all of the security attributes for the specified object. To view the selected attributes, use the **-a List** flag.

If the system is configured to use databases from multiple domains, the privileged commands, privileged devices, and privileged files, as specified by the *Name* parameter, are searched from the domains in the order specified by the **secorder** attribute of the corresponding database stanza in the **/etc/nscontrol.conf** file. If duplicate entries exist in multiple domains, only the first entry instance is listed. Use the **-R** flag to list the objects from a specific domain.

By default, the **lssecattr** command lists the security attributes on one line. It displays the attribute information as the definitions of **Attribute=Value**, each separated by a blank space. To list the attributes in stanza format, use the **-F** flag. To list the attributes as colon-separated records, use the **-C** flag.

Flags

Item	Description
-a List	Lists the attributes to display. The <i>List</i> variable requires a blank space between attributes to list multiple attributes. If you specify an empty list, only the object names are displayed. The attributes that can be listed in the <i>List</i> variable are dependent on which one of the -c , -d , and -p flags is specified. For a list of the valid attribute names for each flag, see the setsecattr command.
-A	Display the list of authorizations used by a specified process. This flag can only be used with the -p flag.
-c	The <i>Name</i> parameter specifies the full paths to one or more commands on the system that have entries in the /etc/security/privcmds privileged command database.
-C	Displays the privileged security attributes in colon-separated records as follows: <pre>#name:attribute1:attribute2: ... name:value1:value2: ... name:value1:value2: ...</pre> The output is preceded by a comment line that has details about the attribute represented in each colon-separated field. If the -a flag is specified, the order of the attributes matches the order specified in the -a flag. If an object does not have a value for a given attribute, the field is still output but is empty. The last field in each entry is terminated by a newline character rather than a colon.
-d	The <i>Name</i> parameter specifies the full paths to one or more devices on the system that have entries in the /etc/security/privdevs privileged device database.

Item	Description
-f	The <i>Name</i> parameter specifies the full paths to one or more files on the system that have entries in the /etc/security/privfiles privileged files database.
-F	Displays the output in stanza format, with each stanza identified by an object name. Each pair of Attribute=Value is listed on a separate line: <pre style="background-color: #f0f0f0; padding: 10px;">Name : attribute1=value attribute2=value attribute3=value</pre>
-h	Displays the full hierarchy of privileges for the process. By default, only the highest level of privilege is listed.
-o	The <i>Name</i> parameter specifies one of the following entries in the /etc/security/domobjs domain-assigned object database. <ul style="list-style-type: none"> • the full paths to one or more devices/files on the system • the port or port ranges prefixed with TCP_ or UDP_ • the network interfaces
-p	The <i>Name</i> parameter specifies the numeric process identifiers (PID) of one or more active processes on the system. The -p flag cannot be listed with the -R flag as they are mutually exclusive.
-R load_module	Specifies the loadable module to query the <i>Name</i> entry from.

Parameters

Item	Description
ALL	For all commands, devices or processes.
<i>Name</i>	The object to modify. The <i>Name</i> parameter is interpreted according to which one of the -c , -d , -p , and -o flags is specified.

Security

The **lssecattr** command is a privileged command. It is owned by the root user and the security group, with mode set to 755. You must assume a role with at least one of the following authorizations to run the command successfully.

Item	Description
aix.security.cmd.list	Required to list the attributes of a command with the -c flag.
aix.security.device.list	Required to list the attributes of a device with the -d flag.
aix.security.file.list	Required to list the attributes of a file with the -f flag.
aix.security.proc.list	Required to list the attributes of a process with the -p flag.
aix.security.dobject.list	Required to list the attributes of a domain-assigned object with the -o flag.

File Accessed

Item	Description
File	Mode
<code>/etc/security/privcmds</code>	r
<code>/etc/security/privdevs</code>	r
<code>/etc/security/privfiles</code>	r
<code>/etc/security/domobjs</code>	r

Examples

1. To display the access authorization and the innate privileges of the `/usr/sbin/mount` command, enter the following command:

```
lssecattr -c -a accessauths innateprivs /usr/sbin/mount
```

2. To display all the security attributes of the `/dev/mydev` device, enter the following command:

```
lssecattr -d /dev/mydev
```

3. To display all the security attributes of the `/dev/mydev` device in LDAP, enter the following command:

```
lssecattr -R LDAP -d /dev/mydev
```

4. To display the privileges for the effective and used privilege sets of two processes in a colon format, enter the following command:

```
lssecattr -p -C -a eprivs uprivs 38483,57382
```

5. To display the read authorization list of the `/etc/security/user` file, enter the following command:

```
lssecattr -f -a readauths /etc/security/user
```

6. To display the used authorizations for a process in a stanza format, enter the following command:

```
lssecattr -F -p -A 34890
```

7. To display all the domain attributes of the `/dev/dev1` device, enter the following command:

```
lssecattr -o /dev/dev1
```

8. To display all the domain attributes of the network interface `en0` device, enter the following command:

```
lssecattr -o en0
```

lssecmode Command

Purpose

Displays the current or pending security mode configuration and key types in a formatted output.

Syntax

```
lssecmode [ -p ] [ -d | -D delim ] [ -x ] [ -T ] [ -V ] [ -h ]
```

Description

The **lssecmode** command displays information about the current or pending security mode configuration and key types. This information consists of the compliance mode, public or private key type, and default symmetric key types.

Note: If no flag is specified, the current security configuration mode and key types are displayed.

Flags

Item	Description
-d	Displays the delimiter-formatted output. The default delimiter is a colon (:). You can use the -D flag to change the default delimiter.
-D delim	Specifies the delimiter to be used in the formatted output. By default, the colon (:) character is used as the delimiter in the output. You can use this flag to format the output with another delimiter that can contain one or more characters. For example, if the output data already contains colons, the default delimiter might result in confusion.
-h	Displays the command usage.
-p	Displays the pending security mode configuration and key types.
-T	Writes the command trace messages to standard output.
-V	Writes the command verbose messages to standard output.
-x	Specifies that the header information must not be displayed.

Exit status

0	Successful.
1	Missing argument error.
2	Invalid option error.
3	API error.

Examples

1. To display the current security configuration mode and key types, enter the following command:

```
# lssecmode
```

An output similar to the following example is displayed:

```
Current Security Mode Configuration
Compliance Mode : none
Asymmetric Key Type : rsa512
Symmetric Key Type : default
```

2. To display the pending security configuration mode and key types, enter the following command:

```
# lssecmode -p
```

If there is no pending security configuration mode and if the staging file is not present, the following output is displayed:

```
2650-384 There are no pending configuration available
```

3. To format the output with a delimiter `::`, enter the following command:

```
# lssecmode -D "::"
```

An output similar to the following example is displayed:

```
Current Security Mode Configuration
Compliance Mode :: none
Asymmetric Key Type :: rsa512
Symmetric Key Type :: default
```

4. To display the pending security configuration mode and key types with command verbose messages and without header information, enter the following command:

```
# lssecmode -p -x -V -D "::" -T
```

An output similar to the following example is displayed:

```
Invoked with parameters: -p -x -V -D :: -T
Invoking lssecmode to get the pending security mode and key types....
No header information required..
Compliance Mode :: none
Asymmetric Key Type :: rsa512
Symmetric Key Type :: default
Checking lssecmode log file size and backup if necessary....
No log file exist. No backup is needed
```

Location

/opt/rsct/bin/lssecmode

Contains the **lssecmode** command.

lssensor Command

Purpose

Displays information about sensors and microprocessors that are defined to the resource monitoring and control (RMC) subsystem.

Syntax

```
lssensor [-m] [-a | -n host1[,host2...]] [-N { node_file "-" }] [-l | -t | -d | -D delimiter] [-x] [-h] [-v | -V] [-A  
| sensor_name1 [sensor_name2...]]
```

Description

The **lssensor** command displays the attributes of one or more sensors. If you do not specify any *name* parameters, the **lssensor** command lists the names of all of the sensors. Use the **-A** flag to list all of the sensors and all of their attributes and values. Use the **-m** flag to display information about microprocessors.

The **lssensor** command displays values for attributes that you can set using a sensor command or a microprocessor module, if the attributes are monitored. If the attributes are not monitored, **lssensor** does not display their values. A sensor command is a command or script that the sensor resource manager runs to set and update a sensor's attribute values. A microprocessor module is a loadable module that the microprocessor resource manager runs to set and update a microprocessor's attribute values.

Use the **-l**, **-t**, **-d**, or **-D** flags to display the output in long format, table format, or delimiter format. The **-x** flag omits headings when any of these flags are used.

The **lssensor** command runs on any node. If you want **lssensor** to run on all of the nodes in a domain, use the **-a** flag. If you want **lssensor** 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*.

The **lssensor** command lists the following information about defined sensors:

Field	Description
Name	The name of the sensor.
Command	The command that is run to update the sensor attributes
ConfigChanged	Information about changes to access or to persistent attributes
ControlFlags	Indicates whether any special handling is required for this sensor
Description	This field is not used
ErrorExitValue	Indicates how the exit value is interpreted by the sensor resource manager
ErrorMessage	This field is not used
ExitValue	The exit code from the command that is running
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
MonitorStatus	This attribute is set to 1 when certain sensor attributes are being monitored
NodeNameList	The name of the node where the sensor resource is defined
RefreshInterval	The interval (in seconds) during which the sensor attribute values are updated when the sensor command is run
SavedData	An output string from the sensor command
SD	Contains all dynamic resource attributes except ConfigChanged, Quantum, and ExitValue as its elements
String	The type string attribute for this sensor resource
TimeCommandRun	Indicates the date and time that the sensor command was run
Uint32	The type uint32 attribute for this sensor resource
Uint64	The type uint64 attribute for this sensor resource
UserName	The user ID that is used when run the sensor command is run

The **lssensor** command displays the following information about defined microsensors:

Field	Description
Name	The name of the microsensor.
ActivePeerDomain	The peer domain for which information is being displayed.
Arguments	The arguments for this microsensor resource.

Field	Description
ConfigChanged	Information about changes to persistent attributes or to access.
CustomDynamicAttributes	The custom dynamic attributes for this microsensor resource.
Description	Information about the microsensor and what it monitors.
Float32	The type float32 attribute for this microsensor resource.
Float32Array	The type float32 array attribute for this microsensor resource.
Float64	The type float64 attribute for this microsensor resource.
Float64Array	The type float64 array attribute for this microsensor resource.
Int32	The type int32 attribute for this microsensor resource.
Int32Array	The type int32 array attribute for this microsensor resource.
Int64	The type int64 attribute for this microsensor resource.
Int64Array	The type int64 array attribute for this microsensor resource.
LastQueryRC	The return code from the microsensor module from the last time the microsensor was called for an attribute of the microsensor resource.
LastQueryTime	The time of LastQueryRC.
ModuleName	The path name to the loadable microsensor module.
MonitorStatus	This attribute is set to 1 when any of the other microsensor attributes is being monitored.
NodeNameList	The name of the node where this microsensor is defined.
RefreshInterval	The interval (in seconds) during which the microsensor attribute values are updated when the microsensor callback is called.
String	The type string attribute for this microsensor resource.
StringArray	The type string array attribute for this microsensor resource.
UInt32	The type uint32 attribute for this microsensor resource.
UInt32Array	The type uint32 array attribute for this microsensor resource.
UInt64	The type uint64 attribute for this microsensor resource.
UInt64Array	The type uint64 array attribute for this microsensor resource.

Flags

-a

Lists 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, `lssensor -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.

-A

Displays all of the sensors with their attributes and values.

- 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 something 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 that the information be displayed in "long" format. Each attribute is displayed on a separate line.
- m**
Specifies that information about microsensors will be displayed.
- n *host1[,host2...]***
Specifies the node from which the sensor should be listed. By default, the sensor is listed from 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 to the left of a node name are ignored
 - Any characters to the right of a node name are ignored
 Use **-N "-"** in a management domain or a peer domain to read the node names from standard input.
- t**
Specifies table format. Each attribute is displayed in a separate column, with one sensor resource per line.
- x**
Suppresses header printing when **-l**, **-t**, **-d**, or **-D** is specified.
- h**
Writes the command's usage statement to standard output.
- v | -V**
Writes the command's verbose messages to standard output.

Parameters

- sensor_name1* [*sensor_name2...*]**
Specifies the names of one or more sensors to display.

Security

To display sensor information using this command, you need read permission for the **IBM.Sensor** resource class. To display microsensor information using this command, you need read permission for the **IBM.MicroSensor** resource class. Permissions are specified in the access control list (ACL) file on the contacted system. See the *Administering RSCT* 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 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.

Examples

1. To list the names of all of the sensors, enter:

```
lssensor
```

The output will look like this:

```
sensor1
sensor2
sensor3
```

2. To list the names and attributes of all sensors, enter:

```
lssensor -A
```

The output will look like this:

```
Name = sensor1
ActivePeerDomain =
Command = /usr/local/bin/sensorcmd1
```

```

ConfigChanged = 0
ControlFlags = 1
Description =
ErrorExitValue = 1
ExitValue = 0
Float32 = 1.06381e+06
Float64 = 1.06381e+06
Int32 = 1063814
Int64 = 1063814
NodeNameList = {somenode.pok.ibm.com}
RefreshInterval = 60
SavedData = Last SavedData
SD = [string from sensor1,1063814,1063814,1063814,1063814,1.06381e+06,1.06381e+06]
String = string from sensor1
UInt32 = 1063814
UInt64 = 1063814
UserName = root
-----
Name = CFMRootModTime
ActivePeerDomain =
Command = /opt/csm/csmbin/mtime/cfmroot
ConfigChanged = 0
ControlFlags = 0
Description =
ErrorExitValue = 1
ExitValue = 0
Float32 = 0
Float64 = 0
Int32 = 0
Int64 = 0
NodeNameList = {somenode.pok.ibm.com}
RefreshInterval = 60
SavedData =
SD = [,0,0,0,0,0,0]
String =
UInt32 = 0
UInt64 = 0
UserName = root
-----
Name = ErrorLogSensor
ActivePeerDomain =
Command = /opt/csm/csmbin/monerrorlog
ConfigChanged = 0
ControlFlags = 0
Description =
ErrorExitValue = 1
ExitValue = 0
Float32 = 0
Float64 = 0
Int32 = 0
Int64 = 0
NodeNameList = {somenode.pok.ibm.com}
RefreshInterval = 60
SavedData =
SD = [,0,0,0,0,0,0]
String =
UInt32 = 0
UInt64 = 0
UserName = root
-----
.
.
.

```

3. To list the attributes of sensor2, enter:

```
lssensor sensor2
```

The output will look like this:

```

Name = sensor2
Command = /usr/local/bin/sensorcmd2
ConfigChanged = 0
ControlFlags = 0
Description =
ErrorExitValue = 1
ExitValue = 127
Float32 = 0
Float64 = 0
Int32 = 0

```

```
Int64 = 0
NodeNameList = {somenode.pok.ibm.com}
RefreshInterval = 60
SavedData =
SD = [,0,0,0,0,0,0]
String =
Uint32 = 0
Uint64 = 0
UserName = root
```

4. To list all of the sensors' information using delimited output, enter:

```
lssensor -dA
```

The output will look like this:

```
Displaying sensor information:
Name:ActivePeerDomain:Command:ConfigChanged:ControlFlags:Description:ErrorExitValue:ErrorMessage:ExitValue:
:
Float32:Float64:Int32:Int64:MonitorStatus:NodeNameList:RefreshInterval:SD:SavedData:
String:TimeCommandRun:Uint32:Uint64:UserName:
JoeExample:JoeDomain:cat /etc/motd:0:0:1:0:
:::0:{node1.myhost.com}:60:[,0,0,0,0,0,0]::
:Fri Feb 6 19:00:00 2009:::root:
JoeSample:JoeDomain:/opt/rsct/install/bin/ctversion:0:0:1:0:
:::0:{node1.myhost.com}:60:[,0,0,0,0,0,0]::
:Fri Feb 6 19:00:00 2009:::root:
JoeSens:JoeDomain:/tmp/sensor/numusers:0:1:1:0:
:::0:{node1.myhost.com}:0:[,2,0,0,0,0,0]::
:Tue Mar 3 10:27:19 2009:::root:
```

5. To list the names of all of the sensors on the nodes that are listed in the `/u/joe/common_nodes` file, enter:

```
lssensor -N /u/joe/common_nodes
```

where `/u/joe/common_nodes` contains:

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

The output will look like this:

```
sensor1
sensor2
sensor3
```

6. To list the names of all of the microsensors, enter:

```
lssensor -m
```

The output will look like this:

```
IBM.MSensor1
IBM.MSensor2
IBM.MSensor3
```

7. To list the attributes of the microsensor **IBM.MSensor2**, enter:

```
lssensor -m IBM.MSensor2
```

The output will look like this:

```
Name = IBM.MSensor2
ActivePeerDomain =
Arguments = all
ConfigChanged = 0
CustomDynamicAttributes = {[CDA1,19,1,3,0,1],[CDA2,20,2,2,0,1],[CDA3,21,3,2,0,1]}
Description =
Float32 =
Float32Array =
Float64 =
Float64Array =
Int32 = 52
Int32Array = {36, 45, 2, 73}
Int64 =
```

```
Int64Array =
LastQueryRC = 0
LastQueryTime = Tue Mar 31 18:00:00 2009
ModuleName = /usr/slib/msensors/sensor2
MonitorStatus = 0
NodeNameList = {node2.gumby.com}
RefreshInterval = 600
String =
StringArray =
UInt32 =
UInt32Array =
UInt64 =
UInt64Array =
```

Location

`/opt/rsct/bin/lssensor`

lsslot Command

Purpose

Displays dynamically reconfigurable slots, such as hot plug slots, and their characteristics.

Syntax

lsslot **-c** *ConnectorType* [**-a** | **-o** | **-l** *DeviceName* | **-s** *Slot*] [**-F** *Delimiter*]

Description

The **lsslot** command displays all the specified hot plug slots and their characteristics. Hot plug slots are the plug-in points for connecting entities that can be added and removed from the system without turning the system power off or rebooting the operating system. The **-c** flag is required. It specifies the type of hot plug connector, for example, `pci` for hot pluggable PCI adapters. You can display only the empty, that is, available, hot plug slots with the **-a** flag, the occupied slots with the **-o** flag, or a specific slot by using the **-s** flag. The **-l** flag can be used to locate the slot associated with specified *DeviceName*, as listed by the **lsdev** command.

The **lsslot** command is used to list the connectors which are connection points for either physical entities like PCI adapters or logical entities like logical slots or logical host-Ethernet adapter ports. The command can list the following types of connectors:

- `pci`: a physical connector
- `slot`: a logical connector
- `phb`: a logical connector
- `port`: a logical connector

The **-a** and the **-o** flags will be ignored for the logical connectors. The **lsslot** command in the case of the logical connectors displays the logical entities that are currently assigned to the partition, depending upon the connector type specified. When there are multiple slots under a PHB, a logical slot entity can be associated with a logical slot connector. Otherwise, it can be associated with a logical PHB connector. You can run the **lsslot -c slot** command and the **lsslot -c phb** command to view all logical slot entities.

The output of the **lsslot** command is dependent on the *ConnectorType* and the platform on which the command is executed. The characteristics of a slot may include the following:

- Slot name or identification
- Connector type or slot description, for example, a PCI hot plug slot
- Connected device name(s), for example, `scsi0`, `ent0`

When the PHBs are listed using the **lsslot** command, the Device(s) Connected column will display the ODM name of the PHB followed by the ODM names of the devices corresponding to the logical slots underneath the PHB, with all the ODM devices associated with each logical slot displayed on each

separate line under the ODM name of the PHB. In case there is no ODM name for the PHB, a blank line will be displayed.

Flags

Item	Description
-a	Displays available hot plug slots and their characteristics. Available slots are those slots that do not have a hot plug device connected. This flag is ignored for connector types of slot and phb.
-c ConnectorType	Displays the slots of the specified <i>ConnectorType</i> . <i>ConnectorType</i> identifies the type of connector. For example, the <i>ConnectorType</i> for a hot plug PCI slot is pci, for logical slots, it is slot and for PHBs, it is phb. This flag is required.
-F Delimiter	Specifies a single character to delimit the output. The heading is not displayed and the columns are delimited by the <i>Delimiter</i> character.
-l DeviceName	Displays the characteristics of the slot to which <i>DeviceName</i> is associated. The <i>DeviceName</i> is the logical device name of the device connected to the slot, as listed by the lsdev command.
-o	Displays the characteristics of the occupied slots. Occupied slots have a hot plug device connected. This flag is ignored for connector types of slot and phb.
-s Slot	Displays characteristics for the specified <i>Slot</i> . The format of <i>Slot</i> is platform/connector_type dependent.

Examples

1. To list the available PCI hot plug slots, enter:

```
lsslot -c pci -a
```

The system displays a message similar to the following:

Item	Description	Device(s) Connected
Slot name	Description	Device(s) Connected
U0.4-P1-I1	PCI 64 bit, 66MHz, 3.3 volt slot	empty
U0.4-P1-I2	PCI 64 bit, 66MHz, 3.3 volt slot	empty
U0.4-P1-I3	PCI 64 bit, 66MHz, 3.3 volt slot	empty

2. To list the PCI hot plug slot associated with a scsi adapter named scsi1, enter:

```
lsslot -c pci -l scsi1
```

The system displays a message similar to the following:

Item	Description	Device(s) Connected
Slot name	Description	Device(s) Connected
U0.4-P1-I1	PCI 64 bit, 33MHz, 5 volt slot	scsi1

3. To list all the PCI hot plug slots, enter:

```
lsslot -c pci
```

The system displays a message similar to the following:

Item	Description	Device(s) Connected
Slot name	Description	Device(s) Connected
U0.4-P1-I1	PCI 64 bit, 33MHz, 3.3 volt slot	empty
U0.4-P1-I2	PCI 64 bit, 33MHz, 3.3 volt slot	scsi0
U0.4-P1-I3	PCI 64 bit, 33MHz, 3.3 volt slot	unknown
U0.4-P1-I5	PCI 64 bit, 33MHz, 3.3 volt slot	empty

Slots that have *unknown* in the Device(s) Connected column have a device connected to the slot, but the device isn't in the ODM customized device (CuDv) database. This can be due to the device having been newly added but not configured yet, deleted with the **rmdev -d** command, or the system may not be installed with the software packages associated with the device.

4. To list all the PCI Host Bridges that are assigned to the partition, enter:

```
lsslot -c phb
```

This displays output similar to the following:

PHB Name	Description	Device(s) Connected
PHB 1	Logical PCI Host Bridge	pci0
PHB 2	Logical PCI Host Bridge	pci2 scsi1
		pci1
		pci3
		pci4 scsi2

5. In case the PCI Host Bridge is assigned to the partition but has no ODM data, column will show blank as shown in this example. For example, when you enter:

```
lsslot -c phb
```

The output will look similar to the following:

PHB Name	Description	Device(s) Connected
PHB 4	Logical PCI Host Bridge	
PHB 5	Logical PCI Host Bridge	

6. To list all the logical host-Ethernet adapter-port devices that are assigned to the partition, enter:

```
lsslot -c port
```

Item	Description	Device(s) Connected
LHEA port name	Description	Device(s) Connected
Port 1	Logical HEA Port	ent4
Port 2	Logical HEA Port	ent7

When the logical host-Ethernet adapter port is assigned to the partition but has no ODM data, the column shows Unknown as shown in the example:

```
lsslot -c port
```

Item	Description	Device(s) Connected
LHEA port name	Description	Device(s) Connected
Port 4	Logical HEA Port	Unknown
Port 5	Logical HEA Port	Unknown

Files

Item

`/usr/sbin/lsslot`

Description

Contains the **lsslot** command.

>lssmbcmnt Command

Purpose

Lists the Server Message Block (SMB) client mount points that are defined in the `/etc/filesystems` file.

Syntax

```
lssmbcmnt [-c | -l | -p] [FileSystem]
```

Description

The **lssmbcmnt** command lists the specified SMB client mount points that are defined in the `/etc/filesystems` file.

Flags

- c**
Displays the SMB client mount points in a colon-delimited format.
- l**
Displays the SMB client mount points in standard format in which each field is separated by white space. This is the default setting.
- p**
Displays the SMB client mount points in a pipe-delimited format.

Parameters

FileSystem

Displays the characteristics of the specified SMB client file system. If you do not specify this parameter, all SMB client file systems are displayed.

Exit status

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

Example

- To list all SMB client mount points that are defined in the `/etc/filesystems` file, enter the following command:

```
lssmbcmnt
```

- To list all SMB client mount points in colon-delimited format, enter the following command:

```
lssmbcmnt -c
```

- To list SMB client file system entry for SMB client mount point, /mnt, enter the following command:

```
lsmbmnt /mnt
```

Location

/usr/sbin/lsmbcmnt



lsmbcred Command

Purpose

Lists the credential entries that are stored in the /etc/smbcred file for the specified server and username.

Syntax

```
lsmbcred [-s server_name] [-u user_name]
```

Description

When you run the **lsmbcred** command without any flags, the **lsmbcred** command lists all of the credential entries that have password for the username in the /etc/smbcred file. If you specify the **-s** or **-u** flag, the **lsmbcred** command displays credentials that match the specified server name or the specified username.

Flags

-s server_name

Specifies the remote host, which is the Server Message Block (SMB) server for which the credentials are displayed if a matching credential entry is found in the /etc/smbcred file.

-u user_name

Specifies the username for which the credentials are displayed if a matching credential entry is found in the /etc/smbcred file.

Exit status

0

The command completed successfully.

>0

An error occurred.

Example

To list all the credential entries on the xxx.in.ibm.com server, enter the following command:

```
lsmbcred -s xxx.in.ibm.com
```

An output similar to the following example is displayed:

```
server: xxx.in.ibm.com user: user1
server: xxx.in.ibm.com user: user2
```

Location

/usr/sbin/lssmbcred

Files

/etc/smbcred

Stores the credentials of the SMB client file system.

lssrad Command

Purpose

Displays the system SRADID (Scheduler Resource Allocation Domain Identifier) hierarchy and topology.

Syntax

```
lssrad [ -v ] { -s SRADID | -a }
```

Description

The **lssrad** command displays information related to SRADIDs, such as the processor and memory associated with the SRAD (Scheduler Resource Allocation Domain) and REF1 system detail level, where REF1 is the first hardware provided reference point that identifies sets of resources that are near each other. This command also displays the SRADID hierarchy and topology.

Flags

Item	Description
-a	Displays all SRADs in the system.
-s <i>SRADID</i>	Displays the specified SRADID.
-v	Displays resources in the SRAD, along with the REF1 System Detail Level that the SRAD belongs to, in verbose mode.

Examples

1. To display the list of all SRADs in the system, enter:

```
# lssrad -a
```

2. To verify that a specific SRAD exists, enter:

```
# lssrad -s 0  
SRAD  
0
```

```
# lssrad -s 5  
SRAD 5: No such SRAD
```

3. To display the topology of a specific SRAD, enter:

```
# lssrad -v -s 5
```

4. To display the SRADID hierarchy and topology, enter:

```
# lssrad -v -a
```

Files

Item	Description
<code>/usr/sbin/lssrad</code>	Contains the lssrad command

lssrc Command

Purpose

Gets the status of a subsystem, a group of subsystems, or a subserver.

Syntax

To Get All Status

```
lssrc [ -h Host ] -a
```

To Get Group Status

```
lssrc [ -h Host ] -g GroupName
```

To Get Subsystem Status

```
lssrc [ -h Host ] [ -l ] -s Subsystem
```

To Get Status by PID

```
lssrc [ -h Host ] [ -l ] -p SubsystemPID
```

To Get Subserver Status

```
lssrc [ -h Host ] [ -l ] -t Type [ -p SubsystemPID ] [ -o Object ] [ -P SubserverPID ]
```

To Get Subsystem Status in SMIT Format

```
lssrc -S [ -s Subsystem | -d ]
```

To Get Subserver Status in SMIT Format

```
lssrc -T [ -t Type ]
```

To Get Notify in SMIT Format

```
lssrc -N [ -n NotifyName ]
```

Description

The **lssrc** command sends a request to the System Resource Controller to get status on a subsystem, a group of subsystems, or all subsystems. The **lssrc** command sends a subsystem request packet to the daemon to be forwarded to the subsystem for a subserver status or a long subsystem status.

You can choose whether to request a short or long status for a subserver. When the **-l** flag is absent, the status request is assumed to be a short status. A short status of a subsystem, group of subsystems, or all subsystems is handled by the System Resource Controller.

When the **-l** flag is present for a subsystem, a status request is taken to the subsystem and the subsystem sends the status back. The **-l** flag is supported only for those subsystems not using signals as their communication method. For either a long or short status of a subserver, the subsystem is sent a status request packet, and the subsystem sends the status back.

The **lssrc** command output can sometimes show two entries for a particular daemon. One instance will be active and another instance will be inoperative. This can happen if the subsystem is modified (using the **mkssys** command or **chssys** command) without stopping the subsystem. The original subsystem will remain active and the modified instance will be inoperative until the subsystem is stopped and started again.

Flags

Item	Description
-a	Lists the current status of all defined subsystem.
-d	Specifies that the default record is printed.
-g <i>GroupName</i>	Specifies a group of subsystems to get status for. The command is unsuccessful if the <i>GroupName</i> variable is not contained in the subsystem object class.
-h <i>Host</i>	Specifies the foreign host on which this status 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.
-l	Requests that a subsystem send its current status in long form. Long status requires that a status request be sent to the subsystem; it is the responsibility of the subsystem to return the status.
-n <i>NotifyName</i>	Specifies the name of a notify method.
-N	Specifies that the Object Data Manager (ODM) records are output in SMIT format for the notify object class.
-o <i>Object</i>	Specifies that a subserver <i>Object</i> variable is passed to the subsystem as a character string.
-p <i>SubsystemPID</i>	Specifies a particular instance of the <i>SubsystemPID</i> variable to get status for, or a particular instance of the subsystem to which the status subserver request is to be taken.
-P <i>SubserverPID</i>	Specifies that a <i>SubserverPID</i> variable is to be passed to the subsystem as a character string.
-s <i>Subsystem</i>	Specifies a subsystem to get status for. The <i>Subsystem</i> variable can be the actual subsystem name or the synonym name for the subsystem. The command is unsuccessful if the <i>Subsystem</i> variable is not contained in the subsystem object class.
-S	Specifies that the ODM records are output in SMIT format for the subsystem object class.
-t <i>Type</i>	Requests that a subsystem send the current status of a subserver. The command is unsuccessful if the subserver <i>Type</i> variable is not contained in the subserver object class.
-T	Specifies that the ODM records are output in SMIT format for 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.

Exit status

- 0**
The command ran successfully.

1

The command failed.

Examples

1. To get the status of all subsystems on the local machine, enter:

```
lssrc -a
```

This gets the status of all subsystems known on the local machine.

2. To get the status of all subsystems on a foreign host, enter:

```
lssrc -h zork -a
```

This gets the status of all subsystems known on the zork machine.

3. To get the status of the srctest subsystem, enter:

```
lssrc -s srctest
```

This gets the status of all instances of the srctest subsystem on the local machine.

4. To get the status of the subsystem by PID, enter:

```
lssrc -p 1234
```

This gets the status of the subsystem with the subsystem PID of 1234 on the local machine.

5. To get the status of the tcpip subsystem group, enter:

```
lssrc -g tcpip
```

This gets the status of all instances of subsystems in the tcpip group on the local machine.

6. To get the status of the tester subserver, enter:

```
lssrc -t tester -p 1234
```

This gets the status of tester subserver that belongs to the srctest subsystem with the subsystem PID of 1234 on the local machine.

7. To get the status of the subsystem by PID, enter:

```
lssrc -l -p 1234
```

This gets the long status of the subsystem with the PID of 1234.

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 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.

lsts Command

Purpose

Lists thin server information. This command can be run on a thin server.

Syntax

```
lsts [ [-l{1|2|3}] . . . ] [-v] [ThinServer]
```

Description

The `lsts` command lists information pertaining to a thin server. The level of information to be listed depends on the numeric value specified by the `-l` flag, with a level ranging from 1 - 3 (3 being the most detailed). If a level is not specified, a default of level 1 information is displayed. This command can be run on both a NIM master or a thin server. When run on a NIM master and no argument is provided, the `lsts` command lists all thin servers in the environment controlled by the caller of the `lsts` command.

Flags

Item	Description
<code>-l{1 2 3}</code>	Specifies the level of output. 1 This level displays very limited information related to a thin server. The information listed shows only a brief summary of the thin server, such as the common image it is using. 2 This level displays more than just basic information related to a thin server. The level includes information pertaining to the software content of the thin server. 3 This level displays more in-depth information related to a thin server. The level includes information pertaining to the installation log of the thin server.
<code>-v</code>	Enables verbose debug output when the <code>lsts</code> command runs.

Parameters

Item	Description
<i>Thinserver</i>	Specifies the thin server where the command lists information about the client.

Exit Status

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

Security

Access Control: You must have root authority to run the `lsts` command.

Examples

1. To list brief status information for a thin server named lobo, enter:

```
lsts lobo
```

Information similar to the following is displayed:

```
Lobo:
  class      = machines
  type       = diskless
  platform   = chrp
  netboot_kernel = 64
  if1        = master_net jsblade04 0 ent1
  cable_type1 = bnc
  Cstate     = diskless or dataless boot is enabled
  prev_state = in the process of booting
  Mstate     = currently running
  boot       = boot
  dump       = dump_res
  paging     = paging_res
  root       = root_res
  spot       = 530spot_res
  cpuid      = 00012A80D000
  control    = master
  Cstate_result = success
```

2. To list software content for a thin server named lobo, enter:

```
lsts -l2 lobo
```

Software content similar to the following is displayed from the common image:

Fileset (Uninstaller)	Level	State	Type	Description

bos.64bit	5.2.0.75	C	F	Base Operating System 64 bit Runtime
bos.diag.com	5.2.0.75	C	F	Common Hardware Diagnostics
bos.diag.rte	5.2.0.75	C	F	Hardware Diagnostics
.				
.				
.				

3. To list both software content and status information for a thin server named lobo, enter:

```
lsts -l1 -l2 lobo
```

Location

/usr/sbin/lsts

Files

Item	Description
/etc/niminfo	Contains variables used by NIM.

lstun Command

Purpose

Lists tunnel definition(s).

Syntax

```
lstun [-v 4|6] [-t tid_list] [-p manual] [-a]
```

Description

Use the **lstun** command to list the tunnel definition(s) and their current status. This command can either list the tunnels in the tunnel database or in the active system.

Flags

Item	Description
-v	This flag specifies the IP version. For listing IP version 4 tunnel only, use the value of 4 . For listing IP version 6 tunnel only, use the value of 6 . If this flag is not used, both the version 4 and version 6 tunnels will be listed.
-t	Only list the tunnel definition and its current status for the tunnel whose tunnel ID is in tid_list . If this flag is not used, all the tunnel definitions and their current status will be listed.
-p	Selects the type of the tunnel to be listed. Using the -p flag with the value of manual lists manual tunnels only. The -p flag is for listing tunnel definitions in the tunnel database only and thus is mutually exclusive with the -a flag.
-a	Lists the tunnels active in the IP Security subsystem.

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.

lsuser Command

Purpose

Displays user account attributes.

Syntax

```
lsuser [ -R load_module ] [ -c | -C | -f ] [ -a List ] { ALL | Name [ ,Name ] ... }
```

Description

The **lsuser** command displays the user account attributes. You can use this command to list all attributes of all the system users or all the attributes of specific users. Since there is no default parameter, you must enter the **ALL** keyword to see the attributes of all the users. By default, the **lsuser** command displays all user attributes. To view selected attributes, use the **-a List** flag. If one or more attributes cannot be read, the **lsuser** command lists as much information as possible, but does not display empty attributes.

Note: If the *domainlessgroups* attribute is set in the */etc/secvars.cfg* file, the **lsuser** command lists the merged group from the LDAP module and the LOCAL module, if present.

By default, the **lsuser** command lists each user's attributes on one line. It displays attribute information as *Attribute=Value* definitions, each separated by a blank space. To list the user attributes in stanza format, use the **-f** flag. To list the information as colon-separated records, use the **-c** or **-C** flag.

You can use the System Management Interface Tool (SMIT) **smit lsusers** fast path to run this command.

Flags

Item	Description
------	-------------

-a <i>List</i>	Lists the attributes to display. The <i>List</i> variable can include any attribute that is defined in the chuser command and requires a blank space between attributes. If you specify an empty list, only the user names are displayed.
-----------------------	--

-c	Displays the user attributes in colon-separated records, as follows:
-----------	--

```
# name: attribute1: attribute2: ...
User: value1: value2: ...
```

If a value contains a `:` symbol, then in the output `:` symbol is prefixed with the `#!` symbols.

-C	Displays the user attributes in colon-separated records that are easier to parse than the output of the -c flag:
-----------	---

```
#name:attribute1:attribute2: ...
User1:value1:value2: ...
User2:value1:value2: ...
```

The output is preceded by a comment line that has details about the attribute represented in each colon-separated field. If you also specify the **-a** flag, the order of the attributes matches the order specified in the **-a** flag. If you do not have a value for a given attribute, the field is still displayed, but is empty. If a value contains a `:` symbol, then in the output the `:` symbol is prefixed with `#!` symbols. The last field in each entry ends with a newline character rather than a colon.

-f	Displays the output in stanzas, with each stanza identified by a user name. Each <i>Attribute=Value</i> pair is listed on a separate line:
-----------	--

```
user:
    attribute1=value
    attribute2=value
    attribute3=value
```

-R <i>load_module</i>	Specifies the loadable I&A module that is used to display the user account attributes. If the <i>domainlessgroups</i> attribute is set in the <code>/etc/secvars.cfg</code> file and the -R LDAP command is used, the attribute list is obtained from the LOCAL module. This condition applies if the user exists on the LOCAL module, and does not exist on the LDAP module. This condition also applies to the -R files command.
---------------------------------	--

Exit Status

This command returns the following exit values:

Item	Description
0	The command runs successfully and all requested changes are made.
>0	An error occurred. The printed error message lists further details to the type of failure.

Security

Access Control: This command must be a general user program with execute (x) access for all users. Since the attributes are read with the access rights of the user who starts the command, some users might not be able to access all the information. This command must have the *trusted computing base* attribute.

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 must also have the **aix.security.user.audit** authorization.

Files Accessed:

Mode	File
r	/etc/passwd
r	/etc/security/user
r	/etc/security/user.roles
r	/etc/security/limits
r	/etc/security/envIRON
r	/etc/group
r	/etc/security/audit/config
r	/etc/security/enc/LabelEncodings
> >lr	/etc/security/uattr <<<

Examples

1. To display the user `id` and group-related information about the `smith` account in stanza form, enter the following command:

```
lsuser -f -a id pgrp groups admgroups smith
```

Information similar to the following is displayed:

```
smith:
  ID=2457
  pgrp=system
  groups=system,finance,staff,accounting
  admgroups=finance,accounting
```

2. To display the user `id`, groups, and home directory of `smith` in colon format, enter the following command:

```
lsuser -c -a id home groups smith
```

Information similar to the following is displayed:

```
# name: ID:home:groups
smith: 2457:/home/smith:system,finance,staff,accounting
```

3. To display all the attributes of user `smith` in the default format, enter the following command:

```
lsuser smith
```

All the attribute information is displayed, with each attribute separated by a blank space.

4. To display all the attributes of all the users, enter the following command:

```
lsuser ALL
```

All the attribute information is displayed, with each attribute separated by a blank space.

Files

Item	Description
/usr/sbin/luser	Contains the luser command.
/etc/passwd	Contains basic user information.
/etc/security/limits	Defines resource quotas and limits for each user.
/etc/security/user	Contains the extended attributes of users.
/etc/security/user.roles	Contains the administrative role attributes of users.
/etc/security/environ	Contains the environment attributes of users.
/etc/group	Contains basic group attributes.
/etc/security/audit/config	Contains the audit configuration files.
> > /etc/security/uattr	Contains further user attributes. < <

lsusil Command

Purpose

Lists one or more user-specified installation location (USIL) instances.

Syntax

```
lsusil [-R RelocatePath | ALL]
```

Description

The **lsusil** command lists one or more USIL instances.

Flags

Item	Description
-R <i>RelocatePath</i>	The path to an existing USIL location

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/lsusil	Contains the lsusil command.

lsvfs Command

Purpose

Lists entries in the [/etc/vfs](#) file.

Syntax

```
lsvfs { -a | VfsName }
```

Description

The **lsvfs** command lists entries in the **/etc/vfs** file. You can display information about a specific Virtual File System (VFS) type or all known VFS types.

Flag

Item	Description
------	-------------

-a	Lists all stanzas in the /etc/vfs file, including the default stanza.
-----------	--

Parameter

Item	Description
------	-------------

<i>VfsName</i>	Specifies the name of a virtual file system.
----------------	--

Examples

1. To list the vfs entry named `newvfs`, enter:

```
lsvfs newvfs
```

2. To list all vfs types, enter:

```
lsvfs -a
```

Files

Item	Description
------	-------------

/etc/vfs	Contains descriptions of virtual file system types.
-----------------	---

lsvg Command

Purpose

Displays information about volume groups.

Syntax

```
lsvg [ -L ] [ -o ] | [ -n descriptorphysicalvolume ] | [ -i ] [ -l | -M | -p ] [ -m ] [ -P ] volumegroup ...
```

Description

The **lsvg** command displays information about volume groups. If you use the *volume*group parameter, only the information for that volume group is displayed. If you do not use the *volume*group parameter, a list of the names of all defined volume groups is displayed.

When information from the Device Configuration database is unavailable, some of the fields will contain a question mark (?) in place of the missing data. The **lsvg** command attempts to obtain as much information as possible from the description area when the command is given a logical volume identifier.

Note: To determine a volume group's major number, use the **ls -al /dev/VGName** command. This command lists the special device file that represents the volume group. The volume group major number is the same as the major device number of the special device file. For example, for a volume group named ha1vg, enter the following command:

```
ls -al /dev/ha1vg
```

This command returns the following:

```
crw-rw---- 1 root system 52, 0 Aug 27 19:57 /dev/ha1vg
```

In this example, the volume group major number is 52.

You can use the System Management Interface Tool (SMIT) **smit lsvg** fast path to run this command.

Flags

Item	Description
-i	Reads volume group names from standard input.
-l	Lists the following information for each logical volume within the group specified by the <i>volume group</i> parameter: LV A logical volume within the volume group. Type Logical volume type. LPs Number of logical partitions in the logical volume. PPs Number of physical partitions used by the logical volume. PVs Number of physical volumes used by the logical volume. Logical volume state State of the logical volume. Opened/stale indicates the logical volume is open but contains partitions that are not current. Opened/syncd indicates the logical volume is open and synchronized. Closed indicates the logical volume has not been opened. Mount point File system mount point for the logical volume, if applicable.
-L	Specifies no waiting to obtain a lock on the Volume group. Note : If the volume group is being changed, using the -L flag gives unreliable data.
-m	Lists the mirror pool that each logical volume copy in the volume group belongs to.

Item	Description
-M	<p>Lists the following fields for each logical volume on the physical volume:</p> <pre style="background-color: #f0f0f0; padding: 5px;">PVname:PPnum [LVname: LPnum [:Copynum] [PPstate]]</pre> <p>PVname Name of the physical volume as specified by the system.</p> <p>PPnum Physical partition number. Physical partition numbers can range from 1 to 1016.</p> <p>LVname Name of the logical volume to which the physical partitions are allocated. Logical volume names must be system-wide unique names, and can range from 1 to 64 characters.</p> <p>LPnum Logical partition number. Logical partition numbers can range from 1 to 64,000.</p> <p>Copynum Mirror number.</p> <p>PPstate Only the physical partitions on the physical volume that are not current are shown as stale.</p>
-n <i>descriptorphysicalvolume</i>	<p>Accesses information from the descriptor area specified by the <i>descriptorphysicalvolume</i> variable. The information may not be current, since the information accessed with the -n flag has not been validated for the logical volumes. If you do not use the -n flag, the descriptor area from the physical volume that holds the most validated information is accessed, and therefore the information displayed is current. The volume group need not be active when you use this flag.</p>
-o	<p>Lists only the active volume groups (those that are varied on). An active volume group is one that is available for use.</p>
-p	<p>Lists the following information for each physical volume within the group specified by the <i>volume group</i> parameter:</p> <p>Physical volume A physical volume within the group.</p> <p>PVstate State of the physical volume.</p> <p>Total PPs Total number of physical partitions on the physical volume.</p> <p>Free PPs Number of free physical partitions on the physical volume.</p> <p>Distribution The number of physical partitions allocated within each section of the physical volume: outer edge, outer middle, center, inner middle, and inner edge of the physical volume.</p>
-P	<p>Lists the mirror pool that each physical volume in the volume group belongs to.</p>

Information displayed if you do not specify any flags:

Item	Description
VOLUME GROUP	Name of the volume group. Volume group names must be unique systemwide and can range from 1 to 15 characters.
VOLUME GROUP STATE	State of the volume group. If the volume group is activated with the varyonvg command, the state is either active/complete (indicating all physical volumes are active) or active/partial (indicating some physical volumes are not active).
PERMISSION	Access permission: read-only or read-write .
MAX LVs	Maximum number of logical volumes allowed in the volume group.
LVs	Number of logical volumes currently in the volume group.
OPEN LVs	Number of logical volumes within the volume group that are currently open.
TOTAL PVs	Total number of physical volumes within the volume group.
ACTIVE PVs	Number of physical volumes that are currently active.
VG IDENTIFIER	The volume group identifier.
PP size	Size of each physical partition.
TOTAL PPs	Total number of physical partitions within the volume group.
FREE PPs	Number of physical partitions not allocated.
ALLOC PPs	Number of physical partitions currently allocated to logical volumes.
QUORUM	Number of physical volumes needed for a majority.
VGDS	Number of volume group descriptor areas within the volume group.
AUTO-ON	Automatic activation at IPL (yes or no).
CONCURRENT	States whether the volume group is Concurrent Capable or Non-Concurrent Capable.
AUTO-CONCURRENT	States whether you should autovary the Concurrent Capable volume group in concurrent or non-concurrent mode. For volume groups that are Non-Concurrent Capable, this value defaults to Disabled.
VG MODE	The vary on mode of the volume group: Concurrent or Non-Concurrent.
NODE ID	Node id of this node if volume group is varied on in concurrent node.
ACTIVE NODES	Node ids of other concurrent nodes that have this volume group varied on.
MAX PPs Per PV	Maximum number of physical partitions per physical volume allowed for this volume group.
MAX PVs	Maximum number of physical volumes allowed in this volume group. This information is displayed only for 32 and 128 PV volume groups.
LTG size	Logical track group size of the volume group. The maximum amount of data that can be transferred in one I/O request to the disks of the volume group. The LTG size will be displayed in kilobytes unless the LTG size is greater than 1 MB, in which case megabytes will be used. If the volume group was created on AIX 5.3, then it is capable of dynamically determining the LTG size based-on the disk topology and it is listed as Dynamic . If that capability is disabled by the user with the varyonvg -M option, then it will be listed as Static . If the capability does not exist because the volume group was created prior to AIX 5.3, then the VG will not be listed as Static or Dynamic .

Item	Description
BB POLICY	Bad block relocation policy of the volume group.
SNAPSHOT VG	Snapshot volume group name if the snapshot volume group is active, else snapshot volume group identifier.
PRIMARY VG	Original volume group name of a snapshot volume group if the original volume group is active, else original volume group identifier.
PV RESTRICTION	Displays the existing PV type restriction on the physical volumes comprising the volume group. A value of <i>none</i> indicates no PV restriction exists for the volume group. A value of <i>SSD</i> indicates the volume group has a PV restriction requiring all PVs to be <i>SSD</i> type PVs. No other values are supported.
INFINITE RETRY	Lists the infinite retry option of the volume group.
CRITICAL VG	Lists whether the <code>Critical VG</code> option is turned on or off for the volume group.
FS SYNC OPTION	Lists whether the logical volume manager resynchronizes the blocks allocated only by the Enhanced Journaled File System (JFS2), if JFS2 is mounted.
CRITICAL PVs	Lists Critical PVs option of the volume group. This information is available in IBM AIX 7.2 with Technology Level 1, or later.
>> ENCRYPTION	Lists the data encryption option of the volume group. This information is available in AIX 7 with 7200-05, or later. <<<

Examples

1. To display the names of all active volume groups, enter the following command:

```
lsvg -o
```

2. To display the names of all volume groups within the system, enter the following command:

```
lsvg
```

3. To display information about volume group `vg02`, enter the following command:

```
lsvg vg02
```

The characteristics and status of both the logical and physical partitions of volume group `vg02` are displayed.

4. To display the names, characteristics, and status of all the logical volumes in volume group `vg02`, enter the following command:

```
lsvg -l vg02
```

Files

Item	Description
<code>/usr/sbin</code>	Contains the directory where the lsvg command resides.

lsvgfs Command

Purpose

Displays a list of file systems belonging to a volume group.

Syntax

lsvgfs *volume group*

Description

The **lsvgfs** command displays a list of file systems that belong to the specified volume group.

Parameters

Item	Description
<i>volume group</i>	Specifies a volume group.

Examples

1. To display a list of file systems in the volume group vg02, enter the following command:

```
lsvgfs vg02
```

Exit Status

The **lsvgfs** command returns the following exit values:

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

Files

Item	Description
<i>/usr/sbin</i>	Contains the directory where the lsvgfs command resides.

lsvirprt Command

Purpose

Displays the attribute values of a virtual printer.

Syntax

lsvirprt [**-q** *QueueName* **-d** *DeviceName* { [**-f** *Format*] [**-n**] [**-a** *AttributeName* | **-s** *SectionName*] ... | **-i** | **-D** }]

Description

The **lsvirprt** command displays the attribute values for the virtual printer assigned to the *PrintQueueName* and *QueueDeviceName* variables.

The **lsvirprt** command becomes interactive if no flags are specified with the command. A list of print queue names is displayed, and a prompt appears requesting that the desired print queue name be selected. After a valid print queue name is selected, a prompt appears requesting that attribute names be entered. If an attribute name of * (asterisk) is entered, a list of all attributes is displayed.

Note: Attribute names for default values of the **qprt** command line flags can be specified by entering the flag letters. For example, to view the default value for the **-w** flag (page width), enter the **w** attribute name. All other attribute names must be 2 characters long.

You can use the System Management Interface Tool (SMIT) **smit lsvirprt** fast path to run this command.

Flags

Item	Description
-a <i>AttributeName</i>	Specifies the name of an attribute for which information is to be displayed. The flag cannot be used with the -s flag. The -a flag can be specified many times to list multiple attributes. The <i>AttributeName</i> value can be a single-character name (for example, j), a simple two-character name (for example, ci), or a regular expression that specifies multiple attributes (for example, ^i.*).
-d <i>QueueDeviceName</i>	Specifies the name of the queue device to which the virtual printer is assigned. This flag is optional, but can be specified only if the -q flag is specified.
-D	Displays data streams supported by a given queue and queue device name variable values. The -D flag displays the default data stream first and all other supported data streams in alphabetical order.
-f <i>Format</i>	Specifies the display format for attribute information. Attribute information includes the attribute value, limits field, and attribute description. The <i>Format</i> value is specified in printf format. The -f <i>Format</i> option also supports the following predefined set of position arguments: Note: [.*] is not a required element for the following format values. %1\$[.*]s Message catalog name %2\$[.*]d Message number %3\$[.*]s Attribute name %4\$[.*]s Limits field %5\$[.*]s Attribute value %6\$[.*]s Attribute description %7\$c Second character of attribute name.
-i	Sets the command to interactive mode. The -q and -d flags must be specified with the -i flag. If values have been assigned to the <i>QueueName</i> and <i>DeviceName</i> variables, the command does not prompt for the queue and device names and accepts attribute names interactively.
-n	Displays only the specified attributes that have nonnull values.
-s <i>SectionName</i>	Specifies a section name in the virtual printer attribute database of the specified queue and queue device. The <i>SectionName</i> values begin with two underscores and contain three characters that identify a section. For example, the name of a section that contains all the flag attributes is __FLG . The -s flag can not be used with the -a flag. This option can be repeated to list multiple attributes. The <i>SectionName</i> variable value can be a regular expression.

Item	Description
-q <i>PrintQueueName</i>	Specifies the name of the print queue to which the virtual printer is assigned. This flag is optional, but can be specified only if the -d flag is specified.

Examples

1. To show the attribute values for the **w** (default page width) and **si** (user to receive the "Intervention Required" messages) attributes for the virtual printer assigned to the **mypro** queue device on the **proq** print queue, enter:

```
lsvirprt -dmypro -qproq -a w -a si
```

The output from this command is:

Name	Description	Value
_w	COLUMNS per page	136
si	USERS to get intervention messages	

2. To show the same attributes in Example 1, but to be prompted for the flag values, enter:

```
lsvirpt
```

The output from this command is:

1	e4039c	@piobe	ibm4039 (PCL Emulation)
2	e4039s	@piobe	ibm4039 (PostScript)
3	fjzhp4s	jzfile	hplj-4 (PostScript)
4	hpc14	hp@pc15	hplj-4 (PCL)
...			

3. To list attributes in a section for header and trailer pipelines for the **que** queue and the **dev** device, enter:

```
lsvirpt -qqe -ddev -s__HTP
```

The output from this command is:

Name	Description	Value
sh	Pipeline for Header Page	%Ide/pioburst %F[H] %Idb/H.ascii %Ide/pioformat -@%Idd/%Imm -!%Idf/piof5202 -L! -J! %IsH
st	Pipeline for Trailer Page	%Ide/pioburst %F[H] %Idb/T.ascii %Ide/pioformat -@%Idd/%Imm -!%Idf/piof5202-L! -t%o%G_1%r%{14}%-%d %IsT

4. To list all the data streams supported for the **que** queue and the **dev** device, enter:

```
lsvirpt -qqe -ddev -D
```

The output from this command is:

```
a ASCII  
p pass-through  
s PostScript
```

5. To list names and descriptions of all attributes in a printer attribute database for the **que** queue and the **dev** device in a specific format, enter:

```
lsvirprt -qqe -ddev -a'.*' -f' %3$5.5s: %6$s\\n'
```

The output from this command is:

```
__FLG: Values That May Be Overridden With Flags
_A:    stderr returned?
_E:    Double spacing flag
_F:    (not used) Font file name
_H:    Name to Replace Host Name of Burst Page
...
```

6. To list all the sections in a printer attribute data base for the que queue and the dev device in a specific format, enter:

```
lsvirprt -qqe -ddev -a'__.*' -f'%3$s: %6$s\\n'
```

The output from this command is:

```
__FLG: Values That May Be Overridden With Flags On the Command
Line
__SYS: Other Values Of Interest To the Streams Administrator
__IDS: Pipelines For Input Data Streams (2 char,1st="i",2nd=data
stream name)
__PFL: Flags Prohibited For Input Data Streams (2 char,1st="I",
2nd=data stream name)
__FIL: Command Strings For Filter Flags (2 char, 1st="f",
2nd=flag)
__DIR: Directories
...
```

Files

Item	Description
<code>/etc/qconfig</code>	Contains the configuration file.
<code>/usr/sbin/lsvirprt</code>	Contains the lsvirprt command.
<code>/var/spool/lpd/pio/@local/custom/*</code>	Contains virtual printer attribute files.
<code>/var/spool/lpd/pio/@local/ddi/*</code>	Contains the digested virtual printer attribute files.

lsvmode Command

Purpose

Display the current video mode of the X server.

Note: This command is usable only while the X server is running.

Syntax

lsvmode

Description

The **lsvmode** command displays the current output device and viewport size used by the X server.

Security

Access Control: Any User

Auditing Events: None

Exit Status

The following exit values are returned:

Item	Description
------	-------------

- | | |
|----|------------------------|
| 0 | Successful completion. |
| >0 | An error occurred. |

Examples

To display the current video mode of the X server.

```
lsvmode
```

Something similar to the following displays:

```
Current video mode information
Logical screen size  [1024x768]
Viewport size       [ 640x480]
Vertical sync. (Hz) [ 60]
Active output device [LCD][CRT]
```

Files

Item	Description
<code>/usr/bin/X11/lsvmode</code>	Contains the lsvmode command.

lsvpd Command

Purpose

Lists the vital product data (VPD) associated with the field replaceable units (FRUs) configured on a system.

Syntax

```
lsvpd [-m] [-s serial_number] [-t type_model] [-v]
```

Description

The `lsvpd` command collects vital product data (VPD) for field replaceable units (FRUs). It reads the appropriate device configuration object classes in the Object Data Manager (ODM) and gathers VPD and general system information. The `lsvpd` command can extract additional VPD by reading data structures that are specific to the platform on which it is running. Data is provided in a format that aids service personnel in monitoring device quality and performance.

Note: Output from the `lsvpd` command is informational only and subject to change as hardware definitions change. Portable applications should not parse this data.

Flags

Item	Description
-m	Distinguishes between a FRU with global VPD and a FRU with partition private VPD. FRUs with global VPD begin with a line in the format of *FC *****. FRUs with partition private VPD begin with a line in the format of *FC =====. If this flag is not specified, the output begins with a line in the format of *FC ????????. For LPARs, this option distinguishes between FRUs associated with the overall system and FRUs assigned to a specific partition.
-s <i>serial_number</i>	Specifies the serial number for the system. The optional <i>serial_number</i> parameter is obsolete and should not be used. If the serial number is entered, that value will be used in the output of the command. In some cases, <code>lsvdpd</code> is unable to automatically determine the serial number. In these cases, the user must supply the value in order for it to be displayed in the command output.
-t <i>type_model</i>	Specifies the type model for the system. The optional <i>type_model</i> parameter is obsolete and should not be used. If the type model is entered, that value will be used in the output of the command. In some cases, <code>lsvdpd</code> is unable to automatically determine the type model. In these cases, the user must supply the value in order for it to be displayed in the command output.
-v	Produces verbose output for debugging purposes only.

Exit Status

Item	Description
0	The command completed successfully.
1	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. Output for the `lsvdpd` command is similar to the following.

Note: Portable applications should not parse this data.

```
*VC 5.0
*TM IBM,7029-6E3
*SE IBM,0110B721E
*PI 000B721E
*OS AIX 5.3.0.0
```


*FC ????????
*DS Platform Firmware
*YL U0.1-P1-X1/Y1
*RM 3F041029
*VK RS6K
*FC ????????
*DS System Firmware
*YL U0.1-P1-X1/Y2
*RM RG041029_d79e00_regatta
*VK RS6K
*FC ????????
*DS System VPD
*YL U0.1
*SE 10B721E
*TM 7029-6E3
*MN IBM980
*VK RS6K
*PA Y
*BR I0
*FC ????????
*DS PS CEC OP PANEL
*YL U0.1-L1
*SN YL1124350190
*EC H64013
*CC 28D3
*FN 97P3352
*DC BD 200210290851
*VK RS6K
*FC ????????
*DS 2 WAY BACKPLANE
*YL U0.1-P1
*SN YL1123354433
*PN 80P3099
*CC 26F5
*CE 1
*FN 80P3099
*VK RS6K
*FC ????????
*DS CSP
*YL U0.1-P1-X1
*SN YL1024360048
*PN 80P5573
*CC 28D0
*CE 1
*FN 80P5573
*RM 3F041029
*VK RS6K
*FC ????????
*DS IBM 1.8V VRM
*YL U0.1-P1-V1
*FN 24P6892
*VK RS6K
*FC ????????
*DS IBM 2.5V VRM
*YL U0.1-P1-V2
*FN 53P5623
*VK RS6K
*FC ????????
*DS IBM 1.2V VRM
*YL U0.1-P1-V3
*FN 53P5621
*VK RS6K
*FC ????????
*DS A IBM AC PS
*YL U0.1-V2
*SN YL1023C90045
*EC H85582
*CC 51B5
*FN 97P5101
*VK RS6K
*FC ????????
*DS IBM Air Mover
*YL U0.1-F1
*FN 53P4612
*VK RS6K
*FC ????????
*DS IBM Air Mover
*YL U0.1-F2
*FN 53P4612
*VK RS6K
*FC ????????
*DS IBM Air Mover

```

*YL U0.1-F3
*FN 53P4612
*VK RS6K
*FC ????????
*DS VSBPD4E1 U4SCSI
*YL U0.1-P2
*SN YL11243550F4
*PN 80P4611
*EC H85823
*CC 28D2
*FN 80P4610
*FS
*VK RS6K
*FC ????????
*DS MEDIA BACKPLANE
*YL U0.1-P4
*SN YL1124341459
*PN 80P3510
*EC H85610
*CC 28D1
*FN 80P3516
*VK RS6K
*FC ????????
*DS PCI-X Dual Channel Ultra320 SCSI Adapter
*AX sisscsia1
*PL 1Z-08
*CD 10140266
*PN 97P6513
*FN 97P6513
*SN YL11A5013461
*MN 001A
*EC 1
*RM 05080064
*Z0 5702
*YL U0.1-P1-I1
*FC ????????
*DS IDE DVD-ROM Drive
*AX cd0
*PL 1G-19-00
*MF IBM
*TM DROM00205
*RL NR38
*Z0 058002028F000010
*YL U0.1-P1-X1/Q6-A0
*FC ????????
*DS 16 Bit LVD SCSI Disk Drive
*AX hdisk0
*PL 1S-08-00-5,0
*MF IBM
*TM ST336607LC
*FN 00P3068
*RL 4335304A
*SN 000D7D3B
*EC H12094
*PN 00P2676
*Z0 000003129F00013E
*Z1 0812C512
*Z2 0002
*Z3 04341
*Z4 0001
*Z5 22
*Z6 H12094
*YL U0.1-P1/Z1-A5
*FC ????????
*DS 16 Bit LVD SCSI Disk Drive
*AX hdisk1
*PL 1S-08-00-8,0
*MF IBM
*TM ST336607LC
*FN 00P3068
*RL 4335304A
*SN 000D7996
*EC H12094
*PN 00P2676
*Z0 000003129F00013E
*Z1 0812C512
*Z2 0002
*Z3 04340
*Z4 0001
*Z5 22
*Z6 H12094
*YL U0.1-P1/Z1-A8

```

```

*FC ????????
*DS Diskette Drive
*AX fd0
*PL 01-D1-00-00
*YL U0.1-P1-X1-D1
*FC ????????
*DS Asynchronous Terminal
*AX tty0
*PL 01-S1-00-00
*YL U0.1-P1-X1/S1-L0
*FC ????????
*DS SCSI Enclosure Services Device
*AX ses0
*PL 1S-08-00-15,0
*MF IBM
*TM VSBPD4E1 U4SCSI
*RL 4610
*SN 243550F4
*Z0 0D0002022F004000
*FN 80P4610
*FL DB1
*FS
*YL U0.1-P1/Z1-Af
*FC ????????
*DS IBM MS 512 MB
*YL U0.1-P1-M5
*SN YL10243591YT
*PN 00P5767
*CC 30D2
*FN 00P5767
*SZ 512
*VK RS6K
*FC ????????
*DS IBM MS 512 MB
*YL U0.1-P1-M7
*SN YL10243591YP
*PN 00P5767
*CC 30D2
*FN 00P5767
*SZ 512
*VK RS6K
*FC ????????
*DS IBM MS 512 MB
*YL U0.1-P1-M4
*SN YL1024359208
*PN 00P5767
*CC 30D2
*FN 00P5767
*SZ 512
*VK RS6K
*FC ????????
*DS IBM MS 512 MB
*YL U0.1-P1-M2
*SN YL1024359204
*PN 00P5767
*CC 30D2
*FN 00P5767
*SZ 512
*VK RS6K

```

Location

/usr/sbin/lsvpd

lsvsd Command

Purpose

Displays configured virtual shared disks and their characteristics.

Syntax

lsvsd [-l | -s[*vsd_name...*]] | [-i]

Description

The **lsvsd** command displays information about virtual shared disks currently configured on the node on which the command is run. If a list of virtual shared disks follows the flags, information about those virtual shared disks is displayed. **lsvsd** with no arguments or flags lists the names of all the virtual shared disks currently configured on the node.

The **lsvsd** command displays information about both the configuration and the usage of a virtual shared disk.

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

```
smit vsd_mgmt
```

and select the **Show All Managed Virtual Shared Disk Characteristics** option.

Flags

-l

Lists the name of the virtual shared disk, the minor number, the state, the current server node number, and, at the server only, the major and minor number of the logical volume. (This flag is lowercase **l**, as in **list**.)

The state field can have one of the following values:

- STP Stopped
- SUS Suspended
- ACT Active

An asterisk (*) in front of any of these values indicates that the virtual shared disk has been fenced from this node.

This flag is not compatible with the **-s** flag.

The *server_list* of the virtual shared disk is listed.

-s

Lists usage statistics about the virtual shared disks. It lists the number of local logical read and write operations, the number of remote logical read and write operations, the number of client logical read and write operations, the number of physical reads and writes, and the number of 512-byte blocks read and written. The number of blocks read and written is cumulative, so issue **ctlvsd -V** to reset this count before measuring it.

The local logical operations are requests which were made by a process executing at the local node, whereas the remote logical operations were made by a process executing on a remote node. *Client operations* are those local logical requests that cannot be satisfied locally, and have to be sent to a remote node. *Physical* operations are those server operations which must be passed to the underlying disk device.

This flag is not compatible with the **-l** flag.

-i

Lists the “node to IP address” map that is currently used by the virtual shared disk driver.

Parameters

vsd_name

Specifies a virtual shared disk. This parameter is valid only with the **-l** and **-s** flags.

Security

You must be in the AIX **bin** group 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

1. To list all virtual shared disks in the system, enter:

```
lsvsd
```

The system displays a message similar to the following:

```
vsd00
vsd01
.
.
```

2. To list virtual shared disks and their characteristics, enter:

```
lsvsd -l
```

The system displays a message similar to the following:

minor	state	server	lv_major	lv_minor	vsd_name	size (MB)
83	STP	-1	0	0	vsdn08v3	20
84	STP	-1	0	0	vsdn08v4	16

3. To list statistics about virtual shared disks and precede the column output with a header, enter:

```
lsvsd -s
```

The system displays a message similar to the following:

lc-rd	lc-wt	rm-rd	rm-wt	c-rd	c-wt	p-rd	p-wt	br	bw	vsd_name
84	84	2858	169	0	0	348	253	164	184	vsd.vsd1
0	0	0	0	0	0	0	0	0	0	vsd.rl01
0	0	0	0	0	0	0	0	0	0	vsd.rl02

The following table spells out the names of the headers used in the displays for the **-l** and **-s** options:

Header

Meaning

minor

Virtual shared disk minor number

state

State of this virtual shared disk: *active, stopped, suspended*

server

Primary node for this virtual shared disk

lv major

Logical volume major number

lv minor

Logical volume minor number

vsd_name

Name of this virtual shared disk

lc-rd

Local logical reads

lc-wt

Local logical writes

rm-rd

Remote logical reads

rm-wt

Remote logical writes

c-rd

Client logical reads

c-wt

Client logical writes

p-rd

Physical reads

p-wt

Physical writes

br

Blocks read

bw

Blocks written

Location`/opt/rsct/vsd/bin/lsvsd`

lswlmconf Command

Purpose

Lists Workload Manager (WLM) configurations.

Syntax`lswlmconf [-r | -s | -c | -d Config] [-l] [-t TimeSpec]`**Description**

The **lswlmconf** command lists by default all WLM configurations and, using its flags, it is able to do the following:

- Tell which is the current configuration or set name.
- Give the list of all existing regular WLM configurations.
- Give the list of all existing WLM configuration sets.
- Tell which configuration of a set is (or would be) applicable currently (or at some time of the week).
- Tell the type of a configuration.

Flags

Item	Description
-c	Restricts the displayed configurations to the current configuration or set.
-d <i>Config</i>	Restricts the displayed configurations to the <i>Config</i> configuration or set.
-l	Modifies the way dates are displayed for configuration sets (ineffective for regular configurations). Sets are displayed with their currently applicable regular configuration, in the form <i>confset/config</i> .

Item	Description
-r	Restricts the displayed configurations to regular configurations only.
-s	Restricts the displayed configurations to configuration sets only.
-t <i>TimeSpec</i>	Uses <i>TimeSpec</i> instead of the current time to display applicable regular configuration of sets. <i>TimeSpec</i> consists in the day of the week (0 for Sunday to 6 for Saturday) and the time of the day in 24 hours format separated with a comma, in a form similar to time ranges as described in the confsetcntrl command. For example, to know which configuration would apply on Mondays at noon, use -t 1,12:00 .

Note: The **-t** flag is only effective with **-l** flag.

Examples

The following examples demonstrate how to display, change, and use WLM configurations using the **lswlmconf** command, the **confsetcntrl** command, the **wlmcheck** command, and the **wlmcntrl** command.

1. To find the WLM configurations, type:

```
lswlmconf
```

The output to this command might look similar to the following:

```
standard
template
fvtrules
fvtlimits
fvtreregul
fvtdfct
fvtsynt
fvthreads
```

2. To show the current WLM configuration, type:

```
lswlmconf -c
```

The output might look similar to the following:

```
fvtlimits
```

3. To show configuration sets, use the **lswlmconf** with the **-s** flag as follows:

```
lswlmconf -s
```

Since this example configuration contains no configuration sets, this command produces a message indicating that no matching configuration was found.

4. In order to create a configuration set using `standard` as the default configuration, type:

```
confsetcntrl -C confset1 standard
```

5. Now, use the **lswlmconf** command to show the new configuration set, as follows:

```
lswlmconf -s
```

The command now produces the following output:

```
confset1
```

6. In order to use the `fvtlimits` configuration for `confset1` on week days (Monday through Friday) by specifying a time range, type:

```
confsetcntrl -d confset1 -a fvtlimits 1-5
```

7. You might want this configuration only in the morning. You cannot change a time range. Instead you must remove the time range and then create a new time range.

First, remove the old time range, as follows (confsetcntrl accepts day names, as reported by **locale day** or **locale abday** commands):

```
confsetcntrl -d confset1 -r fvtlimits monday-friday
```

Then create the new time range, as follows:

```
confsetcntrl -d confset1 -a fvtlimits 1-5,8:00-12:00
```

8. In order to add another time range for using the fvtregul configuration on Sundays, type:

```
confsetcntrl -d confset1 -a fvtregul 0
```

9. In order to display configuration set confset1, type:

```
confsetcntrl -d confset1
```

In this example, this command produces the following output:

```
fvtlimits:
    time = "1-5,8:00-12:00"

fvtregul:
    time = "0"

standard:
    time = "-"
```

10. In order to create a configuration set called confset2 using template as the default configuration, type:

```
confsetcntrl -C confset2 template
```

In order change confset2 so it will use the configuration fvtsynt every nigh, type:

```
confsetcntrl -d confset2 -a fvtsynt 18:00-10:00
```

11. In order to display the list of regular configurations, type:

```
lswlmconf -r
```

In this example, this produces the following output, (which demonstrates that in this example the list of regular configurations has not changed):

```
standard
template
fvtrules
fvtlimits
fvtregul
fvtdfct
fvtsynt
fvthreads
```

However, as expected, the list of configurations sets in this example has changed, as shown by the following command:

```
lswlmconf -s
```

This command produces the following output in this example:

```
confset1
confset2
```


12. In order to show which configuration would be currently active when that the **date** command reports the current time as Tue Jul 16 18:55:10 EET 2002 with configuration set confset2, type:

```
lswlmconf -d confset2 -l
```

In this example, this command produces the following output:

```
confset2/fvtsynt
```

You can also show which configurations would be active at another time. To show which configurations would be active on Sunday at 9:00am, type:

```
lswlmconf -l -t 0,9:00
```

This command produces the following output in this example:

```
standard
template
fvtrules
fvlimits
fvregul
fvtdfct
fvtsynt
fvthreads
confset1/fvregul
confset2/fvtsynt
```

In order to display this information only for configuration sets, type:

```
lswlmconf -s -l -t 0,9:00
```

This produces the following output in this example:

```
confset1/fvregul
confset2/fvtsynt
```

13. In order to remove configuration set confset2, type:

```
confsetcntrl -D confset2
```

lswlmconf -s now produces the following output in this example:

```
confset1
```

14. In order to check configuration set confset1, use the **wlmcheck** command as follows:

```
wlmcheck -d confset1
```

In this example, this produces the following output:

```
WLM is not running.
Checking classes and rules for 'confset1' configuration...
fvlimits/System
fvlimits/Default
fvlimits/Shared
fvlimits/login
fvregul/System
fvregul/Default
fvregul/Shared
standard/System
standard/Default
standard/Shared
```

15. In order to start using configuration set confset1 used in this example, type:

```
wlmcntrl -a -d confset1
```

The command **lswlmconf -c** now produces the following output:

```
confset1
```

The command **lswlmconf -cl**, which shows the active regular configuration, now produces the following output:

```
confset1/standard
```

Files

The configurations or sets files are subdirectories of **/etc/wlm**.

lswpar Command

Purpose

Lists characteristics of workload partitions.

Syntax

Tabular Formats:

```
lswpar [-b | -Br | -Bf | -D | -I | -M | -N] [-X] [-a fieldname [,...]] [-q] [-s state] [-t type] [wparname ...]
```

Paragraph Formats:

```
lswpar {-G | -L | -R | -S | -T} [-s state] [-t type] [wparname ...]
```

Delimited Formats:

```
lswpar {-c | -d delim} [-a fieldname [,...]] | -G | {-b | -Br | -Bf | -D | -X | -I | -M | -N} [-a fieldname [,...]] | -R | -S | -T] [-q] [-s state] [-t type] [wparname ...]
```

Description

The **lswpar** command prints information about 1 or more specified workload partition (or all workload partitions if none are specified) to standard output.

You can filter all listings according to the following workload partition states by using the **-s** flag:

Item	Description
Defined	The workload partition has been defined by the mkwpar command and is ready for use, but is not active. Start workload partitions in this state with the startwpar command.
Loaded	The workload partition has been configured in the kernel, but processes have not yet been started. Note: This state is visible only to programmatic consumers that use the lswpar command to start a workload partition.
Active	The workload partition is running normally.
Frozen	A checkpoint operation is initiated, and the processes of the workload partition are quiesced, awaiting the storing phase. Note: The Frozen state is only visible when you use the lswpar command to checkpoint a workload partition. The checkpoint or restart function requires additional software package other than base WPAR.

Item	Description
Paused	A checkpoint or restart operation has been performed, and the processes of the workload partition are ready to be resumed or killed. The checkpoint or restart functionality requires additional software.
Maintenance	A workload partition can be put into maintenance mode with the startwpar command. During maintenance mode, the workload partition has been configured in the kernel and the file systems have been mounted, but processes do not start.
Moving	An asynchronous checkpoint-restart operation has been performed. Although the workload partition is Active on the arrival server, the workload partition appears in the Moving state on the departure server until all resources have been successfully transferred. The checkpoint or restart functionality requires additional software.
Transitional	An administrative operation is in progress. The workload partition is being created, started, stopped, configured, and so on.
Broken	An administrative operation failed, leaving this workload partition in an unusable state.
Error	An error occurred because of invalid elements such as workload partition name and flags.

You can filter all listings according to the following workload partition types by using the **-t** flag:

Item	Description
Application	This type is an application workload partition, running a single process (or a group of processes that are invoked by that means) without isolated system services. The process or group of processes inherits its operating environment (file systems, security, devices, and so on) from the environment where the application workload partition was created.
System	This type is a system workload partition, emulating an independent, fully functional instance of the operating system.

If additional checkpoint or restart software is installed, you can also specify the following type:

Item	Description
Checkpointable	This workload partition is enabled for checkpoint or restart functions. Tip: This type is not a mutually exclusive workload partition type. Checkpointable workload partitions are still either System or Application workload partitions.

If additional versioned workload partition software is installed, you can also specify the following type:

Item	Description
Versioned	This workload partition is running in operating system compatibility mode. Tip: This type is not a mutually exclusive workload partition type. Versioned workload partitions are still System workload partitions.

Versioned

Tabular Formats

If no options are used, the output is tabular as shown in the following example:

Name	State	Type	Hostname	Directory	RootVG	WPAR

<i>wpar name</i>	<i>state</i>	<i>type</i>	<i>hostname</i>	<i>root directory</i>	<i>yes/no</i>	
...	

In tabular formats, there might be multiple records per WPAR. The **-D**, **-I**, **-M**, and **-N** flags display in a tabular format, but can be combined with the **-c** and **-d** flags to generate a delimited format. You can use the **-a** flag to customize which fields are displayed in tabular formats. You can use the **-q** flag to suppress the table headers.

Tip: Do not rely on the exact format and content of tabular output for automated purposes. Delimited formats are provided for output that can be parsed.

The width of each field in a tabular format is expanded according to the longest value in that column. Therefore, the output might wrap on narrow screens, depending on the fields requested.

Paragraph Formats

In paragraph formats, each field has one value for one WPAR. You can use the **-G**, **-R**, **-S**, and **-T** flags to display paragraph-style subsets of workload partition configurations. The **-L** flag displays a long listing, which is a combination of the data that is presented by the **-D**, **-G**, **-I**, **-M**, **-N**, **-R**, **-S**, and **-T** flags. Otherwise, formats cannot be combined.

Delimited Formats

Delimited formats are used to produce machine-readable formats. You can select any delimiting characters. You can generate delimited formats by using the **-c** or **-d** flag. You can use the **-a** flag to customize which fields are displayed. You can use the **-q** flag to suppress the header line. The paragraph format flags (**-G**, **-R**, **-S**, and **-T**) and tabular format flags (**-D**, **-I**, **-M**, and **-N**) can be used individually to limit the display to the corresponding predefined set of fields.

Flags

Item

`-a fieldname`

Description

Limits tabular or delimited displays to the specified 1 or more fields. Multiple field names must be separated by commas with no spaces. This flag is mutually exclusive with the `-G`, `-R`, `-S`, `-L`, or `-T` flag.

By default, the display consists of 1 WPAR per line. You can specify any of the following fields:

General

- **Name** (the WPAR name)
- **Cid** (the ID of a WPAR)
- **Key** (the key of a WPAR)
- **Rootvgwpar** RootVG WPAR (A yes/no value is displayed to identify whether the WPAR is a RootVG WPAR)
- **Uuid** (the UUID of a WPAR)
- **Vipwpar** VIP WPAR (A yes/no value is displayed to identify whether the WPAR is a VIP WPAR. This field only applicable for an Application WPAR)

- **State**
- **Type** (system or application)
- **Hostname**
- **Routing**
- **Directory**
- **Privateusr**

The sample output is displayed as shown here:

```
O> lswpar -a name,privateusr test
Name Private /usr?
-----
test      no
```

- **Script** (user-supplied start or stop script)
- **Auto**
 - If the value for this field is **yes**, the process is automatically started on global system restart.
 - If the value for this field is **no**, the process is not automatically started on global system restart.
- **Application** (tracked process for application WPARs)
- **Checkpointable**
- **Owner**
- **OStype** (a non-zero value indicates a versioned WPAR, a value of 0 or null indicates a native WPAR)

Resource Controls

- **Active**
 - If the value for this field is **yes**, resource controls are active
 - If the value for this field is **no**, resource controls are inactive
- **Rset**
- **Shares_CPU**
- **CPU**
- **Shares_memory**
- **Memory**
- **ProcVirtMem**
- **TotalProcesses**
- **TotalThreads**
- **totalPTYs**
- **totalLargePages**
- **totalVirtmem**
- **pct_msgIDs**
- **pct_semIDs**
- **pct_shmIDs**
- **pct_pinMem**

Item**Description**

(Fields that you can specify with the **-a** flag, are as follows)

Devices

- **Name** (the name of the WPAR)
- **Devname** (the name of the device)
- **Devtype** (pseudo, disk, clone)
- **Rootvg**

The display consists of 1 device per line. The following displays sample output:

```
O> lswpar -Da name,devname,rootvg test
```

```
Name  Device Name  RootVG
-----
test  hdisk1         yes
```

```
O> lswpar test
```

```
Name  State  Type  Hostname  Directory  RootVG WPAR
-----
test  D      S     test     /wpars/test  yes
```

Kernel Extensions

- **Name** (the name of the WPAR)
- **Kext** (the full path to the kernel extension)
- **Local**
- **Major**
- **kextstatus** (allocated or exported)
- **checksum** (checksum of the kernel extension)
- **mtime** (modification time of the kernel extension)

The display consists of 1 kernel extension per line.

WPAR-Specific Routes

A workload partition might have more than 1 route. So if you use the **-I** flag, you can specify the **-a** flag with the following fields:

- **name** (the WPAR name)
- **rtdest**
- **rtgateway**
- **rtinterface**
- **rttype**
- **rtfamily**

The display consists of 1 route per line.

Networks

A WPAR might have more than 1 network. So when you use the **-N** flag, you can specify the **-a** flag with the following fields:

- **Name** (the WPAR name)
- **Interface**
- **Address**
- **Netmask**
- **Broadcast**

The display consists of 1 network per line.

Mounts

A workload partition might have more than 1 mount. So when you use the **-M** flag, you can specify the **-a** flag with the following fields:

- **Name** (the WPAR name)
- **Mountpoint** (the mount point name)
- **Device** (the object mounted)
- **Vfs** (the virtual-file-system type)
- **Nodename** (node name, if the mount is remote)
- **Options** (any mount options)

The display consists of 1 mount per line.

Security

- **Privs** (the list of privileges)

Operation

- **Opname** (the name of the administration operation that is being performed)
- **Oppid** (the process ID of the operation)
- **Opstart** (the start time of the operation)

Item	Description
	<p>Bootlist</p> <p>When you use the -b flag, you can specify the -a flag with the following fields:</p> <ul style="list-style-type: none"> • name (the name of the WPAR) • bootlist (an ordered list of bootsets in comma-separated format) <p>The display consists of 1 bootlist per line.</p> <p>Bootset</p> <p>When you use the -Br flag, you can specify the -a flag with the following fields:</p> <ul style="list-style-type: none"> • name (the name of the WPAR) • devname (the name of the device) • vdevname (the name of the virtual device) • rootvg • bootset (the bootset devices of the WPAR) <p>The display consists of 1 bootset per line.</p> <p>When you use the -Bf flag, you can specify the -a flag with the following fields:</p> <ul style="list-style-type: none"> • name (the name of the WPAR) • mountpoint (the name of the mountpoint) • device (the object mounted) • vfs (the type of the virtual file system) • options (any mount options) • bootset (the bootset file systems of the WPAR) <p>The display consists of 1 bootset per line.</p>
-b	Shows the bootlist of the workload partition. If the -c or -d flag is not specified, the output for each WPAR has the following tabular format: Name - Bootlist
-Br	Produces detailed bootset information for each requested RootVG WPAR. If the -c or -d flag is not specified, the output for each WPAR has the following tabular format: Name - Device Name - Type - Virtual Device - RootVG - Bootset
-Bf	Produces detailed bootset information for each requested non-RootVG WPAR. If the -c or -d flag is not specified, the output for each WPAR has the following tabular format: Name - Mount Point - Device - Vfs - Options - Bootset
-c	Produces colon-separated output suitable for machine parsing. It is mutually exclusive with the -L flag. The default output format (when the -D , -G , -I , -M , -N , -R , -S , and -T flags are not used) is as follows: name:state:type:hostname:directory The state field is one or more of the following valid states: D Defined L Loaded A Active F Frozen P Paused N Maintenance M Moving T Transitional B Broken E Error The type field is one or more of the following valid types: A Application workload partition S System workload partition L Versioned system workload partition
-d delim	Produces delimiter-separated output suitable for machine parsing. It is mutually exclusive with the -L flag. The output format when the -d flag is specified is the same as with when the -c flag is specified, but with <i>delim</i> as the delimiter output between fields.

Item**Description****-D**

Produces detailed device information for each requested WPAR. It is mutually exclusive with the **-G**, **-I**, **-L**, **-M**, **-N**, **-R**, **-S**, or **-T** flag. If the **-c** or **-d** flag is not specified, each WPAR output has the following tabular format:

```
=====
Name - Device Name - Type - Virtual Device - RootVG - Status
```

-G

Produces detailed general setting information for each requested WPAR. It is mutually exclusive with the **-I**, **-L**, **-M**, **-D**, **-N**, **-R**, or **-T** flag. If you do not specify the **-c** or **-d** flag, each workload partition output has the following paragraph format:

```
=====
Name - State
=====
Type:                {S|A}
Hostname:            HostnameWPAR
-Specific Routing:  {yes|no}
Directory:          Directory
Start/Stop Script:  /path/to/userScript
Auto Start:         {yes|no}
Private /usr:       {yes|no}
Checkpointable:     {yes|no}
Application:        /path/to/trackedProcess
Owner:
Architecture: WPAR compatibility architecture
OStype:             <i>Integer value representing operating system type<i>
Cross-WPAR IPC:    {yes|no}
UUID:              String value representing universally unique ID
```

With the **-c** or **-d** flag, the output is as follows:

```
name:state:type:rootvgwpar:hostname:routing:directory:owner:script:
auto:privateusr:checkpointable:application:ostype
```

-I

Produces detailed information about user-specified network routes. The **-I** flag is mutually exclusive with the **-D**, **-G**, **-L**, **-M**, **-N**, **-R**, **-S**, or **-T** flag. The **-I** flag displays only the routing table entries that are explicitly specified with the **-I** flag of the **mkwpar**, **wparexec**, or **chwp** command. To see the full routing table for a workload partition, use the **netstat** command with the **-r** and **-@** flags. If you do not specify the **-c** or **-d** flag, tabular output is as produced as shown in the following example:

Name	Type	Destination	Gateway	Interface
<i>name</i>	<i>net host</i>	<i>destination</i>	<i>gateway</i>	<i>if</i>
...

With the **-c** or **-d** flag, delimited output is produced as shown in the following example:

```
name:rttype:rtdest:rtgateway:rtinterface:rtfamily
```

You can use the **-I** flag with the **-a** flag to limit the output to any combination of the following fields:

- **name** (the workload partition name)
- **rtdest**
- **rtgateway**
- **rtinterface**
- **rttype**
- **rtfamily**

Item

Description

-L

Specifies long format. Produces detailed paragraph-formatted information for each requested workload partition. It is mutually exclusive with the **-c**, **-d**, **-D**, **-G**, **-I**, **-M**, **-N**, **-q**, **-R**, **-S**, or **-T** flag.

If you want to parse data, do not use the **-L** output. Use the delimiter-separated forms (the **-c** or **-d** flag) for generating output that can be parsed. Each workload partition has formatted output similar to the following example:

```

=====
Name - State
=====
GENERAL
Type:                {S|A}
Hostname:            HostnameWPAR
-Specific Routing:   {yes|no}
Directory:           Directory
Start/Stop Script:   /path/to/userScript
Auto Start:          {yes|no}
Private /usr:        {yes|no}
Checkpointable:      {yes|no}
Application:         /path/to/trackedProcess
Owner:
OStype:              <i>Integer value representing operating system type<i>
Cross-WPAR IPC:      {yes|no}
Architecture:WPAR compatibility architecture
UUID:                String value representing universally unique ID

NETWORK
Interface            Address                Mask/Prefix            Broadcast
-----
if A.B.C.D A.B.C.D    A.B.C.D
...                  ...
USER-SPECIFIED ROUTES
Type                 Destination Gateway Interface
-----
net|host destination gateway if
...                  ...
FILESYSTEMS
MountPoint           Device                Vfs                    Nodename            Options
-----
mountpoint device vfs node options
...                  ...

(A long-format example by the -L flag, is as follows)

RESOURCE CONTROLS
Active:               {yes|no}
RSet:                 rset
CPU Shares:           n
CPU Limits:           m%-S%,H%
Memory Shares:        n
Memory Limits:        m%-S%,H%
Per Process Virtual Memory Limit: nMB
Total Processes:      n
Total Threads:        n
Total PTYS:           n
Total Large Pages:    n
Max Message queue IDs: n%
Max Semaphore IDs:    n%
Max Shared memory IDs: n%
Max Pinned memory:    n%

OPERATION
Operation: %c
Process ID: %p
Start time: %t

SECURITY SETTINGS
Privileges: privilege list
...

DEVICE EXPORTS
Name                 Type
Virtual Device       RootVG              Status
device name type virtual device name yes/no device status
...

```

Item	Description
-M	<p>Produces detailed mount information for each requested workload partition. The file systems that are mounted from outside the workload partition are listed and the file systems that are defined within the workload partition are not included. The -M flag is mutually exclusive with the -G, -I, -L, -N, -R, or -T flag. If you do not specify the -c or -d flag, tabular output is produced as shown in the following example:</p> <pre>Name MountPoint Device Vfs Nodename Options ----- name mountpoint device vfs node options</pre> <p>With the -c or -d flag, delimited output is produced as shown in the following example:</p> <pre>name:mountpoint:device:vfs:nodename:options</pre> <p>It can be used with the -a flag to limit the output to any combination of the following fields:</p> <ul style="list-style-type: none"> • Name (the workload partition name) • Mountpoint (the mount point name) • Device (the object mounted) • Vfs (the virtual-file-system type) • Nodename (node name, if the mount is remote) • Options (any mount options)
-N	<p>Produces detailed network information for each requested workload partition. It is mutually exclusive with the -G, -I, -L, -M, -R, -D, -S, or -T flag. If you do not specify the -c or -d flag, tabular output is produced as shown in the following example:</p> <pre>Name Interface Address(6) Mask/Prefix Broadcast ----- name if A.B.C.D A.B.C.D A.B.C.D ... name if S:T:U:V:W:X:Y:Z R</pre> <p>With the -c or -d flag, delimited output is produced as shown in the following example:</p> <pre>name:interface:address:mask_prefix:broadcast</pre> <p>You can specify the -N flag with the -a flag to limit the output to any combination of the following fields:</p> <ul style="list-style-type: none"> • Name (the WPAR name) • Interface • Address (the IPv4 or IPv6 address) • Mask_Prefix (the IPv4 netmask field or the IPv6 prefixlen field) • Broadcast <p>If a WPAR contains 1 or more name-mapped interfaces, the lswpar command shows only the information that is specified in the configuration file when the WPAR is in the Defined state. When the WPAR is in the Active state, the actual runtime network attributes are displayed.</p> <p>Note: When a delimited output is expected to contain IPv6 addresses, use the -d flag to specify an alternative delimiter because IPv6 addresses contain colons.</p>
-q	Suppresses table headers (quiet). It is valid only for tabular and delimited output formats.
-R	<p>Produces detailed resource control information for each requested WPAR. It is mutually exclusive with the -G, -I, -L, -M, -N, -D, -S, or -T flag. If you do not specify the -c or -d flag, each workload partition output has the following paragraph format:</p> <pre>===== Name - State ===== Active: {yes no} RSet: rset CPU Shares: n CPU Limits: m%-S%, H% Memory Shares: n Memory Limits: m%-S%, H% Per-Process Virtual Memory Limit: nMB Total Processes: n Total Threads: n Total PTYs: n Total Large Pages: n Max Message queue IDs: n% Max Semaphore IDs: n% Max Shared memory IDs: n% Max Pinned memory: n%</pre> <p>With the -c or -d flag, delimited output is as shown in the following example:</p> <pre>name:state:active:rset:shares_CPU:CPU:shares_memory:memory: procVirtMem:totalProcesses:totalThreads:totalPTYs: totalLargePages:pct_msgIDs:pct_semIDs:pct_shmIDs:pct_pinMem</pre>
-s {[D] [L] [A] [F] [P] [N] [M] [T] [B]}	Filters the output based on workload partition states. You can use more than 1 state code. See the -c flag for a description of the state codes.
-S	<p>Produces detailed security privilege information for each requested WPAR. It is mutually exclusive with the -D, -G, -I, -L, -M, -N, -R, or -T flag. If you do not specify the -c or -d flag, each workload partition output has the following paragraph format:</p> <pre>===== Name - State ===== Privileges: comma-separated list of privileges assigned to the workload partition</pre>
-t {[A][S][C][L]}	Filters the output based on workload partition types. You can use more than 1 type code. See the -c flag for a description of the type codes.

Item	Description
-T	<p>Produces detailed locking information for each requested workload partition. This flag is mutually exclusive with the -D, -G, -I, -L, -M, -N, -R, -S, or -s flag. It is mutually exclusive with the -q flag unless the -c flag is also specified. If you do not specify the -c flag, each workload partition output has the following format:</p> <pre> ===== Name - State ===== Operation: %c Process ID: %p Start time: %t </pre> <p>With the -c or -d flag, the output is as shown in the following example:</p> <pre> name:state:opname:opid:opstart </pre>
-X	<p>Produces detailed kernel extension information for each requested workload partition in turn. It is mutually exclusive with the -D, -G, -I, -L, -M, -N, -R, -S, or -T flag. If the -c or -d flag is not specified, each workload partition output has the following tabular format:</p> <pre> Name Extension Name Local Major Status checksum ----- name /path/to/extension local major status checksum ... </pre>
<i>wparname</i>	<p>Specifies 1 or more workload partitions. It must be last on the command line. It can contain shell-style wildcards to match multiple workload partition names. (In this case, use appropriate shell quotation marks to preclude shell expansion before the lswpar command receives the metacharacters.)</p>

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 view tabular information about all workload partitions, enter the following command:

```

# lswpar
Name      State Type  Hostname      Directory      RootVG WPAR
-----
bar       A     S     bar.austin.ibm.com /wpars/bar     yes
foo       D     S     foo.austin.ibm.com /wpars/foo     no
trigger   A     A     trigger        /

```

2. To view limited tabular information about application workload partitions only, enter the following command:

```

# lswpar -t A -a name,application,script
Name      Application      Script
-----
trigger   /usr/sbin/apachectl start /home/joe/trigger.script

```

3. To view colon-separated general information with no headers for all active and defined workload partitions, enter the following command:

```

# lswpar -G -c -q -s AD
bar:A:S:bar.austin.ibm.com:/wpars/bar:/home/bar/wpar.scr:no:no:yes::no
foo:D:S:foo.austin.ibm.com:/wpars/foo::no:no:no::no
trigger:A:A:trigger:~/home/joe/trigger.script
:no:no:yes:/usr/sbin/apachectl start:no

```

4. To view extended information about the workload partition named `trigger`, enter the following command:

```

# lswpar -l trigger
=====
trigger - Active
=====
GENERAL
Type:                A
Hostname:            triggerWPAR
-Specific Routing:  yes
Directory:           /
Start/Stop Script:  /home/joe/trigger.script
Auto Start:          no
Private /usr:        no

```

```

Checkpointable:      yes
Application:         /usr/sbin/apachectl start

NETWORK
Interface  Address          Mask/Prefix      Broadcast
-----
en0        1.2.3.4          255.255.255.0   1.2.3.255
en1        5.6.7.8          255.255.255.0   5.6.7.255

USER-SPECIFIED ROUTES
Type       Destination      Gateway          Interface
-----
net        9.1.2.24         1.2.3.1         en0
host       192.168.1.2     1.2.3.1         en1

FILESYSTEMS
MountPoint          Device          Vfs      Nodename  Options
-----
/share              /nfs2/share     nfs      nfserver  rw

RESOURCE CONTROLS
Active:              yes
RSet:                isp1
CPU Shares:          2
CPU Limits:           5%-10%,50%
Memory Shares:       3
Memory Limits:       10%-20%,30%
Per-Process Virtual Memory Limit: 1024MB
Total Processes:     64
Total Threads:       1024
Total PTYs:          8
Total Large Pages:   16
Max Message queue IDs: 20%
Max Semaphore IDs:   30%
Max Shared memory IDs: 50%
Max Pinned memory:   20%
OPERATION:
Operation:           restart
Process ID:          905266
Start time:          11:19

Privileges:
PV_AU_,PV_AU_ADD,PV_AU_ADMIN,PV_AU_PROC,
PV_AU_READ,PV_AU_WRITE,PV_AZ_ADMIN,
PV_AZ_CHECK,PV_AZ_READ,PV_AZ_ROOT,PV_DAC_,
PV_DAC_GID,PV_DAC_O,PV_DAC_R,PV_DAC_RID,
PV_DAC_UID,PV_DAC_W,PV_DAC_X,PV_DEV_CONFIG,
PV_DEV_QUERY,PV_FS_CHOWN,PV_FS_CHROOT

DEVICE EXPORTS
Name              Type          Virtual Device  RootVG      Status
-----
hdisk4            disk          hdisk4          yes         ALLOCATED
/dev/null         pseudo
/dev/tty          pseudo
/dev/irandom      pseudo
/dev/urandom      pseudo
/dev/console      pseudo
/dev/zero         pseudo
/dev/clone        pseudo
/dev/sad          clone         ALLOCATED

```

5. To view machine-readable network information that is separated by pipes for workload partitions called `roy`, enter the following command:

```

# lswpar -d'|' -N roy
#name|interface|address|mask_prefix|broadcast
roy|en0|192.168.1.50|255.255.255.128|192.168.1.127
roy|en1|2001:DB8::|32|

```

6. To view machine-readable, resource-control information for all workload partitions, enter the following command:

```

# lswpar -cR
#name:state:active:rset:shares_CPU:CPU:shares_memory:memory:procVirtMem:
totalProcesses:totalThreads:totalPTYs:
totalLargePages:pct_msgIDs:pct_semIDs:pct_shmIDs:pct_pinMem
dale:A:no:::::
roy:A:yes:rogers:3::2::32:128
trigger:A:yes:isp1:2:5%-10%,50%:3:10%-20%,30%:1024MB:64:1024:8:
16:20%:30%:50%:20%

```

7. To view operation information about the workload partition named `foo`, enter the following command:

```

# lswpar -T foo
=====
foo - Transitional
=====
Operation: restart
Process ID: 905266
Start time: 11:19

```

8. To view information about devices that are exported and allocated in the workload partitions named `roy`, enter the following command:

```
9. # lswpar -D roy
Name Device Name      Type      Virtual Device  RootVG  Status
-----
roy  /dev/null           pseudo
...
roy  fcs0                adapter
roy  hdisk2              disk      hdisk0         yes     EXPORTED
```

10. To view information about bootset of a RootVG workload partition, enter the following command:

```
lswpar -Bt <WPAR name>
```

11. To view information about bootlist of a workload partition, enter the following command:

```
lswpar -b <WPAR name>
```

luit Command

Purpose

Supports locale and ISO 2022 for Unicode terminals.

Syntax

```
luit [ options ] [ -- ] [ program [ args ] ]
```

Description

The **luit** command is a filter that runs between an arbitrary application and a UTF-8 terminal emulator. The **luit** command converts application output from the locale's encoding into UTF-8, and converts terminal input from UTF-8 into the locale's encoding.

Note: Multilingual applications must be set to generate only the UTF-8 code. You must not use the command to use different output other than UTF-8.

The **luit** command is invoked transparently by the terminal emulator. For information on running the **luit** command from the command line, see Examples.

Options

Item	Description
-h	Displays the help summary.
-list	Lists the supported charsets and encodings .
-v	Verbose.
-c	Converts the standard input to standard output.
-x	Exits as soon as the child function dies. This might cause the luit command to lose data at the end of the output of the child function.
-argv0 name	Sets the name of the child that is passed along with the argv[0] command.
-encoding encoding	Specifies that the luit command use encoding rather than the current locale encoding.
+oss	Disables interpretation of single shifts in the application output.
+ols	Disables interpretation of locking shifts in the application output.

Item	Description
+osl	Disables interpretation of character set selection sequences in the application output.
+ot	Disables interpretation of all sequences and passes all sequences in the application output to the terminal unchanged.
-k7	Generates 7-bit characters for keyboard input.
+kss	Disables generation of single-shifts for keyboard input.
+kssgr	Uses GL codes after a single shift for keyboard input. By default, GR codes are generated after a single shift when generating 8-bit keyboard input.
-kls	Generates locking shifts (SO/SI) for keyboard input.
-gl gn	Sets the initial assignment of GL. The argument must be one of g0 , g1 , g2 , or g3 . The default value depends on the locale, and is usually g0 .
-gr gk	Sets the initial assignment of GR. The default value depends on the locale, and is usually g2 except for EUC locales, where it is g1 .
-g0 charset	Sets the value of charset that is initially selected in G0 . The default value depends on the locale, but is usually ASCII.
-g1 charset	Sets the value of charset that is initially selected in G1 . The default value depends on the locale.
-g2 charset	Sets the value of charset that is initially selected in G2 . The default value depends on the locale.
-g3 charset	Sets the value of charset that is initially selected in G3 . The default value depends on the locale.
-ilog filename	Logs all the bytes received from the child into <i>filename</i> .
-olog filename	Logs all the bytes sent to the terminal emulator into <i>filename</i> .

Examples

1. To adapt an instance of XTerm to the locale's encoding, current versions of XTerm invoke the **luit** command automatically when it is needed. If you are using an older release of XTerm, or a different terminal emulator, you can invoke the **luit** command manually:

```
$ xterm -u8 -e luit
```

2. If you are running in a UTF-8 locale but need to access a remote machine that does not support UTF-8, the **luit** command can adapt the remote output to your terminal:

```
$ LC_ALL=fr_FR luit ssh legacy-machine
```

Files

Item	Description
<code>/usr/lib/X11/fonts/encodings/encodings.dir</code>	Contains the system-wide encoding directory.
<code>/usr/lib/X11/locale/locale.alias</code>	Contains the file mapping locales to locale encoding.

lvmo Command

Purpose

Manages lvm pbuf tunable parameters.

Syntax

```
lvmo -v Name -o Tunable [=NewValue]
```

```
lvmo -a [-v vgname]
```

```
lvmo -L [Tunable]
```

Description

The `lvmo` command sets or displays pbuf tuning parameters. The equal sign can be used to set a particular tunable to a given value. Otherwise, if no equal sign is used, the value of the tunable will be displayed.



Attention: Misuse of the `lvmo` command can cause performance degradation or operating-system failure.

The `lvmo -a` command generates pbuf and blocked I/O statistics. The pbuf and blocked I/O report has the following label:

Label	Description
vgname	Volume group name specified with the <code>-v</code> option.
pv_pbuf_count	The number of pbufs that are added when a physical volume is added to the volume group.
total_vg_pbufs	Current total number of pbufs available for the volume group.
max_vg_pbuf_count	The maximum number of pbufs that can be allocated for the volume group.
pervg_blocked_io_count	Number of I/O's that were blocked due to lack of free pbufs for the volume group.
pv_min_pbuf	The minimum number of pbufs that are added when a physical volume is added to any volume group.
global_blocked_io_count	Number of I/O's that were blocked due to lack of free pbufs for all volume groups.
aio_cache_pbuf_count	Current total number of pbufs available for aio_cache logical volume in the volume group.

Flags

Item	Description
-a	Displays value for all tunable parameters, one per line in pairs <code>tunable = value</code> .
-o <i>Tunable</i> [=NewValue]	Displays the value or sets <i>Tunable</i> to <i>NewValue</i> .

Item

-L [Tunable]

Description

Lists the characteristics of one or all of the tunables, one per line, using the following format:

NAME	CUR	DEF	BOOT	MIN	MAX	UNIT	TYPE
global_blocked_io_count	0	0	n/a	0	0		S
pervg_blocked_io_count							
max_vg_pbufs	16384	n/a	n/a	16384	none		S
max_vg_pbuf_count							
pv_min_pbuf							
pv_pbuf_count							
total_vg_pbufs							
max_vg_pbuf_count	0	0	n/a	0	none		M
max_vg_pbufs							
pv_min_pbuf							
pv_pbuf_count							
total_vg_pbufs							
pervg_blocked_io_count	0	0	n/a	0	0		S
global_blocked_io_count							
pv_min_pbuf	512	512	n/a	0	none		D
max_vg_pbufs							
max_vg_pbuf_count							
pv_pbuf_count							
total_vg_pbufs							
pv_pbuf_count	512	512	n/a	1	16384		D
max_vg_pbufs							
max_vg_pbuf_count							
pv_min_pbuf							
total_vg_pbufs							
total_vg_pbufs	512	n/a	n/a	0	0		S
max_vg_pbufs							
max_vg_pbuf_count							
pv_min_pbuf							
pv_pbuf_count							

...

where:
n/a means parameter not supported by the current platform or kernel

Parameter types:
S = Static: cannot be changed
D = Dynamic: can be freely changed
B = Bosboot: can only be changed using bosboot and reboot
R = Reboot: can only be changed during reboot
C = Connect: changes are only effective for future socket connections
M = Mount: changes are only effective for future mountings
I = Incremental: can only be incremented
d = deprecated: deprecated and cannot be changed

Value conventions:
K = Kilo: 2¹⁰ G = Giga: 2³⁰ P = Peta: 2⁵⁰
M = Mega: 2²⁰ T = Tera: 2⁴⁰ E = Exa: 2⁶⁰

Tunable Parameters**Item**

pv_pbuf_count

Description

The number of pbufs that is added when a physical volume is added to the volume group.

max_vg_pbuf_count

The maximum number of pbufs that can be allocated for the volume group. **Note:** The volume group must be varied off and varied on again for this value to take effect. This value does not affect rootvg.

pv_min_pbuf

The minimum number of pbufs that is added when a physical volume is added to any volume group. **Note:** Use the **ioo** command to change this value.

aio_cache_pbuf_count

The total number of pbufs that is allocated for aio_cache logical volume in the volume group.

workQ_size

The size of the hash table that is used to track I/O requests for logical volumes in the volume group.

Exit Status

This command returns zero for successful completion; otherwise it returns nonzero.

Security

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 that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To display the value for the `pv_pbuf_count`, type the following:

```
lvmo -v rootvg -o pv_pbuf_count
```

2. To set the `pv_pbuf_count` value to 2048, type the following:

```
lvmo -v rootvg -o pv_pbuf_count=2048
```

3. To generate pbuf and blocked I/O statistics, type the following:

```
lvmo -a
```

Location

`/usr/sbin/lvmo`

lvostat Command

Purpose

Reports input/output statistics for logical partitions, logical volumes and volume groups. Also reports pbuf and blocked I/O statistics.

Syntax

```
lvostat { -l | -v } Name [ -e | -d ] [ -F ] [ -C ] [ -c Count ] [ -s ] [ Interval [ Iterations ] ]
```

```
lvostat -v Name -r [ -L | -C ]
```

Description

The **lvostat** command generates reports that can be used to change logical volume configuration to better balance the input/output load between physical disks.

By default, the statistics collection is not enabled in the system. You must use the **-e** flag to enable this feature for the logical volume or volume group in question. Enabling the statistics collection for a volume group enables for all the logical volume in that volume group.

Note: The **-e** flag and the **-d** flag are not applicable for the space reclamation statistics specified by the **-r** flag.

The first report generated by **lvostat** provides statistics concerning the time since the system was booted. Each subsequent report covers the time since the previous report. All statistics are reported each

time **lvmstat** runs. The report consists of a header row followed by a line of statistics for each logical partition or logical volume depending on the flags specified.

If the **-l** flag is specified, *Name* is the logical volume name, and the statistics are for the physical partitions of this logical volume. The mirror copies of the logical partitions are considered individually for the statistics reporting. They are listed in descending order of number of i/os (*iocnt*) to the partition.

The *Interval* parameter specifies the amount of time, in seconds, between each report. The first report contains statistics for the time since the volume group startup, **varyonvg**. Each subsequent report contains statistics collected during the interval since the previous report. If the *Count* parameter is specified, only the top *Count* lines of the report are generated. For a logical volume if *Count* is 10, only the 10 busiest partitions are identified. If the *Iterations* parameter is specified in conjunction with the *Interval* parameter, then only that many iterations are run. If no *Iterations* parameter is specified, **lvmstat** generates reports continuously. If *Interval* is used to run **lvmstat** more than once, no reports are printed if the statistics did not change since the last run. A single period . (period) is printed instead.

The **lvmstat** command is useful in determining whether a physical volume is becoming a hindrance to performance by identifying the busiest physical partitions for a logical volume.

Note: The **lvmstat** commands reports I/O statistics of the local node only.

Input/Output Reports

The **lvmstat** command generates two types of reports, per partition statistics in a logical volume and per logical volume statistics in a volume group. The reports have the following format:

Column	Description
Log_part	Logical partition number
mirror#Log_part	Mirror copy number of the logical partition
iocntLog_part	Number of read and write requests
Kb_readLog_part	The total number of kilobytes read
Kb_wrtnLog_part	The total number of kilobytes written
KbpsLog_part	The amount of data transferred in kilobytes per second

Space reclaim statistics reports

The **lvmstat -r** command generates report for space reclaim statistics for physical volumes in the **volume** group. The reports have the following format:

Volume Group	Description
PV_name	Physical volume name
Reclaim	Space reclamation state. The possible state values are: on Space reclaim is supported for the physical volume. off Space reclaim is not supported for the physical volume. suspend Space reclaim is suspended by LVM configuration commands.
Mb_freed	Amount of physical partition space is freed from logical volume by commands like rmlv , rmlvcopy , and chfs in megabytes
Mb_pending	Space reclamation pending for the physical volume space in megabytes.
Mb_success	Space reclamation requests succeeded at disk driver in megabytes.
Mb_failed	Space reclamation requests failed by the disk driver in megabytes.

Volume Group	Description
Mb_reused	Free physical partition space reused for the logical volume without requesting the space reclamation in megabytes.
Mb_inprog	Amount of space reclaim request outstanding at the disk driver in megabytes.
io_count	Number of space reclaim I/O requests submitted to disk driver.
io_failed	Number of space reclaim I/O requests failed by disk driver.
io_misalign	Number of space reclaim requests reported as misaligned by the disk driver.
Mb_misalign	Amount of space reclaim failed by the disk driver due to misalignment in megabytes.
Mb_resubmit	Amount of space reclaim resubmitted due to reclaim block not being aligned on physical partition blocks.
num_pp_free	Number of physical partitions are freed by LVM commands like rmlv , rmlvcopy , chfs , and so on.
Kb_blksize	Space reclaim block size reported by the disk driver for alignment purpose.

Flags

Item	Description
-c Count	Prints only the specified number of lines of statistics.
-C	Causes the counters that keep track of the <code>iocnt</code> , <code>Kb_read</code> and <code>Kb_wrtn</code> be cleared for the specified logical volume/volume group. This flag can also be used to resets the space reclaim statistics.
-d	Specifies that statistics collection should be disabled for the logical volume/volume group in question.
-e	Specifies that statistics collection should be enabled for the logical volume/volume group in question.
-F	Causes the statistics to be printed colon-separated.
-l	Specifies the name of the stanza to list.
-L	Displays the space reclaim statistics in long listing mode.
-r	Prints the space reclaim statistics for all physical volumes in the volume group.
-s	Suppresses the header from the subsequent reports when <i>Interval</i> is used.
-v	Specifies that the <i>Name</i> specified is the name of the volume group.

Security

To use **lvmstat**, you must have root user authority.

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 the statistics collection for volume group `datavg` (all the LVs in `datavg` are enabled), enter the following command:

```
lvmstat -v datavg -e
```

2. To display the history of all the partitions of logical volume hd2, enter the following command:

```
lvmstat -l hd2
```

3. To display the history of top five logical volumes of volume group uservg, enter the following command:

```
lvmstat -v uservg -c 5
```

4. To display a continuous report at two second intervals for the logical volume ramlv, enter the following command:

```
lvmstat -l ramlv 2
```

5. To display six reports at two second intervals for the volume group rootvg, enter the following command:

```
lvmstat -v rootvg 2 6
```

6. To reset the counters for statistics for all the logical volumes in the volume group uservg, enter the following command:

```
lvmstat -v uservg -C
```

7. To disable statistics collection for datalv, enter the following command:

```
lvmstat -l datalv -d
```

8. To display statistics for space reclamation, enter the following command:

```
lvmstat -v uservg -r
```

9. To display statistics for space reclamation in long listing mode, enter the following command:

```
lvmstat -v uservg -r -L
```

10. To clear the statistics for space reclamation, enter the following command:

```
lvmstat -v uservg -r -C
```

Files

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

lvupdateInit Command

Purpose

Manages the list of entries to be added to the `/etc/inittab` file that is used to start the surrogate partition during the AIX Live Update operation.

Syntax

To add an entry to the `/etc/inittab` file in the surrogate partition, use the following syntax:

```
lvupdateInit -a [-i Identifier] { [ Identifier ] : [ RunLevel ] : [ Action ] : [ Command ] }
```

To list the entries to be added to the `/etc/inittab` file in the surrogate partition, use the following syntax:

```
lvupdateInit -l
```

To remove an entry from the list to be added to the `/etc/inittab` file in the surrogate partition, use the following syntax:


```
lvupdateInit -r -i Identifier
```

Description

The Live Update operation creates a customized root volume group (rootvg) to start the surrogate partition and prepare it for the Live Update operation. When the surrogate partition is started, it runs commands in a customized `/etc/inittab` file and other scripts like `/etc/rc.tcpip`. The `/etc/inittab` file is customized to exclude the services that are moved from the original partition during the mobility phase.

Some services that are running on the original partition might choose not to be check-pointed and keep running on the original partition. In such cases, this command can be used to add that service to the `/etc/inittab` file for the surrogate partition so that it is restarted during the Live Update operation. The `/etc/liveupdate/lvup_inittab` file contains a list of command entries to be added to the customized `/etc/inittab` file.

Parameters

Item	Description
<i>Identifier</i>	A 14-character parameter that uniquely identifies an object. The <i>Identifier</i> must be unique. If the <i>Identifier</i> is not unique, the command is unsuccessful.
<i>RunLevel</i>	A 20-character parameter that defines the run levels in which the <i>Identifier</i> parameter can be processed. Each process started by the init command can be assigned with one or more run levels in which it can be started.
<i>Action</i>	A 20-character parameter that informs the init command how to process the <i>Command</i> parameter that you specify. Refer to the mkinitab command for a list of valid actions that are recognized by the init command.
<i>Command</i>	A 1024-character field that specifies the shell command to be run.  Attention: To avoid possible corruption of system files, the <code>stdin</code> , <code>stdout</code> , and <code>stderr</code> files must be specified in the <i>Command</i> parameter with redirection, or they must be explicitly opened by the program being run by the command line.

Flags

Item	Description
-a	Adds an entry to the <code>/etc/liveupdate/lvup_inittab</code> file.
-i <i>Identifier</i>	Specifies an identifier for the new entry.
-r <i>Identifier</i>	Removes an entry from the <code>/etc/liveupdate/lvup_inittab</code> file.
-l	List the entries in the <code>/etc/liveupdate/lvup_inittab</code> file.

Examples

1. To add an entry to the `/etc/inittab` file that is used to start the Live Update operation in the surrogate partition, enter the following command:

```
# lvupdateInit -a -i myserver myserver:2:once:/opt/myapp/start_my_server
```

2. To remove an entry from the `/etc/inittab` file of the surrogate partition, enter the following command:

```
# lvupdateInit -r -i myserver
```

3. To list the entries to be incorporated into the `/etc/inittab` file of the surrogate partition, enter the following command:

```
# lvupdateInit -l
```

lvupdateRegKE Command

Purpose

The **lvupdateRegKE** command is used to register a command to be used to load a kernel extension on the surrogate logical partition (LPAR) during the AIX Live Update operation.

Syntax

```
lvupdateRegKE [ -a kext_path -c command | -r kext_path | -l ]
```

Description

The Live Update operation includes an opportunity to load specific kernel extensions before the workload resumes execution on the updated surrogate LPAR. This command manages the list of kernel extensions to be loaded on the surrogate partition when it is started. This command can be used to add a kernel extension to the list along with loading and configuring the kernel extension. It also provides options to remove a kernel extension from the list or to display the list of kernel extensions. The list is placed in the `/etc/liveupdate/lvup_preload_KE` file. To be loaded at the start of the surrogate LPAR, a kernel extension must be included in that file, and must be loaded on the original LPAR when the Live Update operation starts.

Note: The command that is to be registered with the **lvupdateRegKE** command must be present in one of the following file systems: `/`, `/var`, `/usr`, `/opt`, `/tmp`. Also, the kernel extension to be loaded by this command must be present in one of these file systems.

Parameters

Item	Description
<i>kext_path</i>	A string of up to 1024 characters that specifies the full path of a kernel extension.
<i>command</i>	A string of up to 1024 characters that specifies a command to be used to load a kernel extension on the surrogate LPAR before the applications are resumed.

Flags

Item	Description
-a <i>kext_path</i>	Adds a kernel extension to the list of kernel extensions to be loaded during the Live Update operation. Note: If the -a flag is specified, the -c flag is required.
-c <i>command</i>	Loads the kernel extension. Command arguments can be included by double-quoting the string.
-l	Lists the set of kernel extensions that are loaded during a Live Update operation and the commands that are specified to load them.
-r <i>kext_path</i>	Removes a kernel extension from the list.

Examples

1. To list all the commands that are registered to load kernel extensions during the Live Update operation, enter the following command:

```
# lvupdateRegKE -l
```

2. To register a command to load a kernel extension on the surrogate LPAR during the Live Update operation, enter the following command:

```
# lvupdateRegKE -a /usr/lib/drivers/mykext -c "/opt/myapp/bin/load_kext -x -y"
```

3. To remove a command that was previously registered to load a kernel extension during the Live Update operation, enter the following command:

```
# lvupdateRegKE -r /usr/lib/drivers/mykext
```

lvupdateRegScript Command

Purpose

The **lvupdateRegScript** command is used to register a script to be executed at notification points during an AIX Live Update operation.

Syntax

To register a script, use the following syntax:

```
lvupdateRegScript -a -n label -s script -d { orig | surr } -P phase -p priority
```

To unregister a script, use the following syntax:

```
lvupdateRegScript -r -n label -d { orig | surr } -P phase [ -p priority ]
```

To list all registered scripts, use the following syntax:

```
lvupdateRegScript -l
```

Description

The valid phases that a script can be registered for are as follows:

LVUP_CHECK

Executed at the beginning of a Live Update operation. This phase ensures the executed scripts that any associated services are ready for the Live Update operation.

LVUP_PRE

Executed before the applications are frozen on the original logical partition (LPAR). This phase allows the executed scripts to save any data as required before the applications are frozen.

LVUP_PRE_KERNEL

Executed after the applications are frozen on the original LPAR and before the applications are restarted on the surrogate LPAR. This phase is executed on the original LPAR before it is executed on the surrogate LPAR.

LVUP_POST

Executed on the surrogate LPAR after the applications are restarted. These scripts are executed in the `chroot` environment.

LVUP_ERROR

Executed in case of an error during the LVUP_CHECK phase or any later phase of the Live Update operation.

LVUP_COMPLETE

Executed on the surrogate LPAR after the Live Update operation is complete. This phase includes deletion of the original LPAR and resetting of the surrogate LPAR's UUID. The scripts that are registered for this phase are executed in the `chroot` environment.

During the Live Update operation, before the DR_CHECK scripts are invoked, the scripts that are registered with the LVUP_CHECK phase are executed. For the phases LVUP_CHECK, LVUP_PRE, LVUP_PRE_KERNEL, LVUP_POST, and LVUP_COMPLETE, the scripts are executed in a specific priority order that varies between the original and the surrogate logical partitions. On the original LPAR, the order is from priority 1 to priority 10, and the order is reversed on the surrogate LPAR, executing priority 10 first and on down to priority 1. In case of an LVUP_ERROR event, the scripts are executed in the opposite priority order.

The same methodology is applied to rest of the phases.

The script owner must specify if the script must be run on the original or the surrogate LPAR. If the script is to be run on both the original and the surrogate LPARs, it must be registered twice, once for the original LPAR and once for the surrogate LPAR.

When a registered script is executed, it must return 0 to indicate success, or a non-zero value if it failed. The Live Update operation fails if a script fails during the LVUP_CHECK or LVUP_PRE event.

Note: The script that is to be registered with the `lvupdateRegScript` command must be present in one of the following file systems: `/`, `/var`, `/usr`, `/opt`, `/tmp`.

Parameters

Item	Description
<i>label</i>	A string of up to 80 characters that specifies a label, which identifies a particular script.
<i>script</i>	A string of up to 1024 characters that specifies a script to be executed. The string must contain the full path to the script as well as any desired arguments.
<i>phase</i>	A string that specifies one of a set of phases: LVUP_CHECK, LVUP_PRE, LVUP_PRE_KERNEL, LVUP_POST, LVUP_ERROR, and LVUP_COMPLETE.
<i>priority</i>	An integer from 1 to 10 that identifies a priority for executing the script.

Flags

Item	Description
-a	Registers a script to be executed during the Live Update operation.
-d <i>orig</i> <i>surr</i>	Specifies the LPAR on which the script is to be executed: original LPAR (<i>orig</i>) or surrogate LPAR (<i>surr</i>).
-l	Lists the scripts and associated labels that are registered.
-n <i>label</i>	Specifies a label to associate with a registered script.
-P <i>phase</i>	Selects the phase when the script is to be invoked.
-p <i>priority</i>	Specifies a priority from 1 to 10. The scripts that are registered for a particular phase are executed in order from highest (1) to lowest (10) priority.
-r	Unregisters a script.
-s <i>script</i>	Specifies the script to be executed.

Examples

1. To list all the notification scripts that are registered to be invoked during the Live Update operation, enter the following command:

```
# lvupdateRegScript -l
```

2. To register a script to execute on the original LPAR during an LVUP_PRE event of the Live Update operation, enter the following command:
3. To register a script to execute on the surrogate LPAR during the LVUP_POST event of the Live Update operation, enter the following command:

```
# lvupdateRegScript -a -n putFiles -s "/opt/myapp/bin/rest_files /var/myapp/data"  
-P LVUP_POST -p 10 -d surr
```

4. To remove a script that was previously registered to be invoked during the Live Update operation, enter the following command:

```
# lvupdateRegScript -r -n getFiles -d orig -P LVUP_PRE
```

lvupdateSafeKE Command

Purpose

The **lvupdateSafeKE** command is a utility that manipulates the list of safe kernel extensions for the AIX Live Update operation.

Syntax

```
lvupdateSafeKE [ -a kext_path [-p] | -r kext_path | -l ]
```

Description

During a Live Update operation, a new logical partition (LPAR) is dynamically created and booted with the updated AIX kernel. When the new LPAR is booted, the device drivers are reloaded while the devices are configured. The **lvupdateRegKE** command can be used to specify commands to be executed during a Live Update operation to load other kernel extensions. After the new LPAR is prepared, the active workload from the original LPAR is checkpointed and restarted from the same point on the new LPAR.

By default, any data from the kernel extensions is not checkpointed. If there is no state data in a kernel extension, reloading the kernel extension on the new LPAR does not cause any problems while the applications are checkpointed and restarted. This condition makes a kernel extension Live Update *safe*. A kernel extension can be made safe by registering scripts to be executed during the Live Update operation that either capture any necessary state information and restore it on the new LPAR, or quiesce any subsystems that are necessary to ensure that there is no state data in the kernel extension.

A kernel extension that is in the safe kernel list ensures that if it is loaded, it does not prevent a Live Update operation. If a Live Update operation fails because of a loaded kernel extension that is not in the safe list, an error with the kernel extension name is logged under the `/var/adm/ras/liveupdate/logs` directory. Kernel extensions can also be marked safe by specifying the **SYS_LUSAFE** flag when you load the extension by using the `sysconfig()` system call. The **lvupdateSafeKE** command does not list the extensions that were marked safe by using this method.

To bypass the check for safe kernel extensions, the Live Update operation must be started with an `lvupdate.data` file that has the entry, `kext_check = no`.

Note: Any kernel extension that is to be loaded during the Live Update operation must be in one of the following five file systems: `/`, `/var`, `/usr`, `/opt`, `/tmp`.

Parameters

Item	Description
<code>kext_path</code>	A string of up to 1024 characters that specifies the full path of a kernel extension.

Flags

Item	Description
<code>-a kext_path</code>	Adds a kernel extension to the list that indicates that it is safe for the Live Update operations.
<code>-l</code>	Lists the set of kernel extensions that are specified as safe for the Live Update operation.
<code>-p</code>	Specify the <code>-p</code> flag with the <code>-a</code> flag to indicate that the kernel extension can be ignored while checking preview. However, do not use this flag during the Live Update operation.
<code>-r kext_path</code>	Removes a kernel extension from the list.

Examples

1. To add a kernel extension to the safe list for the Live Update operation, enter the following command:

```
# lvupdateSafeKE -a /usr/lib/drivers/mydev_driver
```

2. To list the kernel extensions that are registered as safe, enter the following command:

```
# lvupdateSafeKE -l
```

3. To remove a kernel extension from the safe list, enter the following command:

```
# lvupdateSafeKE -r /usr/lib/drivers/mydev_driver
```

lvupdateSetProcs Command

Purpose

To add, remove, or list entries in the base process list that is used for the AIX Live Update operations.

Syntax

```
lvupdateSetProcs -b [ -n label -a command | -n label -r | -l ]
```

Description

Base processes are not checkpointed during a Live Update operation. These processes are left unchanged on the original logical partition (LPAR), rather than being migrated to the surrogate LPAR. This command provides a mechanism to manage the list of base processes. The list of base processes is placed in the `/etc/liveupdate/lvup_BaseProcs` file.

Parameters

Item	Description
<i>command</i>	A string of up to 1024 characters that specifies an executable file, which includes a full path.
<i>label</i>	A string of up to 80 characters that specifies a label, which is associated with a particular command that executes as a base process.

Flags

Item	Description
-a <i>command</i>	Adds the command to the specified process list. The <i>command</i> specified with the -a flag must execute as a direct child of the <code>init</code> process.
-b	Designates the lvupdateSetProcs command to work on the base process list.
-l	Lists the commands from the specified process list.
-n <i>label</i>	Specifies a label to associate with a command that is being added or removed.
-r <i>command</i>	Removes the command from the specified process list.

Examples

1. To add an entry to the base process list that is used by the Live Update operation, enter the following command:

```
# lvupdateSetProcs -b -n myserv -a /usr/sbin/mysevice
```

2. To list the commands that are registered as base processes, enter the following command:

```
# lvupdateSetProcs -bl
```

m

The following AIX commands begin with the letter *m*.

m4 Command

Purpose

Preprocesses files, expanding macro definitions.

Syntax

```
m4 [ -e ] [ -l ] [ -s ] [ -B Number ] [ -D Name [ =Value ] ] ... [ -H Number ] [ -I Directory ] [ -S Number ] [ -T Number ] [ -U Name ] ... [ File ... ]
```

Description

The **m4** command is a macro processor used as a preprocessor for C and other languages. You can use it to process built-in macros or user-defined macros.

Each *File* parameter is processed in order. If you do not specify a *File* parameter or if you specify the - (dash) as a file name, the **m4** command reads standard input. It writes the processed macros to standard output. Macro calls follow the form:

```
macroname(argument . . . )
```

The left parenthesis must immediately follow *macroname*. If the left parenthesis does not follow the name of a defined macro, the **m4** command reads it as a macro call with no arguments. Macro names consist of ASCII alphabetic letters, digits, and the _ (underscore) character. Extended characters are not allowed in macro names. The first character cannot be a digit.

While collecting arguments, the **m4** command ignores unquoted leading blanks, tabs, and new-line characters. Use single quotation marks to quote strings. The value of a quoted string is the string with the quotation marks stripped off.

When the **m4** command recognizes a macro, it collects arguments by searching for a matching right parenthesis. If you supply fewer arguments than those that appear in the macro definition, the **m4** command considers the trailing arguments in the definition to be null. Macro evaluation proceeds normally during the collection of arguments. All commas or right parentheses within the value of a nested macro call are translated literally; they do not need an escape character or quotation marks. After collecting arguments, the **m4** command pushes the value of the macro back onto the input stream and scans again.

Built-in Macros

The **m4** command makes available the following built-in macros. You may redefine them, but you will lose the original meaning. The values of these macros are null unless otherwise stated:

Item	Description
define (<i>Name,NewName</i>)	Replaces the macro <i>Name</i> with the value of <i>NewName</i> . The <i>NewName</i> string can take the form <i>\$n . . .</i> (where <i>n</i> is a digit). In this case, each occurrence of <i>n</i> in the replacement text is replaced by the <i>n</i> th argument of <i>Name</i> . \$0 is the name of the macro. The null string replaces missing arguments. The number of arguments replaces \$# . A comma-separated list of all arguments replaces \$* . \$@ acts like \$* , but each argument is quoted with the current quotation character (see changequote).
undefine (<i>Name</i>)	Removes the definition of <i>Name</i> .
defn (<i>Name . . .</i>)	Returns the quoted definition of <i>Name</i> .
pushdef (<i>Name, NewName</i>)	Redefines <i>Name</i> with <i>NewName</i> as in define , but saves any previous definition.
popdef (<i>Name . . .</i>)	Removes the current definition of <i>Name</i> and returns to the previous definition, if one existed.
ifdef (<i>Name,True,[False]</i>)	Returns the value of <i>True</i> only if <i>Name</i> is defined, otherwise returns <i>False</i> . If you do not supply <i>False</i> , its value is null. Note: The behavior of ifdef has changed to comply with the Single UNIX Specification, Version 10. The previous behavior of ifdef returns the value of <i>True</i> only if <i>Name</i> is defined and is not defined as 0. By default, ifdef works like it did before UNIX10. The UNIX 10 behavior can be obtained by setting the environment variables <i>XPG_SUS_ENV</i> to <i>ON</i> and <i>XPG_UNIX98</i> to <i>OFF</i> .
shift (<i>Argument . . .</i>)	Returns all but the first argument. The other arguments are quoted and pushed back with commas in between. The quoting nullifies the effect of the extra scan that is subsequently performed.
changequote (<i>L,R</i>)	Changes quote symbols to <i>L</i> and <i>R</i> . The symbols can be up to 5 bytes long. changequote without arguments restores the original values (` `).
changecom (<i>L,R</i>)	Changes left and right comment markers from the default # and new-line character to <i>L</i> and <i>R</i> . With no arguments, the comment mechanism is disabled. With one argument, the left marker becomes the parameter and the right marker becomes a new-line character. With two arguments, both markers are affected. Comment markers can be up to 5 bytes long.
divert (<i>Number</i>)	Changes the current output stream to stream <i>Number</i> . There are 10 output streams, numbered 0-9. The final output is the concatenation of the streams in numerical order. Initially, stream 0 is the current stream. The m4 command discards output diverted to a stream other than 0-9.
undivert (<i>Number . . .</i>)	Causes immediate output of text from the specified diversions (or all diversions if there is no argument). Text may be undiverted into another diversion. Undiverting discards the diverted text.
divnum	Returns the value of the current output stream.
dnl	Reads and discards characters up to and including the next new-line character.

Item	Description
ifelse (<i>String1</i> , <i>String2</i> , <i>True</i> , [<i>False</i>]) . . .)	If <i>String1</i> and <i>String2</i> are the same then the value is <i>True</i> . If they are not and if there are more than four arguments, the m4 command repeats the process with the additional arguments (4, 5, 6, and 7). Otherwise, the value is either <i>False</i> or null if you provide no value for <i>False</i> .
incr (<i>Number</i>)	Returns the value of its argument incremented by 1.
decr (<i>Number</i>)	Returns the value of its argument decreased by 1.
eval (<i>Expression</i> [, <i>Number1</i> [, <i>Number2</i>])	Evaluates its first argument as an arithmetic expression, using 32-bit signed arithmetic. The operators you can use are +, -, *, /, %, ^ (exponentiation), bitwise &, , ~, and ^ relationals, and parentheses. Octal and hex numbers can be specified as in C. <i>Number1</i> specifies the radix for the result of the expression. The default radix is 10. The optional <i>Number2</i> specifies the minimum number of digits in the result. Note: The behavior of eval has changed to comply with the Single UNIX Specification, Version 10. The previous behavior of eval evaluates its first argument as an arithmetic expression, using 32-bit unsigned arithmetic. By default, eval works like it did before UNIX 10. The UNIX 10 behavior can be obtained by setting the environment variables <i>XPG_SUS_ENV</i> to <i>ON</i> and <i>XPG_UNIX98</i> to <i>OFF</i> .
len (<i>String</i>)	Returns the number of bytes in <i>String</i> .
dlen (<i>String</i>)	Returns the number of displayable characters in <i>String</i> ; that is, two-byte extended characters are counted as one displayable character.
index (<i>String1</i> , <i>String2</i>)	Returns the position in the <i>String1</i> string where the <i>String2</i> string begins (zero origin), or -1 if the second parameter does not occur.
substr (<i>String</i> , <i>Position</i> , [<i>Number</i>])	Returns a substring of <i>String</i> . The beginning of the substring is selected with <i>Position</i> , and <i>Number</i> indicates the length of the substring. Without <i>Number</i> , the substring includes everything to the end of the first string.
translit (<i>String</i> , <i>From</i> , <i>To</i>)	Transliterates the characters in <i>String</i> from the set given by <i>From</i> to the set given by <i>To</i> . No abbreviations are permitted. Two-byte extended characters are correctly mapped into the corresponding replacement characters.
include (<i>File</i>)	Returns the contents of <i>File</i> or displays an error message if it cannot access the file.
sinclude (<i>File</i>)	Returns the contents of <i>File</i> , but it gives no error message if <i>File</i> is inaccessible.
syscmd (<i>Command</i>)	Runs the <i>Command</i> . No value is returned.
sysval	Returns the return code from the last call to syscmd .
maketemp (. . . <i>nnnn</i> . . .)	Replaces <i>nnnn</i> in its argument with the current process ID number.

Item	Description
mkstemp (<i>template</i>)	The string in the <i>template</i> argument is a file name with at least six trailing X characters. The mkstemp macro replaces each X character in the file name with a character from the character set of the portable file name. A file is created with the newly created file name and is closed. If the mkstemp macro contains an empty string as the <i>template</i> argument, the file is not created by using the newly created file name and the m4 command writes a diagnostic message to the standard output. Also, if the mkstemp macro contains an empty string as the <i>template</i> argument, the m4 command continues to process the input even if the file is not created and returns a non-zero value as the exit status.
m4exit (<i>Value</i>)	Exits from m4 immediately, returning the specified exit <i>Value</i> (the default is 0).
m4wrap (<i>LastMacro</i>)	Runs <i>LastMacro</i> after reading the end-of-file character. For example, <code>m4wrap (`cleanup () `)</code> runs the <code>cleanup</code> macro at the end of m4 .
errprint (<i>Message</i>)	Includes <i>Message</i> on the diagnostic output file.
dumpdef ([<i>Name . . .</i>])	Writes to standard output the current names and definitions for the named items or for all if no arguments are provided.
traceon (<i>Macro</i>)	Turns on tracing for <i>Macro</i> . If none is named, tracing is turned on for all macros.
traceoff (<i>Macro . . .</i>)	Turns off trace globally and for any <i>Macro</i> specified. Macros specifically traced by traceon can be untraced only by specific calls to traceoff .

Flags

Item	Description
-B <i>Number</i>	Makes the <i>Number</i> variable the size of the push-back and parameter collection buffers (the default is 4096).
-e	Operates interactively. Interrupts are ignored and the output is not buffered.
-H <i>Number</i>	Makes the <i>Number</i> variable the size of the symbol table hash array (the default is 199). The size must be a prime number.
-I <i>Directory</i>	(Uppercase i) Searches the <i>Directory</i> variable first, then searches the directories on the standard list for include (built-in macro) files with names that do not begin with a / (slash).
-l	(Lowercase L) Enables line-numbering output for the assembler (<code>.xline . . .</code>).
-s	Enables the line-sync output for the C preprocessor (<code>#line . . .</code>).
-S <i>Number</i>	Makes the <i>Number</i> variable the size of the call stack (the default is 100 slots). Macros take three slots, and non-macro arguments take one.
-T <i>Number</i>	Makes the <i>Number</i> variable the size of the token buffer (the default is 512 bytes).

Note: All flag options can be interspersed with operands.

The preceding flags must appear before any file names and before any **-D** or **-U** flags.

Item	Description
-D <i>Name</i> [= <i>Value</i>]	Defines the <i>Name</i> variable as the <i>Value</i> variable. If the <i>Value</i> variable is not specified, the <i>Name</i> variable becomes null.
-U <i>Name</i>	Undefines a the <i>Name</i> variable previously defined with the -D flag.

Exit Status

This command returns the following exit values:

Ite	Description
m	
0	Successful completion.
>0	An error occurred.

If the **m4exit** macro is used, the exit value can be specified by the input file.

Examples

To preprocess a C language program with the **m4** command and compile it, enter:

```
m4 prog.m4 > prog.c
cc prog.c
```

Files

Item	Description
/usr/ccs/bin/m4	Contains the m4 command.

mach Command

Purpose

Displays the processor type of the current host .

Syntax

mach

Description

The **mach** command displays the architecture of the system processor.

Exit Status

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

Examples

- To display the processor type of the current host use the **mach** command in the following way:

```
mach
```

Files

Item	Description
<code>/usr/bin/mach</code>	Contains the System V mach command.

machstat Command

Purpose

Reports the value of the first 4 bits of the power status register.

Syntax

```
machstat { -p | -f }
```

Description

The **machstat** command returns the value of a status register. There is no standard output or error except when using the `-f` flag on CHRP hardware.

Flags

Item	Description
------	-------------

- | | |
|-----------------|---|
| <code>-f</code> | On non-CHRP machines, returns Power Status Register bits 10–13. On CHRP machines, displays the EPOW status, EPOW modifier, and, if present, EPOW version. |
| <code>-p</code> | Returns the first 4 bits of the power status register. |

Exit Status

The **machstat** command returns a value of 255 if an error occurs. Otherwise it returns the value of the register.

Security

Access Control: root only

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 see the current value of the power status register, enter:

```
machstat -p
echo $?
```

Files

Item	Description
<code>/etc/rc.powerfail</code>	Shuts down a system when a power failure is detected

macref Command

Purpose

Produces a cross-reference listing of macro files.

Syntax

```
macref [ -n ] [ -s ] [ -t ] [ - ] [ File ... ]
```

Description

The **macref** command reads the named English-language files (which are assumed to be **nroff** or **troff** command input) and produces a cross-referenced listing of the symbols in the input.

The default output is a list of the symbols found in the input, each accompanied by a list of all references to that symbol. The **macref** command lists the symbols alphabetically in the left column, with references following to the right. Each reference is given in the following form:

```
[ [ ( NMName ) ]  
  MName- ]  
  Type LNumber  
[ # ]
```

Generated names are listed under the artificial symbol name ~sym.

Input Parameters

Item	Description
------	-------------

<i>File</i>	Specifies the nroff or troff file from which the macref command produces output containing a list cross-referencing macros.
-------------	--

Output Parameters

Item	Description
------	-------------

<i>NMName</i>	The name of the macro within which MName is defined.
---------------	--

<i>MName</i>	The name of the macro within which the reference occurs. This field is not present if the reference occurs outside a macro.
--------------	---

Item	Description
<i>Type</i>	The type associated, by context, with this occurrence of the symbol. The types can be the following: <ul style="list-style-type: none"> r Request m Macro d Diversion s String n Number register p Parameter. For instance, <code>\\$x</code> is a parameter reference to <code>x</code>. <ul style="list-style-type: none"> Note: Parameters are never modified, and the only valid parameter symbol names are 1, 2, . . . 9.
<i>LNumber</i>	The line number on which the reference occurred.
<i>#</i>	This reference modifies the value of the symbol.

Flags

Ite	Description
m	
-n	Causes one line to be printed for each reference to a symbol.
-s	Causes symbol-use statistics to be printed.
-t	Causes a macro table of contents to be printed.

The flags can be grouped behind one - (minus sign). Use a — (dash) to delimit the end of flags.

Note: The **macref** command does not accept - as standard input.

Files

Item	Description
<code>/tmp/macref.tXXXXXX</code>	Contains a temporary file.
<code>/tmp/macref.sXXXXXX</code>	Contains a temporary file.
<code>/tmp/macref.cXXXXXX</code>	Contains a temporary file.

mail, Mail, or mailx Command

Purpose

Sends and receives mail.

Syntax

To Read Incoming Mail

mail -e

mail -f [**-dlHNn**] [**-F**] [*FileName*]

mail [**-dlHNn**] [**-F**] [**-u** *UserID*]

To Send Mail

mail [**-s** *Subject*] [**-c** *Address(es)*] [**-dinNv**] *Address*

Description

The **mail** command invokes the mail utility, enabling you to:

- Read incoming mail.
- Send mail.

In addition, you can use the available options and subcommands to customize the way you send and receive mail.

The **mail** command operates on two types of mailboxes, the system mailbox and the personal mailbox.

Incoming mail is stored in the system mailbox. By default, a user's system mailbox is a file located in the **/var/spool/mail** directory. The mailbox file is named after the userID. For example, if your user ID is *jeanne*, then your system mailbox is **/var/spool/mail/jeanne**.

By default, when a user has read, deleted, or saved all the mail in their system mailbox, the mailbox is deleted. To prevent the mailbox from being deleted, use the **set** subcommand to set the **keep** option.

In addition to the system mailbox, there is the user's personal mailbox. As messages are read, if they are not deleted or saved to a file, they will be marked to be moved to the personal mailbox. The personal mailbox, by default, is **\$HOME/mbox**. For example, if your home directory is **/home/lance**, then **/home/lance/mbox** is your personal mailbox. The messages remain in your personal mailbox until you move them to a folder or delete them.

Folders provide a way to save messages in an organized fashion. You can create as many folders as you need. Name each folder with a name that pertains to the subject matter of the messages it contains.

Notes:

- Results can be unpredictable when you run multiple instances of the **mail** command on one mailbox.
- Although the command names are different, the **mail**, **Mail**, or **mailx** command provides identical functionality.

Examining the Contents of Your Mailbox

To process your mail, type **mail** at the system prompt. The Mail program displays a one-line entry for each piece of mail in your system mailbox:

```
Mail [5.2 UCB] [AIX 7.1] Type ? for help.
"/var/spool/mail/lance": 2 messages 2 new
>N 1 karen          Thu Sep 17 14:36 13/359 "Dept Meeting"
  N 2 lance@zeus    Thu Sep 17 15:06 10/350 "Delay"
  N 3 karen         Thu Sep 17 14:36 13/359 "Meeting Cancel"
```

The current message is marked by a > at the beginning of a line in the header summary.

Each one-line entry displays the following fields:

Item description of mailbox

Item	Description
status	Indicates the current class of a piece of mail. The status can be any of the following: N A new message P A message to be preserved in system mailbox. U An unread message. An unread message is a message that was listed in the mailbox last time you invoked the Mail program, but whose contents you did not examine. * A message that was saved or written to a file or folder. A message without a status indicates that the message has been read but has not been deleted or saved.
number	Identifies the numerical order of the message.
sender	Identifies the address of the person who sent the mail.
date	Specifies the date the message was received.
size	Defines the number of lines and characters contained in the letter (this includes the header).
subject	Identifies the subject of the message.

Finally, following the list of mail, the Mail program displays the mailbox prompt, which by default is **?**, to indicate that it is waiting for input.

Flags

Flags Description

Item	Description
-c <i>Address(es)</i>	Specifies the list of users to which a copy of the message is sent. You can specify one or more addresses. When specifying more than one address, the list of addresses must be in (" ") quotes.
-d	Specifies the debug information associated with the users mailbox <ul style="list-style-type: none">• uid• user name• mail file folder (the system mailbox)• dead letter (the system saves incomplete messages in the dead.letter file in the \$HOME)• mbox (the personal mailbox) Note: The message is not sent when the program is in the debug mode.
-e	Tests for the presence of mail in the system mailbox. The mail utility will write nothing and exit with a successful return code if there is mail to read.
-f <i>FileName</i>	Reads messages from the named file. If a file operand is not specified, then reads messages from mbox . When you quit from reading the messages, undeleted messages are written back to this file.

Flags Description (continued)

Item	Description
-F	Records the message in a file named after the recipient. The name is the portion of the address found first on the To: line in the mail header. Overrides the record variable if set.
-H	Writes a header summary only.
-i	Causes tty interrupt signals to be ignored.
-n	Inhibits reading the /usr/share/lib/Mail.rc file.
-l	Expands the From User field to 256 characters to handle the long user names.
-N	Suppresses the initial printing of headers.
-s Subject	Specifies a subject for a message to be created.
-u UserID	Specifies an abbreviated equivalent of doing mail -f /var/spool/mail/UserID . Starts the Mail program for a specified user's mailbox. You must have access permission to the specified mailbox.
-v	Puts the Mail program into verbose mode. Displays the details of delivery on the user's terminal.

Environmental Variables

The following environment variables affect the execution of mail:

Description of environment variables

Item	Description
DEAD	Pathname of the file in which to save partial messages in case of interrupts or delivery errors.
EDITOR	Pathname of the editor to use when the edit or ~e command is used.
HOME	Pathname of the user's home directory.
LISTER	String representing the command for writing the contents of the folder directory to standard output when the folders command is given. Any string acceptable as a command_string operand to the sh -c command is valid. If this variable is null or not set, the output command will be <i>ls</i> . The default value is unset.
MAILBOX	Specifies the location of the system mailbox for the mail command. The MAILBOX value is where the mail command searches for mail messages. The system default value if the MAILBOX environment variable is not specified is the /var/spool/mail directory.
MAILRC	Pathname of your personal startup file. The default is \$HOME/.mailrc .
MBOX	Pathname of your personal mailbox where messages are saved from the system mailbox that have been read. The exit command overrides this function, as will saving the message explicitly in another file. The default is \$HOME/mbox .
PAGER	String representing an output filtering or pagination command for writing the output to the terminal. Any string acceptable as a command_string operand to the sh-c command is valid. When standard output is a terminal device, the message output will be piped through the command if the mail internal variable crt is set to a value less the number of lines in the message. If the PAGER variable is null or not set, the paginator is the pg shell command.
SHELL	Pathname of a preferred command interpreter.
VISUAL	Pathname of a utility to invoke when the visual command or ~v command-escape is used. If this variable is not set, the full screen editor will be <i>vi</i> .

Internal Variables in Mail

Internal variables in a mail

Item	Description
allnet	Treats all network names, whose login name components match, identically. Causes the msglist message specifications to behave similarly. The default is noallnet .
append	Adds the message saved in your mailbox to the end, rather than the beginning, of the \$HOME/mbox file. The default is noappend .
ask, asksub	Prompts for the subject of each message if not specified on the command line with the -s option. If you do not wish to create a subject field, press Enter at the prompt. It is not possible to set both ask and noasksub , or noask and asksub . The default is asksub .
askbcc	Prompts for the addresses of people to add to the blind copy list. If you do not wish to send blind copies, press Enter at the prompt.
askcc	Prompts for the addresses of people who should receive copies of the message. If you do not wish to send copies, press Enter at the prompt.
autoprint	Sets the delete subcommand to delete the current message and display the next message.
crt	Specifies the minimum number of lines that a message must contain before any output filtering or pagination is used when the message is displayed.
debug	Displays debugging information. Messages are not sent while in debug mode. This is the same as specifying the -d flag on the command line.
dot	Interprets a period entered on a line by itself as the end of a message you are sending.
escape=c	Sets the command escape character to be the character <i>c</i> . By default the command escape character is ~ (tilde).
Replyall, flipr	Reverses the meanings of the Respond and respond or Reply and reply commands. The default is noflipr .
folder=directory	The directory name in which to store mail folders. After the directory is defined, you can use the + (plus sign) notation to refer to it when using the <i>FileName</i> parameter with mailbox subcommands.
header	Enables writing of the header summary when entering mail in receive mode. The default is header .
hold	Holds messages that you have read but have not deleted or saved in the system mailbox instead of in your personal mailbox. The default is nohold .
ignore	Ignores interrupts while entering messages. Echoes interrupts as @ (at) characters.
ignoreeof	Sets the mail command to refuse the Ctrl+D key sequence as the end of a message. Input can be terminated only by entering a period . (period) on a line by itself or by the ~. command escape. The default is noignoreeof .

Item	Description
indentprefix = <i>string</i>	A string that will be prefixed to each line that is inserted into the message by the ~m command escape. This variable defaults to one tab character.
keep	When a system mailbox, secondary mailbox, or mbox is empty, truncate it to zero length instead of removing it. The default is nokeep .
keepsave	Keep messages that have been saved with the (s)ave or (w)rite subcommands in the system mailbox instead of deleting them. The default is nokeepsave .
metoo	Includes the sender in the alias expansion if sender's name is part of the alias. By default, expanding the alias removes the sender.
onehop	When responding to a message that was originally sent to several recipients, the other recipient addresses are usually forced to be relative to the originating author's machine for the response. This flag disables alteration of the recipient's addresses, improving efficiency in a network where all machines can send directly to all other machines (that is, one hop away). The default is noonehop .
outfolder	Causes the files used to record outgoing messages to be located in the directory specified by the folder variable unless the pathname is absolute. The default is nooutfolder . See the record and folder variables.
page	Insert a form-feed after each message sent through the pipe created by the pipe command. The default is nopage .
prompt = <i>string</i>	Set the command-mode prompt to <i>string</i> . If <i>string</i> is null or if noprompt is set, no prompting will occur. The default is to prompt with the "?" string.
quiet	Refrain from writing the opening message and version when entering mail. The default is noquiet .
record = <i>file</i>	Defines a file in which to record all outgoing mail. The default is norecord .
save	Enables saving of messages in the dead.letter file on interrupt or delivery error. The default is save .
screen = <i>number</i>	Sets the number of lines in a screenful of headers for the headers and z commands.
sendmail = <i>shell_command</i>	Alternative command for delivering messages.
sendwait	Wait for the background mailer to finish before returning. The default is nosendwait .
showto	When the sender of the message was the user who is invoking mail, write the information from the To: line instead of the From: line in the header summary. The default is noshowto .
sign = <i>string</i>	Inserts <i>string</i> into the text of a message when the ~a command escape is given. The default is nosign . The character sequences /t and /n are recognized in the string as tab and newline characters, respectively.

Internal variables in a mail (*continued*)

Item	Description
Sign =string	Inserts <i>string</i> into the text of a message when the -A command escape is given. The default is noSign .
toplines =number	Number of lines displayed by the top subcommand.
verbose	Displays the actual delivery of messages on the terminal. This is the same as specifying the -v flag on the command line.

Setting Environment Variables

The Bourne shell (**bsh** command) uses and checks the following variables. These variables can be set in **\$HOME/.profile**.

Item descriptions of Bourne shell

Item	Description
MAIL	Specifies the location and name of the user's system mailbox that is checked by the Bourne shell to determine whether or not you have mail. If the system mailbox is not empty, the Bourne shell sends a message that you have new mail. The Bourne shell checks the system mailbox periodically based on the value of the MAILCHECK environment variable.
MAILCHECK	Specifies the interval at which the Bourne shell checks the system mailbox for mail.
MAILMSG	Specifies the message sent to your console shell by the system when you have mail. The default message is similar to the following:

```
YOU HAVE NEW MAIL
```

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 the Mail program and list the messages in your mailbox, type the following at the command line prompt:

```
mail
```

The **mail** command lists every messages in your system mailbox. The mail system then displays the mailbox prompt (?) to indicate *waiting for input*. When you see this prompt, enter any mailbox subcommand.

To see a list of subcommands, type:

```
?
```

This entry lists the Mail subcommands.

2. To send the message **letter** to the recipient **user1@host1** and copies to **user2@host2** and **user3@host3**, type:

```
mail -c "user2@host2 user3@host3" user1@host1<letter
```

3. To look at the contents of your personal mailbox, type:

```
mail -f
```

This command displays a list of the messages in your personal mailbox, **\$HOME/mbox**.

4. To look at the contents of a specific mail folder, type:

```
mail -f +dept
```

This command displays a listing of the messages in the dept folder.

5. To send a message to a user on your local system, type:

```
mail ron
```

When you finish typing the message to user ron, press Enter and press either . (period) or Ctrl+D to exit the editor and send the message. To determine if a user is on your local system, check for the user's name in your **/etc/passwd** file.

If your message is delivered successfully, you receive no notification. If your message could not be delivered, an error message is sent to you.

6. To mail a file to another user on your local system, type:

```
mail karen < letter1
```

This command sends the contents of the file letter1 to user karen on your local system. After the command sends the file, the Mail program displays the command line prompt.

7. To send a message to a user on a remote system, type:

```
mail dale@zeus
```

You now can create a message to dale. In this example, you are sending a message to user dale on remote system zeus. To send a message to a user on another system connected to your system through a network, you must know that person's login ID and the name of the other system.

Mailbox Subcommands for the mail, Mail, and mailx Command

From the mail prompt, ? (question mark), you can enter subcommands to manipulate mail in your mailbox. Subcommands that work on more than one message at a time use the *MessageList* parameter. Subcommands that work with files or folders use the *FileName* parameter. These parameters are discussed in [Mail command and subcommands in Networks and communication management](#).

The following list describes the Mailbox subcommands and their functions:

Mailbox subcommands

Item	Description
=	Echoes the number of the current message.
#	Comment character for writing comments in mail script files.
-	Displays the previous message.
?	Displays a brief summary of mailbox subcommands. Identical to the help subcommand.
! <i>Command</i>	Executes the workstation shell command specified by <i>Command</i> .
alias	(a) With no arguments, displays all currently defined aliases and their corresponding addresses. With one argument, displays one alias. With more than one argument, creates a new alias or changes an old alias. Identical to the group subcommand.

Item	Description
alternates <i>AlternatesList</i>	(alt) The alternates subcommand is useful if you have accounts on several machines. Use the subcommand to inform the Mail program that the addresses listed in <i>AlternatesList</i> all refer to you. When you use the reply subcommand to reply to messages, the Mail program does not send a copy of the message to any of the addresses given in <i>AlternatesList</i> . If you enter the alternates subcommand with no argument, the Mail program displays the current set of alternate names.
chdir <i>Directory</i>	(cd) Changes your working directory to the indicated <i>Directory</i> . If no directory is given, it changes to your login directory.
copy [<i>MessageList</i>] <i>File</i>	(c, co) Appends each message in <i>MessageList</i> to the end of <i>File</i> . Displays the file name in quotes, followed by the line count and character count, on the user's terminal. Does not delete any messages when you quit.
Copy [<i>MessageList</i>]	(C) Saves the specified message in a file whose name is derived from the author of the message to be saved, without marking the messages as saved. Otherwise, it is equivalent to the Save subcommand.
delete [<i>MessageList</i>]	(d) Marks the messages in <i>MessageList</i> to be deleted when you quit the Mail program. Entering the d subcommand without a message list deletes the current message. Deleted messages are not saved in your \$HOME/mbox file nor are they available for most other commands. However, you can use the undelete subcommand to restore messages you have deleted while in the same mailbox session. If you delete a message and either change to another mailbox or quit the mailbox with the quit subcommand, the deleted message cannot be recalled.
discard [<i>FieldList</i>]	(di) Identical to the ignore subcommand. Note: The retain subcommand overrides the discard subcommand.
dp	Deletes the current message and displays the next message. If there is no next message, the Mail program displays EOF. Identical to the dt subcommand.
dt	Deletes the current message and displays the next message. If there is no next message, the Mail program displays EOF. Identical to the dp subcommand.
echo <i>String</i>	Displays the character string <i>String</i> on the command line.
edit [<i>MessageList</i>]	(e) Starts the alternate editor using the <i>MessageList</i> as input files. To define an alternate editor, use the set EDITOR= statement or place an entry in your \$HOME/.mailrc file. Any message specified by the <i>MessageList</i> parameter retains the changes made during the editor session.
exit	(ex or x) Leaves the mailbox and returns to the operating system without changing the original contents of the mailbox. The mailbox returns to the condition that it was when the Mail program was started. Messages marked to be deleted are not deleted. Identical to the xit subcommand.
file [<i>Name</i>]	(fi) Identical to the folder subcommand.

Item	Description
folder [<i>Name</i>]	<p>(fo) Switches to a new mail file or folder. With no arguments, the subcommand displays the name of the current mailbox. If an argument is included, it stores the current mailbox with changes (such as messages deleted) and reads in the new mailbox specified by the <i>Name</i> parameter. Identical to the file subcommand.</p> <p>Some special conventions are recognized for the <i>Name</i>:</p> <p># Refers to the previous file.</p> <p>% Refers to the system mailbox (/var/spool/mail/UserID).</p> <p>& Refers to your personal mailbox (\$HOME/mbox).</p> <p>+Name Refers to a file in your folder directory.</p>
folders	Lists the names of the folders in your folder directory.
followup [<i>message</i>]	(fo) Responds to a message, recording the response in a file whose name is derived from the author of the message. Overrides the record variable, if set.
Followup [<i>MessageList</i>]	(F) Responds to the first message in the <i>msglist</i> , sending the message to the author of each message in the <i>msglist</i> . The subject line is taken from the first message and the response is recorded in a file whose name is derived from the author of the first message.
from [<i>MessageList</i>]	(f) Displays the headings of messages in <i>MessageList</i> .
group	(g) Identical to the alias subcommand.
headers [<i>Message</i>]	(h) Lists the headings in the current group of messages (each group of messages contains 20 messages by default; change this with the set screen= statement). If the mailbox contains more messages than can be displayed on the screen at one time, information about only the first group of messages will be displayed. To see information about the rest of the messages, use the h subcommand with a message number in the next range of messages, or use the z subcommand to change the current message group.
help	Displays a brief summary of mailbox subcommands. Identical to the ? subcommand.
hold [<i>MessageList</i>]	(ho) Marks each message in <i>MessageList</i> to be saved in your system mailbox (/var/spool/mail/UserID) instead of in your \$HOME/mbox file. Does not override the delete subcommand. Identical to the preserve subcommand.

Conditional execution of mail subcommands

Item	Description
if Condition elseif endif	Construction for conditional execution of the mail subcommands. Subcommands following if are executed if <i>Condition</i> is true. Subcommands following elseif are executed if <i>Condition</i> is not true. The elseif is not required. The endif ends the construction and is required. The <i>Condition</i> can be receive (receiving mail) or send (sending mail).

Item	Description
ignore [<i>FieldList</i>]	Adds the header fields in <i>FieldList</i> to the list of fields to be ignored. Ignored fields are not displayed when you look at a message with either the type or print subcommand. Use this subcommand to suppress machine-generated header fields. Use either the Type or Print subcommand to print a message in its entirety, including ignored fields. The ignore subcommand with no arguments lists all header fields that are not included when you use a type or print subcommand to display a message. Identical to the discard subcommand.
list	(l) Displays a list of all mailbox subcommands with no explanation of what they do.
mail AddressList	(m) Starts the mail editor. Enables you to create and send a message to people specified in <i>AddressList</i> . The newly created message is independent from any receive messages.
mbox [<i>MessageList</i>]	Indicates that the messages in <i>MessageList</i> are to be sent to your personal mailbox (\$HOME/mbox) when you quit the Mail program. This operation is the default action for messages that you have read if you are looking at your system mailbox (/var/spool/mail/UserID) and the hold option is not set.
more [<i>MessageList</i>]	(mo) Displays the messages in <i>MessageList</i> using the defined pager program to control display to the screen. Identical to the page subcommand.
More [<i>MessageList</i>]	(Mo) Similar to the more subcommand, but also displays ignored header fields.
new [<i>MessageList</i>]	Marks each message in <i>MessageList</i> as <i>not</i> having been read. Identical to the New , unread , and Unread subcommands.
New [<i>MessageList</i>]	Marks each message in <i>MessageList</i> as <i>not</i> having been read. Identical to the new , unread , and Unread subcommands.
next [<i>Message</i>]	(n) Makes the next message in the mailbox the current message and displays that message. With an argument list, it displays the next matching message.
page [<i>MessageList</i>]	(pa) Displays the messages in <i>MessageList</i> using the defined pager program to control display to the screen. Identical to the more subcommand.
Page [<i>MessageList</i>]	(Pa) Similar to the page subcommand but also displays ignored header fields.

pi command

Item	Description
pipe [[<i>msglist command</i>] [[<i>msglist</i>] <i>command</i>]]	(pi) Pipe the messages through the given command by invoking the command interpreter specified by SHELL with two arguments: -c and <i>command</i> . The command must be given as a single argument. This can be accomplished by quoting. If no arguments are given, the current message will be piped through the command specified by the value of the cmd variable. If the page variable is set, a form-feed character will be inserted after each message.
preserve	(pre) Identical to the hold subcommand.
print [<i>MessageList</i>]	(p) Displays the text of a specific message. Identical to the type subcommand.
Print [<i>MessageList</i>]	(P) Displays the text of a specific message along with the ignored header fields. Identical to the Type subcommand.

Item	Description
quit	(q) Leaves the mailbox and returns to the operating system. All messages read, but not deleted or saved are stored in your personal mailbox (\$HOME/mbx). All messages you have marked to be deleted are removed from the mailbox and cannot be recovered. All messages marked with the hold or preserve option and messages you have not viewed are saved in the system mailbox (/var/spool/mail/UserID). If the quit subcommand is given while editing a mailbox file with the -f flag, the edit file is saved with changes. If the edit file cannot be saved, the Mail program does not exit. Use the exit subcommand to exit without saving the changes.
reply [<i>Message</i>]	(r) Allows you to reply to the sender of a message and to all others who receive copies of the message. Identical to the respond subcommand.
Reply [<i>Message</i>]	(R) Allows you to reply to only the sender of a message. Identical to the Respond subcommand.
respond [<i>Message</i>]	Allows you to reply to the sender of a message and to all others who receive copies of a message. Identical to the reply subcommand.
Respond [<i>Message</i>]	Allows you to reply to only the sender of a message. Identical to the Reply subcommand.
retain [<i>FieldList</i>]	Adds the header fields in <i>FieldList</i> to the list of fields to be retained. Retained fields are displayed when you look at a message with the type subcommand or print subcommand. Use this subcommand to define which header fields you want displayed. Use the Type or Print subcommand to print a message in its entirety, including fields that are not retained. If the retain subcommand is executed with no arguments, it lists the current set of retained fields. Note: The retain subcommand overrides the discard subcommand.
save [<i>File</i>]	(s) Saves the current message including header information to a file or folder. If the file already exists, the message is appended to the file. If <i>File</i> is omitted, the message will be saved to the user's mbx .
save [<i>MessageList</i>] <i>File</i>	(s) Saves a <i>MessageList</i> including heading information to a file or folder. If the file already exists, the <i>MessageList</i> is appended to the file. Displays the file name and the size of the file when the operation is complete. If you save a message to a file, that message is not returned to the system mailbox (/var/spool/mail/UserID) nor saved in your personal mailbox (\$HOME/mbx) when you quit the Mail program.
Save [<i>MessageList</i>]	(S) Saves the specified messages in a file whose name is derived from the author of the first message. The name of the file is taken to be the author's name with all network addressing stripped off.

Item	Description
set [<i>OptionList</i> <i>Option=Value...</i>]	<p>(se) With no arguments, displays the options that are currently enabled. Otherwise, sets an option as specified. The argument following the set command can be either:</p> <ul style="list-style-type: none"> • An <i>OptionList</i> giving the name of a <u>binary option</u> (an option that is either set or unset) • An <i>Option=Value</i> entry used to <u>assign a value to an option</u>. <p>The options are listed in the .mailrc file format.</p> <p>Note: The form unset name is equivalent to noname.</p>
shell	(sh) Starts an interactive version of the shell.
size [<i>MessageList</i>]	Displays the sizes in lines/characters of the messages in <i>MessageList</i> .
source <i>File</i>	(so) Reads and executes the mail subcommands from <i>File</i> .
top [<i>MessageList</i>]	Displays the top few lines of the messages specified by <i>MessageList</i> . The number of lines displayed is determined by the valued option toplines and defaults to five.
touch [<i>MessageList</i>]	Within your system mailbox (/var/spool/mail/UserID), this subcommand marks the messages in <i>MessageList</i> to be moved to your personal mailbox (\$HOME/mbox) when you quit the Mail program. The messages are moved even though you have not read them. The messages are displayed in your personal mailbox as unread messages. The last message in <i>MessageList</i> becomes the current message.
type [<i>MessageList</i>]	(t) Displays the text of a specific message. Identical to the print subcommand.
Type [<i>MessageList</i>]	(T) Displays the text of a specific message along with the ignored header fields. Identical to the Print subcommand.
unalias	Deletes the specified alias names.
undelete [<i>MessageList</i>]	(u) Removes the messages in <i>MessageList</i> from the list of messages to be deleted when you quit the Mail program. Entering the u subcommand without a message list recalls the last deleted message.
unread [<i>MessageList</i>]	(U) Marks each message in <i>MessageList</i> as <i>not</i> having been read. Identical to the new , New , and Unread subcommands.
Unread [<i>MessageList</i>]	Marks each message in <i>MessageList</i> as <i>not</i> having been read. Identical to the new , New , and unread subcommands.
unset <i>OptionList</i>	Disables the values of the options specified in <i>OptionList</i> . This action is the inverse of the set subcommand.
	Note: The form unset name is equivalent to noname .
version	(ve) Displays the version banner for the Mail program.
visual [<i>MessageList</i>]	(v) Starts the visual editor using the <i>MessageList</i> as the input field. (This editor can be defined with the set VISUAL= statement.) Any changes made during the editor session are saved back to the messages in the <i>MessageList</i> .

pi command (*continued*)

Item	Description
write [<i>MessageList</i>] <i>File</i>	(w) Saves a message without heading information to a file or folder. Displays the file name and the size of the file when the operation is complete. Does not include message headers in the file.
xit	(x) Identical to the exit subcommand.
z [+ -]	Changes the current message group (group of 20 messages) and displays the headings of the messages in that group. If a + or no argument is given, then headings in the next group are shown. If a - argument is given, the headings in the previous group are shown.

Mail Editor Subcommands for the mail, Mail Command

By default, the Mail program treats lines beginning with the ~ (tilde) character as subcommands. The following list describes the subcommands used while in the mail editor. The editor recognizes subcommands only if you enter them at the beginning of a new line.

Mail editor subcommands

Item	Description
~?	Displays a summary of the mail subcommands.
~!Command	The command interpreter specified by SHELL will be invoked with two arguments: -c and <i>command</i> . The standard output of command will be inserted into the message.
~a	Inserts the value of the sign variable into the text of the message, followed by a newline character. Identical to ~i sign .
~A	Inserts the value of the Sign variable into the text of the message, followed by a newline character. Identical to ~i Sign .
~b AddressList	Adds names in <i>AddressList</i> to the list of addresses to receive blind copies of the message. The ~b subcommand can only be used to add to, not change or delete, the contents of the <code>Bcc : List</code> .
~c AddressList	Adds names in <i>AddressList</i> to the list of people to receive copies of the message. The ~c subcommand can only be used to add to, not change or delete, the contents of the <code>Cc : List</code> .
~d	Appends the contents of the dead.letter file to the end of the message.
~e	Starts the alternate editor using the message text as the input file. (This editor can be defined with the set EDITOR = statement in the Bourne shell.) When you exit that editor, you return to the mail editor, where you may add text, or send the message by exiting the Mail program.
~f [MessageList]	Includes a <i>MessageList</i> in the current message to forward the message to another user. This subcommand reads each message in the <i>MessageList</i> and appends it to the current end of the message, but does not indent the appended message. This subcommand is also used to append messages for reference whose margins are too wide to embed with the ~m subcommand. This subcommand works only if you entered the mail editor from the mailbox prompt using either the mail , reply , or Reply subcommand.
~F [MessageList]	Equivalent of the ~f , except that all headers will be included in the message, regardless of previous discard , ignore , and retain commands.

Item	Description
-h	Enables you to add or change information in all of the heading fields. The system displays each of the four heading fields, one at a time. You can view the contents of each field and delete or add information to that field. Press the Enter key to save any changes to the field and to display the next field and its contents.
-i <i>string</i>	Inserts the value of the named variable, followed by a newline character, into the text of the message. If the string is unset or null, the message will not be changed.
-m [<i>MessageList</i>]	Includes a <i>MessageList</i> in the current message for reference purposes. This subcommand reads each message in the <i>MessageList</i> and appends it to the current end of the message. The included message is indented one tab character from the usual left margin of the message. This subcommand works only if you entered the mail editor from the mailbox prompt using either the mail , reply , or Reply subcommand.
-M [<i>MessageList</i>]	Equivalent of the -m , except that all headers will be included in the message, regardless of previous discard , ignore , and retain commands.
-p	Displays the entire message, including header information.
-q	Quits the editor without sending the message. Saves the message in the dead.letter file in your home directory, unless the nosave option is set. The previous contents of the dead.letter file are replaced with the partially completed message. Note: You can also quit the editor by using the Interrupt (Ctrl+C) key sequence twice.
-r <i>File</i>	Reads the contents of a file into the current message.
-s <i>String</i>	Changes the subject field to the phrase specified in <i>String</i> . You cannot append to the subject field with this subcommand.
-t <i>AddressList</i>	Adds the addresses in <i>AddressList</i> to the To : field of the message. The -t subcommand can only be used to add to, not change or delete, the contents of the To : <i>List</i> .
-v	Starts the visual editor using the message text as the input file. This editor can be defined with the set VISUAL= statement in the Bourne shell.) When you exit that editor, you return to the mail editor where you may add text to the message, or send the message by exiting the Mail program.
-w <i>File</i>	Writes the message to the named file.
-x	Exits as with -q , except the message will not be saved in the dead.letter file.
-: <i>Subcommand</i>	Executes the subcommand specified by <i>Subcommand</i> and returns to the mail editor.
- <i>Command</i>	Pipes the message through the command <i>Command</i> as a filter. If <i>Command</i> gives no output or terminates unusually, it retains the original text of the message. Otherwise, the output of <i>Command</i> replaces the current message. The fmt command is often used as <i>Command</i> to format the message.
-< <i>file</i>	Reads the contents of a file into the current message.

Item	Description
~< Command	Allows you to run a shell command. The shell runs with the -c flag and the Command specified. The standard output of Command is inserted into the message.
~~	Allows you to use the ~ (tilde) character in a message without it being interpreted as a command prefix. The ~~ key sequence results in only one ~ character being sent in the message.

Files

Files

Item	Description
\$HOME/.mailrc	Contains the mail subcommands to customize the Mail program for a specific user.
\$HOME/mbox	Contains your personal mailbox.
/usr/share/lib/Mail.rc	Contains the file with mail subcommands to change the Mail program for all users on the system.
/var/spool/mail/*	Contains system mailboxes for all users.
/usr/bin/mail	Contains the mail command.
/usr/bin/Mail	Contains the Mail command.
/usr/bin/mailx	Contains the mailx command.

mailq Command

Purpose

Prints the contents of the mail queue.

Syntax

/usr/sbin/mailq [**-v**]

Description

The **mailq** and **MAILQ** commands print a list of messages that are in the mail queue. The first line printed for each message shows:

- The internal identifier used on this host for the message with a possible status character
- The size of the message in bytes
- The date and time the message is accepted into the queue
- the envelope sender of the message

Note: Starting from AIX 7 with 7200-04, the non-root users must run the **mailq -Ac** command instead of the **mailq** command to view the messages in the mail queue.

The second line shows the error message that caused the message to be retained in the in the queue, it is not displayed if the message is being displayed for the first time. The status characters are either:

Indicates the job is being processed

X

Indicates that the load is too high to process the job

-

Indicates that the job is too young to process

The following lines show message recipients, one per line.

The **mailq** command is the same as the **sendmail -bp** command.

Specify the **-v** flag to display message priority.

Flags

Item	Description
-v	Prints verbose information. This adds the priority of the message and a single character indicator (+ or blank) indicating whether a warning message has been sent on the first line of the message. Additionally, extra lines may be intermixed with the recipients indicating the <i>controlling user</i> information; this shows who owns any program that are executed on behalf of this message and the name of the alias this command expanded from, if any.
-Ac	Prints the list of messages that are queued in the <code>/var/spool/clientmqueue</code> directory.

Exit Status

The command returns the following exit values:

Item	Description
0	Exits successfully.
>0	An error occurred.

Examples

The **mailq** command prints two types of lists:

- The **mailq** command lists the mail queue as shown in the following example:

```
Mail Queue (1 request)
---QID--- --Size-- -----Q-Time----- -----Sender/Recipient-----
AA02508      3   Thu Dec 17 10:01             root
              (User unknown)
                               bad_user
```

- The **mailq -v** command lists the mail queue as follows:

```
Mail Queue (1 request)
---QID--- --Size-- -Priority- ---Q-Time--- --Sender/Recipient--
AA02508      3   1005   Dec 17 10:01       root
              (User unknown)
                               bad_user
```

- The **mailq -Ac** command lists the `clientmqueue` mail queue as follows:

```
/var/spool/clientmqueue (1 request)
-----Q-ID----- --Size-- -----Q-Time----- -----Sender/Recipient-----
00FGNTRH12845482  5       Wed Jan 15 10:23             root
              (Deferred: Connection refused by [127.0.0.1])
              unknown-user
Total requests: 1
```

The fields have the following meanings:

Item	Description
QID	Contains the message queue ID of the message.
Size	Contains the number of bytes in the body of the message (heading information not included).
Priority	Contains the priority of the message, based primarily on the size of the message.
Q-Time	Contains the time the message entered the queue.
Sender/Recipient	Contains the user ID of the sender and the recipient of the message. A message on the line between the sender and the recipient indicates the status of the message.

Files

Item	Description
<code>/usr/sbin/mailq</code>	Contains the mailq command.
<code>/var/spool/mqueue</code> directory	Contains the log file and temporary files associated with the messages in the mail queue.
<code>/var/spool/clientmqueue</code>	Contains the messages that are in mail queue for non-root users.

mailstats Command

Purpose

Displays statistics about mail traffic.

Syntax

```
mailstats [ -C cfFile ] [ -c ] [ -P ] [ -f StatFile ] [ -o ] [ -p ]
```

Description

The **mailstats** command displays the current mail statistics. The time at which the statistics started displays reads the information in the format specified by **ctime**. The statistics for each mailer are displayed on a single line, with the following fields:

Item	Description
M	Contains the mailer number.
msgsfr	Contains the number of messages received by the local machine from the indicated mailer.
bytes_from	Contains the number of Kbytes in the messages received by the local machine from the indicated mailer.
msgsto	Contains the number of messages sent from the local machine using the indicated mailer.
bytes_to	Contains the number of bytes in the messages sent from the local machine using the indicated mailer.
msgsjrej	Contains the number of messages rejected.
msgsdisc	Contains the number of messages discarded.
Mailer	Contains the name of mailer.

After the statistics are displayed, a line totaling the value of all of the mailers displays, preceeded with a **T**. This information is separated from the statistics by a line containing only = (equal characters). Another line preceeded with a **C** lists the number of connections.

Flags

Item	Description
-C <i>cfFile</i>	Specifies use of the <i>cfFile</i> instead of the default sendmailcf file.
-c	Specifies use of the submit.cf file instead of the sendmail.cf file.
-f <i>StatFile</i>	Specifies use of the <i>StatFile</i> instead of the statistics file specified in the sendmail.cf file.
-o	Specifies that the name of the mailer does not display in the output.
-p	Outputs information in program readable mode and clears the statistics.
-P	Outputs information in program readable mode without clearing the statistics.

Exit Status

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.

Files

Item	Description
/etc/mail/statistics	Default sendmail statistics file.
/etc/mail/sendmail.cf	Default sendmail configuration file.

make Command

Purpose

Maintains, updates, and regenerates groups of programs.

Syntax

```
make [ -DVariable ] [ -d Option ] [ -e ] [ -i ] [ -j [Jobs] ] [ -k ] [ -n ] [ -p ] [ -q ] [ -r ] [ -S ] [ -s ] [ -t ] [ -f MakeFile ... ] [ Target ... ]
```

Description

The **make** command assists you in maintaining a set of programs. Input to the **make** command is a list of file dependency specifications.

There are four types of lines in a makefile: file dependency specifications, shell commands, variable assignments, and comments. In general, lines can be continued from one line to the next by ending them with a \ (backslash). The trailing newline character and initial white space on the following line are compressed into a single space.

File Dependency Specifications

Dependency lines consist of one or more targets, an operator, and zero or more prerequisites (sources). This creates a relationship where the targets depend on the prerequisites and are usually created from them. The exact relationship between the target and the prerequisite is determined by the operator that separates them. The operators are as follows:

Item Description

- : A target is considered out-of-date if its modification time is less than that of any of its prerequisites. Prerequisites for a target accumulate over dependency lines when this operator is used. The target is removed if the **make** command is interrupted, unless the target has the **.PRECIOUS** attribute.
- :: If no prerequisites are specified, the target is always recreated. Otherwise, a target is considered out-of-date if any of its prerequisites were modified more recently than the target. Prerequisites for a target do not accumulate over dependency lines when this operator is used. The target is not removed if the **make** command is interrupted.

File dependency specifications have two types of rules, inference and target. Inference rules specify how a target is to be made up-to-date. These rules have one target with no / (slash) and a minimum of one . (period). Target rules specify how to build the target. These rules can have more than one target.

Makefile Execution

The **make** command executes the commands in the makefile line by line. As **make** executes each command, it writes the command to standard output (unless otherwise directed, for example, using the **-s** flag). A makefile must have a Tab in front of the commands on each line.

When a command is executed through the **make** command, it uses **make**'s execution environment. This includes any macros from the command line to the **make** command and any environment variables specified in the **MAKEFLAGS** variable. The **make** command's environment variables overwrite any variables of the same name in the existing environment.

Note: When the **make** command encounters a line beginning with the word **include** followed by another word that is the name of a makefile (for example, **include depend**), the **make** command attempts to open that file and process its contents as if the contents were displayed where the include line occurs. This behavior occurs only if the first noncomment line of the first makefile read by the **make** command is not the **.POSIX** target; otherwise, a syntax error occurs.

Comments: Comments begin with a # character, anywhere but in a shell command line, and continue to the end of the line.

Environment: The **make** command uses the **MAKEFLAGS** environment variable, if it exists.

Target Rules

Target rules have the following format:

```
target[target...] : [prerequisite...] [;command]
<Tab>command
```

Multiple targets and prerequisites are separated by spaces. Any text that follows the ; (semicolon) and all of the subsequent lines that begin with a Tab character are considered commands to be used to update the target. A new target entry is started when a new line does not begin with a Tab or # character.

Note: The list of prerequisites can be empty.

Special Targets

Special targets cannot be included with other targets; that is, they must be the only target specified. These targets control the operation of the **make** command. These targets are:

Item	Description
.DEFAULT	This is used as the rule for any target (that was used only as a prerequisite) that the make command cannot figure out any other way to create. Only the shell script is used. The < (left angle bracket) variable of a target that inherits .DEFAULT 's commands is set to the target's own name.
.IGNORE	Prerequisites of this target are targets themselves; this causes errors from commands associated with them to be ignored. If no prerequisites are specified, this is the equivalent of specifying the -i flag.
.POSIX	Causes the make command to use a different default rules file. The file, /usr/ccs/lib/posix.mk , provides the default rules as specified in the POSIX standard.
.PRECIOUS	Prerequisites of this target are targets themselves. .PRECIOUS prevents the target from being removed. If no prerequisites are specified, the .PRECIOUS attribute is applied to every target in the file. Usually, when make is interrupted (for example, with SIGHUP , SIGTERM , SIGINT , or SIGQUIT), it removes any partially made targets. If make was invoked with the -n , -p , or -q flags, the target is considered to have the .PRECIOUS attribute.
.SCCS_GET	This special target must be specified without prerequisites. If this special target is included in a makefile, the commands associated with this special target are used to get all SCCS files that are not found in the current directory. The default commands that are used to retrieve the source files from SCCS are replaced by the commands associated with this special target. When source files are named in a dependency list, make treats them just like any other target. When a target has no dependencies, but is present in the directory, make assumes that the file is up-to-date. If, however, a SCCS file named SCCS/s.source_file is found for a target source_file , make additionally checks to assure that the target is up-to-date. If the target is missing, or if the SCCS file is newer, make automatically issues the commands specified for the .SCCS_GET special target to retrieve the most recent version. However, if the target is writable by anyone, make does not retrieve a new version.
.SILENT	Prerequisites of the target are targets themselves. This causes commands associated with the target to not be written to standard output before they are executed. If no prerequisites are specified, the .SILENT attribute is applied to every command in the file.
.SUFFIXES	Use this name to add more suffixes to the list of file suffixes that make recognizes. Prerequisites of the target are appended to the list of known suffixes. If no suffixes are specified, any previously specified suffixes are deleted. These suffixes are used by the inference rules. To change the order of suffixes, you need to specify an empty .SUFFIXES entry and then a new list of .SUFFIXES entries. A makefile must not associate commands with .SUFFIXES .

Inference Rules

The **make** command has a default set of inference rules, which you can supplement or overwrite with additional inference rules definitions in the makefile. The default rules are stored in the external file, **/usr/ccs/lib/aix.mk**. You can substitute your own rules file by setting the **MAKERULES** variable to your own file name from the command line. The following line shows how to change the rules file from the command line:

```
make MAKERULES=/pathname/filename
```

Inference rules consist of target suffixes and commands. From the suffixes, the **make** command determines the prerequisites, and from both the suffixes and their prerequisites, the **make** command determines how to make a target up-to-date. Inference rules have the following format:


```
rule:
<Tab>command
...
```

where `rule` has one of the following forms:

Item	Description
<code>.s1</code>	A single-suffix inference rule that describes how to build a target that is appended with one of the single suffixes.
<code>.s1.s2</code>	A double-suffix inference rule that describes how to build a target that is appended with <code>.s2</code> with a prerequisite that is appended with <code>.s1</code> .

The `.s1` and `.s2` suffixes are defined as prerequisites of the special target, **.SUFFIXES**. The suffixes `.s1` and `.s2` must be known suffixes at the time the inference rule is displayed in the makefile. The inference rules use the suffixes in the order in which they are specified in **.SUFFIXES**. A new inference rule is started when a new line does not begin with a `<Tab>` or `#` character.

If `rule` is empty, for example:

```
rule: ;
```

execution has no effect, and the **make** command recognizes that the suffix exists, but takes no actions when targets are out-of-date.

A `~` (tilde) in the preceding rules refers to an SCCS file. Therefore, the rule, **.c~.o**, would transform an SCCS C language prerequisite file into an object file (**.o**). Because the **s.** of the SCCS file is a prefix, it is incompatible with the **make** command's suffix view. The `~` (tilde) is a way of changing any file reference into an SCCS file reference.

Libraries

A target or prerequisite can also be a member of an archive library and is treated as such if there are parentheses in the name. For example, *library(name)* indicates that *name* is a member of the archive library *library*. To update a member of a library from a particular file, you can use the format `.s1.a`, where a file with the `.s1` suffix is used to update a member of the archive library. The **.a** refers to an archive library.

Using Macros

In makefiles, macro definitions are defined in the format:

```
variable=value
```

Macros can be displayed throughout the makefile, as follows:

- If a macro is displayed in a target line, it is evaluated when the target line is read.
- If a macro is displayed in a command line, it is evaluated when the command is executed.
- If a macro is displayed in a macro definition line, it is evaluated when the new macro is displayed in a rule or command.

If a macro has no definition, it defaults to **NULL**. A new macro definition overwrites an existing macro of the same name. Macros assignments can come from the following, in the listed order:

1. Default inference rules
2. Contents of the environment
3. Makefiles
4. Command lines.

Note: The **-e** flag causes environment variables to override those defined in the makefile.

The **SHELL** macro is special. It is set by the **make** command to the path name of the **shell** command interpreter (**/usr/bin/sh**). However, if it is redefined in the makefile or on the command line, this default setting is overridden.

Note: The **SHELL** macro does not affect, and is not affected by, the **SHELL** environment variable.

Shell Commands

Each target can have associated with it a series of shell commands, usually used to create the target. Each of the commands in this script must be preceded by a Tab. While any target can be displayed on a dependency line, only one of these dependencies can be followed by a creation script, unless the **::** operator is used.

If the first, or first two characters, of the command line are one or all of **@** (at sign), **-** (hyphen), and **+** (plus sign), the command is treated specially, as follows:

Item Description

- @** Causes the command not to be echoed before it is executed.
- Causes any nonzero exit status of the command line to be ignored.
- +** Causes a command line to be executed, even though the options **-n**, **-q**, or **-t** are specified.

A command that has no metacharacters is directly executed by the **make** command. For example, the **make** command consigns the first command in the following example to the shell because it contains the **>** (greater than sign) shell metacharacter. The second command in the following example does not contain any shell metacharacters, so the **make** command executes it directly:

```
target: dependency
    cat dependency > target
    chmod a+x target
```

Bypassing the shell saves time, but it can cause problems. For example, attempting to execute a C shell script from within a makefile by setting the **SHELL** macro to **/bin/csh** will not work unless the command line also contains at least one shell metacharacter.

```
SHELL=/bin/csh

target: dependency
    my_csh_script
```

This makefile fails because the **make** command attempts to run **my_csh_script** instead of consigning it to the C shell.

Variable Assignments

Variables in the **make** command are much like variables in the shell and consist of all uppercase letters. The **=** operator assigns values to variables. Any previous variable is then overridden. Any white space before the assigned value is removed.

Values can be appended to macro values as follows:

```
macro += word ...
macro += macro1
```

The **+=** operator when used in place of **=** appends the new value with a single space inserted between the previous contents of the variable and the appended value.

Variables are expanded by surrounding the variable name with either **{ }** (braces) or **()** (parentheses) and preceding it with a **\$** (dollar sign). If the variable name contains only a single letter, the surrounding braces or parentheses are not required. This shorter form is not recommended.

Variable substitution occurs at two distinct times, depending on where the variable is being used.

Variables in dependency lines are expanded as the line is read. Variables in shell commands are expanded when the **shell** command is executed.

The four classes of variables (in order of increasing precedence) are:

Item	Description
Environment	Variables defined as part of the make command's environment.
Global	Variables defined in the makefile or in included makefiles.
Command line	Variables defined as part of the command line.
Local	Variables defined specific to a certain target. The local variables are as follows: \$< Represents either the full name of a prerequisite that made a target out-of-date (inference rule), or the full name of a target (.DEFAULT rule). \$* Represents the file name section of a prerequisite that made a target out-of-date (in an inference rule) without a suffix. \$@ Represents the full target name of the current target or the archive file name part of the library archive target. \$% Represents a library member in a target rule if the target is a member of the archive library. You can also use these local variables appended with D or F : D Indicates that the local variable applies to the directory part of the name. This is the path name prefix without a trailing / (slash). For current directories, D is a . (period). F Indicates that the local variable applies to the file name part of the name. In addition, the make command sets or knows about the following variables: \$ A single \$ (dollar sign); that is, \$\$ expands to a single dollar sign. LANG Determines the locale to use for the locale categories when both LC_ALL and the corresponding environment variable (beginning with LC_) do not specify a locale. LC_ALL Determines the locale to be used to override any values for locale categories specified by the setting of LANG or any other LC_ environment variable. LC_CTYPE Determines the locale for the interpretation of sequences of bytes of text data as characters; for example, single- versus multibyte characters in arguments. LC_MESSAGES Determines the language in which messages should be written. MAKEFLAGS The environment variable, MAKEFLAGS , can contain anything that can be specified on make 's command line. Anything specified on make 's command line is appended to the MAKEFLAGS variable, which is then entered into the environment for all programs that make executes. Note that the operation of the -f and -p flags in the MAKEFLAGS variable is undefined. Command line flags take precedence over the -f and -p flags in this variable.

Item	Description
VPATH	Allows you to specify a list of directories to search for prerequisites. The list of directories works like the PATH variable in the SHELL . The VPATH variable can specify multiple directories separated by colons. For example:

```
VPATH=src:/usr/local/src
```

This tells the **make** command to search for the following directories in the order given:

- The current directory (this happens even without **VPATH**)
- `src` (a subdirectory in the current directory)
- `/usr/local/src`.

Flags

Item	Description
-DVariable	Sets the value of <i>Variable</i> to 1.
-dOption	Displays detailed information about the files and times that make examines (debug mode). The -d flag without any options or with the <i>A</i> option displays all the debug information available. Individually selectable debug options follow: <ul style="list-style-type: none"> A Displays all possible debug information. a Displays debug information about archive searching and caching. d Displays debug information about directory searching. g1 Displays debug information about input graph before making anything. g2 Displays debug information about input graph after making everything, or before exiting on an error. m Displays debug information about making targets, including modification dates. s Displays debug information about suffix searching. v Displays debug information about variable assignments.
-e	Specifies that environmental variables override macro assignments within makefiles.
-f MakeFile	Specifies a makefile to read instead of the default makefile. If <i>MakeFile</i> is - (hyphen), standard input is read. Multiple makefiles can be specified and are read in the order specified.
-i	Ignores nonzero exit of shell commands in the makefile. Equivalent to specifying - (hyphen) before each command line in the makefile.
-j[Jobs]	Specifies the number of parallel jobs that make should use to build the independent targets. The <i>Jobs</i> parameter can take any positive integral values. If <i>Jobs</i> is not specified, the make command does not limit the number of parallel jobs for building the main target.
-k	Continues processing after errors are encountered, but only on those targets that do not depend on the target whose creation caused the error.

Item	Description
-n	Displays commands, but does not run them. However, lines beginning with a + (plus sign) are executed.
-p	Displays the complete set of macro definitions and target descriptions before performing any commands.
-q	Returns a zero status code if the target file is up-to-date; returns a one status code if the target file is not up-to-date. Targets will not be updated when this option is specified. However, a command line with the + (plus sign) prefix will be executed.
-r	Does not use the default rules.
-S	Terminates the make command if an error occurs. This is the default and the opposite of -k flag.
-s	Does not display commands on the screen as they are performed.
-t	Creates a target or updates its modification time to make it seem up-to-date. Executes command lines beginning with a + (plus sign).
<i>Target</i>	Specifies a target name of the form <i>Target</i> or sets the value of variables.

Exit Status

When the **-q** flag is specified, this command returns the following exit values:

Item	Description
0	Successful completion.
1	The target was not up-to-date.
>1	An error occurred.

Otherwise, this command returns the following exit values:

Item	Description
0	Successful completion.
>1	An error occurred.

Examples

1. To make the first target found in the makefile, type:

```
make
```

2. To display, but not run, the commands that the **make** command would use to make a file:

```
make -n search.o
```

Doing this will verify that a new description file is correct before using it.

3. To create a makefile that says that **pgm** depends on two files, **a.o** and **b.o**, and that they, in turn, depend on their corresponding prerequisite files (**a.c** and **b.c**) and a common file, **incl.h**, type:

```
pgm: a.o b.o
      c89 a.o b.o -o pgm
a.o: incl.h a.c
      c89 -c a.c
b.o: incl.h b.c
      c89 -c b.c
```

4. To make optimized **.o** files from **.c** files, type:

```
.c.o:      c89 -c -o $*.c
or:
.c.o:      c89 -c -o $<
```

5. To view the contents of the built-in rules, type:

```
make -p -f /dev/null 2>/dev/null
```

6. To use the **make** command in parallel mode with a maximum of 10 parallel jobs to be used for building the target specified in the makefile, type:

```
make -j10
```

Files

Item	Description
makefile	Contains a list of dependencies.
Makefile	Contains a list of dependencies.
s.makefile	Contains a list of dependencies. It is an SCCS file.
s.Makefile	Contains a list of dependencies. It is an SCCS file.
/usr/ccs/lib/posix.mk	Contains default POSIX rules for the make command.
/usr/ccs/lib/aix.mk	Contains default rules for the make command.

makedbm Command

Purpose

Makes a Network Information Services (NIS) database map.

Syntax

To Create an Map

```
/usr/sbin/makedbm [ -b ] [ -i NISInputFile ] [ -o NISOutputFile ] [ -d NISDomainName ]  
[ -m NISMasterName ] InputFile OutputFile
```

To Create a Non-dbm Formatted Map

```
/usr/sbin/makedbm [ -u dbmFileName ]
```

Description

The **makedbm** command makes an NIS map. It does this by converting the file named in the *InputFile* parameter into two output files: *OutputFile.pag* and *OutputFile.dir*. Each line in each input file is converted into a single Data Base Manager (DBM) record.

The **makedbm** command is most often invoked from the **/var/yp/Makefile** file to generate NIS maps. All characters leading up to the first space or tab in each line of the **/var/yp/Makefile** file form the key. The rest of the line contains value data. If a line ends with a \ (backslash), data for that record is continued on the next line. NIS clients must interpret the # (pound sign) symbol since the **makedbm** command does not treat it as a comment character. If the *InputFile* parameter is a - (minus sign), the **makedbm** command reads standard input instead.

This command generates a special entry in the output map by using the **YP_LAST_MODIFIED** key, which is the date that the file specified by the *InputFile* parameter was created (or the current time, if the *InputFile* parameter is a - (minus sign)).

Flags

It	Description
----	-------------

- | | |
|-----------|--|
| m | |
| -b | Propagates a map to all servers using the named name server. |
| -i | Creates a special entry with the YP_INPUT_FILE key. |
| -o | Creates a special entry with the YP_OUTPUT_FILE key. |
| -d | Creates a special entry with the YP_DOMAIN_NAME key. |
| -m | Creates a special entry with the YP_MASTER_NAME key. |
| -u | Undoes a DBM file. That is, prints out a DBM file one entry per line, with a single space separating keys from values. |

Files

Item	Description
<code>/var/yp/Makefile</code>	Contains rules for making NIS maps.

makedepend Command

Purpose

Create dependencies in makefiles.

Syntax

```
makedepend [ -DName=Def ] [ -DName ] [ -IIncludeDir ] [ -YIncludeDir ] [ -a ] [ -fMakeFile ] [ -oObjSuffix ] [ -pObjPrefix ] [ -sString ] [ -wWidth ] [ -v ] [ -m ] [ —Options— ] SourceFile ...
```

Description

The **makedepend** command reads each *SourceFile* in sequence and parses it like a C-preprocessor. It processes all `#include`, `#define`, `#undef`, `#ifdef`, `#ifndef`, `#endif`, `#if`, and `#else` directives to determine which `#include` directives need to be used in a compilation. Any `#include` directives can reference files having other `#include` directives, and parsing occurs in these files as well.

Every file that a *SourceFile* includes, directly or indirectly, is what **makedepend** calls a "dependency." These dependencies are then written to a makefile in such a way that the **make** command can determine which object files must be recompiled when a dependency has changed.

By default, **makedepend** places its output in the file named **makefile** if it exists, otherwise **Makefile**. An alternate makefile may be specified with the **-f** flag. **makedepend** first searches the available makefile for the line:

```
# DO NOT DELETE THIS LINE - make depend depends on it.
```

or one provided with the **-s** flag, as a delimiter for the dependency output. If it finds the line, it deletes everything following the line to the end of the makefile and puts the output after the line. If **makedepend** does not find the line, it appends the delimited string to the end of the makefile and places the output immediately after the string.

For each *SourceFile* appearing on the command line, **makedepend** puts lines in the makefile in the following form.

```
SourceFile.o: dfile ...
```

Where *SourceFile.o* is the name from the command line with its suffix replaced with *.o*, and *dfile* is a dependency discovered in an `#include` directive while parsing the *SourceFile* or one of the files it included.

The algorithm used in this command assumes that all files compiled by a single makefile will be compiled with roughly the same **-I** and **-D** flags, and that most files in a single directory will include largely the same files.

Given these assumptions, **makedepend** expects to be called once for each makefile, with all source files that are maintained by the make file appearing on the command line. It parses each source and include file only once, maintaining an internal symbol table for each. As a result, the first file on the command line takes an amount of time proportional to the amount of time that a normal C preprocessor takes. On subsequent files, if it encounters an include file that it has already parsed, it does not parse again.

For example, imagine you are compiling two files, **file1.c** and **file2.c**, each includes the header file **header.h**. The **header.h** file includes the files **def1.h** and **def2.h**. When you run the command:

```
makedepend file1.c file2.c
```

then **makedepend** will first parse **file1.c** and consequently, **header.h** and then **def1.h** and **def2.h**. It then decides that the dependencies for this first file are:

```
file1.o: header.h def1.h def2.h
```

But when the program parses the second file, **file2.c** and discovers that it, too, includes **header.h**, it does not parse the file, but simply adds **header.h**, **def1.h** and **def2.h** to the list of dependencies for **file2.o**.

Note: If you do not have the source for `cpp` (the Berkeley C preprocessor), then **makedepend** will compile in such a way that all `#if` directives will evaluate to `False`, regardless of their actual value. This may cause the wrong `#include` directives to be evaluated. In these cases, it is recommended that you write a new parser for `#if` expressions. The need for a new parser should be clear from the following example:

Imagine you are parsing two files **file1.c** and **file2.c**, each includes the file **def.h**. The list of files that **def.h** includes might be very different when **def.h** is included by **file1.c** than when it is included by **file2.c**. But once **makedepend** arrives at a list of dependencies for a file, it is cast in concrete.

Flags

Note: The **makedepend** command ignores flags it does not understand. Flag usage is similar to that of the **cc** command.

Item	Description
-DName=Def or -DName	Places a definition for the <i>Name</i> variable in the makedepend command's symbol table. Without the <i>=Def</i> specifier, the symbol is defined as 1.
-IIncludeDir	Prepends the <i>IncludeDir</i> variable to the list of directories searched by the makedepend command when it encounters an <code>#include</code> directive. By default, the makedepend command searches only the /usr/include directory.
-YIncludeDir	Replaces all of the standard include directories with a single specified include directory, you can omit <i>IncludeDir</i> to prevent searching the standard include directories.

Item	Description
-a	Appends the dependencies to the end of the file instead of replacing them.
-fMakeFile	Enables you to specify an alternate makefile in which to place command output.
-oObjSuffix	Specifies an object suffix. For example, some systems may have object files whose suffix is something other than .o . This flag allows you to specify another suffix, such as ".b" with -o.b or ":obj" with -o.obj and so forth.
-pObjPrefix	Prepends the object file prefix to the name of the object file. This flag is used to designate a different directory for the object file. The default is the empty string.
-sString	Specifies the starting string delimiter. This flag permits you to specify a different string for makedepend to search for in the makefile.
-wWidth	Changes the maximum line width of output lines. The default maximum is 78 characters.
-v	Causes makedepend to display a list of files included by each input file on standard input.
-m	Causes makedepend to display a warning if any input file includes another file more than once. In previous version of makedepend this was the default behavior. This flag is provided for backward compatibility and to aid in debugging problems related to multiple inclusion.
—Options—	Ignores any unrecognized argument contained within a beginning and ending double hyphen. When makedepend encounters a double hyphen (—) in the argument list, any unrecognized argument following it is silently ignored; a second double hyphen terminates this treatment. The double hyphens enable makedepend to safely ignore esoteric compiler arguments that might normally be found in a CFLAGS make command macro (see the Examples section). All flags that makedepend recognizes and that appear between the pair of double hyphens are processed normally.

Examples

Normally, **makedepend** will be used in a makefile target so that typing **makedepend** updates the dependencies for the makefile.

```
SRCS=file1.c file2.c ...
CFLAGS=-O -DHACK -I../foobar -xyz
depend:
    makedepend -- $(CFLAGS) -- $(SRCS)
```

makedev Command

Purpose

Creates binary description files suitable for reading by the **troff** command and its postprocessors.

Syntax

makedev DESC | FontFile ...

Description

The **makedev** command creates binary files suitable for reading by the **troff** command and its postprocessors. When the **DESC** file is specified, the **makedev** command creates a **DESC.out** file and a set of font description files using the information contained in the **DESC** file. When a font file is specified, the **makedev** command creates the corresponding font description file.

Options

Item	Description
------	-------------

DES	Causes a DESC.out file to be created.
C	

Parameters

Item	Description
------	-------------

<i>FontFile</i>	Causes a <i>FontFile.out</i> file to be created.
-----------------	--

Examples

The following command:

```
makedev B
```

creates a **B.out** file, which contains the font tables for the Times-Bold fonts.

makekey Command

Purpose

Generates an encryption key.

Syntax

makekey

Description

The **makekey** command generates an encryption key for use with programs that perform encryption. Its input and output are usually pipes.

The **makekey** command reads 10 characters from standard input and writes 13 characters to standard output. The first 8 of the 10 input characters can be any sequence of ASCII characters, as specified by the *ASCIICharacters* parameter. The last two input characters, called the salt, are chosen from the sets 0 through 9, a through z, A through Z, . (period), and / (slash). The salt characters are repeated as the first two characters of the output. The remaining 11 output characters are chosen from the same set as the salt and constitute the output key that you use as the encryption key parameter of programs that perform encryption.

Examples

1. To generate an encryption key for input string 1234567890, type the following:

```
$ makekey  
1234567890
```

Then press the Ctrl-D key. The following encryption key is generated, and the \$ (shell prompt) follows immediately after the generated key on the same line:

```
90y744T/NXw1U$
```

2. To allow **makekey** to accept input strings through pipe, type the following command:

```
$ echo 1234567890 | makekey
```

3. To allow **makekey** to accept input strings from a file, type the following command:

```
$ cat infile
1234567890
$ makekey < infile
```

Alternatively, you can type the following command:

```
$ cat infile | makekey
```

makemap Command

Purpose

Creates database maps for **sendmail**.

Syntax

```
makemap [ -C -N -c CacheSize -d -D-e -f -l -o -r -s -t-u -v] Maptype Mapname
```

Description

The **makemap** command creates the database maps used by the keyed map lookups in the **sendmail** command. It reads input from the standard input and outputs them to the indicated *Mapname*.

Parameters

Item	Description
<i>Maptype</i>	Depending upon how it is compiled, this command handles up to three different database formats: dbm DBM format maps. This requires the ndbm library. btree B-tree format maps. This requires the new Berkeley DB library. hash Hash format maps. This requires the new Berkeley DB library. Note: In all cases, this command reads lines from the standard input, consisting of two words separated by white space. The first is the database key, the second is the value. The value may contain "%n" strings to indicate parameter substitution. Literal percents should be doubled ("%"). Blank lines and lines beginning with a "#" are ignored.
<i>Mapname</i>	Name of the map.

Note: Do not use **makemap** command to create the aliases data base, but **newaliases** which puts a special token into the data base that is required by **sendmail** command.

If the `TrustedUser` option is set in the **sendmail** configuration file and **makemap** is invoked as the root user, the generated files are owned by the specified trusted user.

Flags

Item	Description
-c <i>CacheSize</i>	Specifies to use the hash and B-tree cache size.
-C	Indicates to use the specified sendmail configuration file for looking up the TrustedUser option.
-d	Allows duplicate keys in the map. This is only allowed on B-tree format maps. If two identical keys are read, they are both inserted into the map.
-D	Specifies the character to use to indicate a comment (which is ignored) instead of the default of #.
-e	Allows empty values on the right side.
-f	Disables the function of folding all uppercase letters in the key to lowercase. This flag is intended to mesh with the -f flag in the K line in the sendmail.cf file. The value is never case folded.
-l	Lists supported map types.
-N	Includes the null byte that ends strings in the map. This flag must match the -N flag in the sendmail.cf K line.
-o	Specifies to append to an existing file. This flag allows you to augment an existing file.
-r	Allows replacement of existing keys. Normally, the makemap command complains if you repeat a key and does not perform the insert operation.
-s	Ignores safety checks on maps being created. This includes checking for hard or symbolic links in world writeable directories.
-t	Specifies the delimiter to use instead of white space. This flag is also used for dumping files.
-u	Dumps or unmaps the contents of the database to standard output.
-v	Specifies that the command verbosely print its status.

man Command

Purpose

Displays manual entries online.

Syntax

```
man [[ [-c] [-t] [section] ] | [-k | -f] ] [-F] [-m] [-Mpath] [-r] [-a] title ...
```

Description

The **man** command provides reference information on topics, such as commands, subroutines, and files. The **man** command provides one-line descriptions of commands specified by name. The **man** command also provides information on all commands whose descriptions contain a set of user-specified keywords.

The **man** command formats a specified set of manual pages. If you specify a section for the *section* parameter, the **man** command searches in that section of the manual pages for the title specified by the *title* parameter. The value of the *section* parameter can be either an Arabic number from 1 through 8 or a letter.

The section letters are:

Item	Description
C	Specifies commands (including system management commands).
F	Specifies file-type manual pages.
L	Specifies library functions.
n	Specifies new.
l	Specifies local.
o	Specifies old.
p	Specifies public.

Note: The **n**, **l**, **o**, and **p** section specifiers are not valid for reading the hypertext information bases, which contain the operating system documentation.

The section numbers are:

Ite	Description
m	
1	Indicates user commands and daemons.
2	Indicates system calls and kernel services.
3	Indicates subroutines.
4	Indicates special files, device drivers, and hardware.
5	Indicates configuration files.
6	Indicates games.
7	Indicates miscellaneous commands.
8	Indicates administrative commands and daemons.

Note: The operating system documentation in the hypertext information databases is grouped into three sections only: command manual pages (in section 1, equivalent to section C), subroutine manual pages (in section 3, equivalent to section L), and file manual pages (in section 4, equivalent to section F). When searching for hypertext information, specifying section 1, 6, 7, or 8 will default to the command manual pages, section 2 or 3 will default to the subroutine manual pages, and section 4 or 5 will default to the file manual pages.

If the *section* parameter is omitted, the **man** command searches all sections of the manual.

The search path the **man** command uses is a list of directories separated by a : (colon) in which manual subdirectories can be found.

The **man** command displays the manual pages as follows:

1. The **man** command searches the **nroff** directories (**man?**) under the **/usr/share/man** directory.
2. The **man** command searches the formatted version directories (**cat?**) under the **/usr/share/man** directory. If the formatted version is available, and if it has a more recent modify time than the **nroff** command source, the **man** command displays the formatted version. Otherwise, the manual page is formatted with the **nroff** command and displayed. If the user has permission, the formatted manual page is deposited in the proper place, so that later invocations of the **man** command do not format the page again.

Note: There is no **nroff** source for the supplied manual pages. However, you can put **nroff** source for manual pages into the **man** directories and the **man** command can locate and process the **nroff** source.

3. If the **man** command does not find a manual page in the **/usr/share/man/man** or **/usr/share/man/cat** directory, the **man** command searches the paths specified through **-M** option or MANPATH environment variable for nroff directories (man?) and formatted version directories (cat?).
4. If the **man** command does not find a manual page in the **/usr/share/man/man** or **/usr/share/man/cat** or the user-specified **man/cat** directory, the **man** command reads from the hypertext information bases. The hypertext information bases reside in the **/usr/share/man/info** directory structure and contain the operating system documentation. When reading from the hypertext databases, the **man** command does not put any manual pages in the **/usr/share/man/cat** directory structure. The **man** command converts the HTML file into a formatted text file to fit on the display, and displays the manual page using the command described by the PAGER environment variable.
5. If the **man** command does not find a manual page in the hypertext information bases residing in the **/usr/share/man/info** directory structure, it looks for user-specified hypertext information base (through **-M** or MANPATH). The user-defined hypertext information base, should follow the following directory structure:

```
BasePath[%{ L | l }]/DocLibraryname/Section/command_or_routine_or_filename.htm
```

Where:

- %L represents the ISO language notation specified using the LC_MESSAGES, %l represents the first 2 characters of the ISO language notation specified using the LC_MESSAGES. For example, for LC_MESSAGES=en_US the documents can be placed in Path/en_US or Path/en.
- DocLibraryname represents the name of the documentation library.
- Section represents the section name, which must be one of the following:
 - **cmds** — Represents Commands Section
 - **libs** — Represents Library Section
 - **files** — Represents Files Section

Note: If **-m** option is specified, then the search for manual pages will be done only in the order of paths specified through **-M** or the MANPATH environment variable.

When accessing the HTML databases, **man** looks for the operating system library before it proceeds to other LPP libraries. Within these libraries, it processes information in the following order:

Item	Description
cmds	Commands Reference
libs	Subroutines, System Calls
files	Files Reference

If the standard output is a tty, the **man** command pipes its output using the **more** command with the **-s** and **-v** flags. The **-s** flag eliminates multiple blank lines and stops after each page on the screen. The **-v** flag suppresses the display of nonprinting characters to the screen. To continue scrolling, press the space bar. To scroll an additional 11 lines when the output stops, press the Ctrl-D key sequence.

The **PAGER** environment variable can be set to whatever pager is desired. The default value is the **more** command. To change the default pager, enter:

```
PAGER=Somepager
export PAGER
```

For example, if there are customized manual pages which are formatted with reverse or fractional line feeds, the **PAGER** environment variable may be set to **/usr/bin/pg** so that the line feeds are not printed as control characters. This procedure is not necessary for the manual pages.

When the **man** command uses a hypertext database, it can retrieve several articles. For example, **man open** displays several articles. The use of **SIGINT** (Ctrl-C) exits the **man** command completely. On the other hand, **man open c|ose** also displays several articles but the use of **SIGINT** (Ctrl-C) causes **man**

to display the **close** command information instead of exiting. Using **SIGINT** (Ctrl-C) again exits the **man** command completely.

When specifying one of the Network Computing System library routines that contains a **\$** (dollar sign) in its name, enter a **** (backslash) preceding the **\$**.

Flags

Item	Description
-a	Display all matching entries.
-c	Displays the manual information using the cat command.
-f	Displays entries in the keyword database related only to the command name given as the final parameter. You can enter more than one command name, each separated by a space. Use this flag to search for command articles only. To use the -f flag, a root user must have previously entered <code>catman -w</code> to create the <u>/usr/share/man/whatis</u> file.
-F	Display only the first matching entry.
-k	Displays each line in the keyword database that contains a string of characters matching the title given as the final parameter. You can enter more than one title, each separated by a space. To use the -k flag, a root user must have previously entered <code>catman -w</code> to create the <u>/usr/share/man/whatis</u> file.
-m	Only search in the paths specified in MANPATH or -M .
-Mpath	Changes the standard location where the man command searches for manual information. The path is a colon-separated list of paths, where the following special symbols can be used: <ul style="list-style-type: none">• %D -• The default AIX paths for man pages.• %L - A locale-specific directory location corresponding to the LC_MESSAGES category of the current locale.• %l - A locale-specific directory location corresponding to the first 2 characters of the LC_MESSAGES category of the current.
-r	Searches remotely for the manual information. If for any reason the remote search fails, then man performs a local search for the requested man page. Any of the following conditions can cause the remote search to fail: <ul style="list-style-type: none">• The remote machine is not reachable.• There is a problem reading the URL.• A Java™ applet is not installed or it is not found in the user's search path, specified in the PATH environment variable. <p>Note: The DOCUMENT_SERVER_MACHINE_NAME environment variable should be set to the name of the documentation search server machine the user wants to use. If the AIX Base Documentation is not supported for the host's locale, the man command searches for the documentation for an alternate locale. If the search is successful, the documentation page is displayed after conversion to the local host's locale. If the alternate locale is not installed on the local host, the man command fails to display the documentation page.</p>
-t	Formats the manual information using the troff command. This flag is ignored if the manual page is found in a hypertext information base.

Exit Status

This command returns the following exit values:

Item	Description
------	-------------

0	Successful completion.
---	------------------------

>0	An error occurred.
----	--------------------

Examples

1. To display information about the **grep** command, enter the following command:

```
man grep
```

2. To display information about the **rpc_\$register** library routine, enter the following command:

```
man rpc_{$register
```

3. To display all entries in the **/usr/share/man/whatis** keyword database that contain the "mkdir" string, enter the following command:

```
man -k mkdir
```

The output is equivalent to the **apropos** command. You receive output from the **-k** flag only when the **/usr/share/man/whatis** keyword database already exists.

4. To display all entries from the keyword database related to the **nroff** and **troff** commands, enter the following command:

```
man -f nroff troff
```

The output is equivalent to the **whatis** command. You receive output from the **-f** flag only when the **/usr/share/man/whatis** keyword database already exists.

5. To display all **ftp** command related articles in the **/usr/share/man** or **/usr/share/man/local** path, enter the following command:

```
man -M/usr/share/man:/usr/share/man/local ftp
```

6. To display all matching entries, enter the following command:

```
man -a title
```

7. To display only the first matching entry, enter the following command:

```
man -F title
```

8. To search only in the paths specified in MANPATH or **-M**, enter the following command:

```
man -m -M PATH title
```

9. To search in the user-defined PATH, enter the following command:

```
man -M PATH title
```

Files

Item	Description
------	-------------

Item	Description
<code>/usr/share/man</code>	Standard manual directory structure.
<code>/usr/share/man/cat?/*</code>	Directory containing preformatted pages.
<code>/usr/share/man/whatis</code>	Contains the keyword database.
<code>/usr/share/man/man?/*</code>	Directory containing nroff format manual pages.

manage_disk_drivers Command

Purpose

Displays information about storage families and the driver that manages each family or changes the driver that manages a storage family.

Syntax

```
manage_disk_drivers [ -l ]
```

```
manage_disk_drivers [ -d [ device ] ] [ -o [ driver_option ]
```

```
manage_disk_drivers [ -h ]
```

Description

The **manage_disk_drivers** command displays a list of storage families and the driver that manages or supports each family. A storage family is a storage model. All disks within the family must be managed by the same driver.

There are two types of drivers supported: multipath I/O (MPIO) and non-MPIO. Most users select the MPIO version of the driver (AIX_AAPCM), which is the default behavior. However, there are instances where a third-party multipath driver is installed. In this case, you might want to disable the built-in AIX MPIO feature. You can perform this operation by selecting the AIX_non_MPIO option as the driver option.

Note: Not all the storage families support the AIX_non-MPIO driver.

If you specify the **-d** flag with the storage device name, the **manage_disk_drivers** command changes the driver selection to the alternate supported driver that is specified with the **-o** flag.

The **manage_disk_drivers** command saves the Object Data Manager (ODM) database that reflects the driver change, and displays a message to restart the system for the change to take effect.

Flags

Item	Description
-l	Lists all the storage devices and their supported drivers.
-d	Specifies the storage device family when you are selecting a driver. Note: The -o flag is required to use this flag.
-o	Specifies the driver to be used for the selected storage family. Note: The -d flag is required to use this flag.

NO_OVERRIDE driver option

If the present driver attribute is set to NO_OVERRIDE, the AIX operating system selects an alternate path control module (PCM), for example, a storage vendor PCM, if it is installed.

Examples

1. To list all the storage families and their supported drivers, enter the following command:

```
manage_disk_drivers -l
```

The system displays a message similar to the following example:

Device	Present Driver	Driver Options
2810XIV	AIX_AAPCM	AIX_AAPCM,AIX_non_MPIO
XIVCTRL	MPIO_XIVCTRL	MPIO_XIVCTRL,nonMPIO_XIVCTRL
2107DS8K	NO_OVERRIDE	NO_OVERRIDE,AIX_AAPCM,AIX_non_MPIO
IBMFlash	NO_OVERRIDE	NO_OVERRIDE,AIX_AAPCM,AIX_non_MPIO
IBMSVC	NO_OVERRIDE	NO_OVERRIDE,AIX_AAPCM,AIX_non_MPIO

2. To modify the driver to use the AIX_non_MPIO option that manages 2810XIV device, enter the following command:

```
manage_disk_drivers -d 2810XIV -o AIX_non_MPIO
```

The system displays a message similar to the following example:

```
# manage_disk_drivers -d 2810XIV -o AIX_non_MPIO
***** ATTENTION *****
For the change to take effect the system must be rebooted
```

managefonts Command

Purpose

Provides the user with a simple menu-based interface to update or change the set of installed font families on the system.

Note: You must have root user authority to run the **managefonts** script. The **managefonts** script is contained in the **/usr/lib/ps/ditroff.fonts/managefonts** file.

Syntax

```
managefonts [ Option ]
```

Description

The **managefonts** command provides the user with a simple menu-based interface to update or change the set of installed font families on the system. If no command line arguments are provided, the menu-based interface is used. Command-line arguments can be used to provide the equivalent of the menu selections.

A set of font families is installed on the system at the time the TranScript Tools option of the Text Formatter Services Package is installed on the system. This default setup includes the standard 13 fonts comprising the Times, Courier, and Helvetica font families. You can use the program called up by the **managefonts** command to erase the current configuration and replace it with a new one. There are several predefined packages of font families that can be installed this way:

Item	Description
Times Family Only	This is the most minimal configuration that allows the TranScript Tools option to run.
Standard13 Package	This package builds the Times, Courier, and Helvetica font families. This was the package installed on your system with TranScript.

Item	Description
Standard35 Package	This font family package includes the Standard13 package font families in addition to the following: Avant Garde, Bookman, New Century Schoolbook, and Palatino font families.
All Font Families	This package installs all the font families available for installation.

You can also use the **managefonts** command to add new font families one at a time. A menu of available fonts is displayed and users can select which font family they want to be built. The program prevents building of font families that are already installed.

The **managefonts** command includes help screens to assist the user in installing font families.

Notes:

1. Font families cannot be deleted directly. To delete font families, it is first necessary to install a package containing the minimal subset of families desired. After the package is installed, it is possible to add font families, one at a time, from the Individual Fonts Menu. For instance, if your current configuration is Times, Courier, and Helvetica, and you want only Times and Courier, you can use the **managefonts** program to install the Times Only Package.
2. There is no command-line syntax equivalent to the menu items in the **managefonts** program.

The command line arguments are acted upon in the order they are given, reading left to right. The following are the valid values for the *option* parameter and their meanings:

Item	Description
init0	Initialize for the installation of a font package.
clean	Remove all temporary files and previously installed fonts.
cleanall	Remove all the temporary files, the previously installed fonts, and the TranScript troff font files installed.
default	Install the Standard 13 fonts.
standard13	Install the Standard 13 fonts.
standard35	Install the Standard 35 fonts.
all	Install all possible fonts.
CourierFamily	Install the Courier Family.
HelveticaFamily	Install the Helvetica Family.
HelvNarrowFamily	Install the Helvetica Narrow Family.
AvantGardeFamily	Install the Avant Garde Family.
BookmanFamily	Install the Bookman Family.
GaramondFamily	Install the Garamond Family.
LubalinFamily	Install the Lubalin Family.
NewCenturyFamily	Install the New Century Family.
OptimaFamily	Install the Optima Family.
PalatinoFamily	Install the Palatino Family.
SouvenirFamily	Install the Souvenir Family.
ZapfFamily	Install the Zapf Family.
BaseFamily	Install the Base Family, such as Times Roman.

Examples

1. To install the standard 13 fonts:

```
managefonts cleanall standard13
```

2. To install the standard 35 fonts:

```
managefonts cleanall standard35
```

3. To install all the fonts:

```
managefonts cleanall all
```

4. To install the Courier Family (the Times Roman or Base Family must have been previously installed):

```
managefonts init0 CourierFamily clean
```

mant Command

Purpose

Typesets manual pages.

Syntax

```
mant [ -M Media ] [ -a ] [ -c ] [ -e ] [ -t ] [ -z ] [ -T Name ] [ troffFlags ] [ File ... | - ]
```

Description

The **mant** command uses the manual page macros (**man** macro package) to typeset manual pages. The *File* parameter specifies the files to be processed by the **mant** command. Files must be displayed after all flags. If no file name is specified, the **mant** command prints a list of its flags. If a - (minus sign) is specified for the *File* parameter, standard input is read.

The **mant** command has flags to specify preprocessing by the **tbl** command, **cw** command, or **eqn** command. Flags from the **troff** command can be specified with the *troffFlags* parameter.

If the input contains a **troff** command comment line consisting solely of the string '`\ " x`' (single quotation mark, backslash, double quotation mark, *x*), where *x* is any combination of the three letters **c**, **e**, and **t**, and where there is exactly one character space between the double quotation mark and *x*, then the input is processed through the appropriate combination of the **cw** command, **eqn** command, and **tbl** command, respectively, regardless of the command-line options.

Note: Use the **-oList** flag of the **troff** command to specify ranges of pages to be output. Calling the **mant** command with one or more of the **-c** flag, **-e** flag, **-t** flag, and - (minus) flags together with the **-oList** flag of the **troff** command, give a broken pipe message if the last page of the document is not specified by the *List* variable. This broken pipe message is not an indication of any problem and can be ignored.

The **mant** command, unlike the **troff** command, automatically pipes its output to a specific postprocessor, according to the following flags, environment variable, or default setting unless specifically requested not to do so:

Item	Description
-z	Indicates that no postprocessors are used.
-TName	Prepares the output for the printing device specified by the <i>Name</i> variable.
TYPESETTER	Specifies a particular printing device for the system environment.

Item	Description
default	Sends to ibm3816 .

Flags, other than the ones in the following list, are passed to the **troff** command or to the macro package, as appropriate. All flags must be displayed before the specified file names.

Flags

All flags must appear before the specified file names.

Item	Description
-a	Calls the -a flag of the troff command.
-c	Preprocesses the input files with the cw command.
-e	Preprocesses the input files with the eqn command.
-M Media	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: <ul style="list-style-type: none"> 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. <p style="margin-left: 40px;">Note: The <i>Media</i> variable is not case-sensitive.</p>
-t	Preprocesses the input files with the tbl command.
-z	Prepares the output without the postprocessor.
-TName	Prepares the output for the specified printing device. Possible <i>Name</i> variables are: <ul style="list-style-type: none"> ibm3812 3812 Pageprinter II. ibm3816 3816 Pageprinter. hplj Hewlett-Packard LaserJet II. ibm5587G 5587-G01 Kanji Printer multi-byte language support. psc PostScript printer. X100 AIXwindows display.
-	Forces input to be read from standard input.

mark Command

Purpose

Creates, modifies, and displays message sequences.

Syntax

```
mark [ +Folder ] [ -list ] [ -sequence Name [ Messages... ] [ -add | -delete ] [ -zero | -nozero ] [ -public | -npublic ] ]
```

Description

The **mark** command creates, deletes, adds, and lists messages in a sequence. The **mark** command by default lists all of the sequences and their messages for the current folder. If you use the **-add** or **-delete** flag, you must also use the **-sequence** flag. When all messages are deleted from a sequence, the **mark** command removes the sequence name from the folder.

To create a new sequence, enter the **-sequence** flag with the name of the sequence you want to create. The **mark** command creates the sequence starting with the current message. By default, the **mark** command places the sequence in the current folder. If you specify a folder, that folder becomes the current folder.

Flags

Item	Description
-add	Adds messages to a sequence. The -add flag is the default. If you do not specify a message, the mark command uses the current message. Note: You can only use this flag with the -sequence flag.
-delete	Deletes messages from a sequence. If you do not specify a message, the current message is deleted by default. Note: You can only use this flag with the -sequence flag.
<i>+Folder</i>	Specifies the folder to examine.
-help	Lists the command syntax, available switches (toggles), and version information. Note: For MH, the name of this flag must be fully spelled out.
-list	Displays the messages in a sequence. By default, the -list flag displays all the sequence names and messages defined for the current folder. To see a specific sequence, use the -sequence flag with the -list flag.
-npublic	Restricts a sequence to your usage. The -npublic flag does not restrict the messages in the sequence, only the sequence itself. This option is the default if the folder is write-protected from other users.
-nozero	Modifies the sequence by adding or deleting only the specified messages. This flag is the default.
-public	Makes a sequence available to other users. The -public flag does not make protected messages available, only the sequence itself. This flag is the default if the folder is not write-protected from other users.
-sequence <i>Name</i>	Specifies a sequence for the -list , -add , and -delete flags. You cannot use new as a sequence name.

Item	Description
-zero	Clears a sequence of all messages except the current message. When the -delete flag is also specified, the -zero flag places all of the messages from the folder into the sequence before deleting any messages.
<i>Messages</i>	Specifies messages in a sequence. You can specify more than one message at a time. Messages are identified with following references: <ul style="list-style-type: none"> Number Number of the message all All the messages in a folder cur or . (period) Current message (the default) first First message in a folder last Last message in a folder new New messages in a folder next Message following the current message prev Message preceding the current message <p>If the -list flag is used, the default for the <i>Messages</i> parameter is all. Otherwise, the default is the current message.</p>

Profile Entries

The following entry is found in the *UserMHDirectory/context* file:

Item	Description
<code>Current-Folder:</code>	Specifies the default current folder.

The following entry is found in the **\$HOME/.mh_profile** file:

Item	Description
<code>Path:</code>	Specifies the MH 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

- To see the list of all sequences defined for the current folder, enter:

```
mark
```

The system displays a message similar to the following:

```
cur: 94
test: 1-3 7 9
```

In this example, message 94 is the current message number in the current folder. The message sequence called `test` includes message numbers 1, 2, 3, 7, and 9.

2. To see the list of all the sequences defined for the `meetings` folder, enter:

```
mark +meetings
```

The system displays a message similar to the following:

```
cur: 5
dates: 12 15 19
```

3. To create a new message sequence called `schedule` in the current folder, enter:

```
mark -sequence schedule
```

The system displays the shell prompt to indicate that the `schedule` sequence was created. By default, the system adds the current message to the new sequence.

4. To delete message 10 from the `schedule` sequence, enter:

```
mark -sequence schedule 10 -delete
```

Files

Item	Description
<code>\$HOME/.mh_profile</code>	Specifies the MH user profile.
<code>/usr/bin/mark</code>	Contains the mark command.

mesg Command

Purpose

Permits or refuses write messages.

Syntax

```
mesg [ n | y ]
```

Description

The **mesg** command controls whether other users on the system can send messages to you with either the **write** command or the **talk** command. Called without arguments, the **mesg** command displays the current workstation message-permission setting.

The shell startup process permits messages by default. You can override this default action by including the line `mesg n` in your `$HOME/.profile` file. A user with root user authority can send write messages to any workstation, regardless of its message permission setting. Message permission has no effect on messages delivered through the electronic mail system.

If you add `mesg y` to your `$HOME/.profile`, you will be able to receive messages from other users via the **write** command or the **talk** command.

If you add `mesg n` to your `$HOME/.profile`, you will not be able to receive messages from other users using the **write** command or the **talk** command.

Flags

Item	Description
-------------	--------------------

- | | |
|----------|---|
| n | Allows only the root user the permission to send messages to your workstation. Use this form of the command to avoid having others clutter your display with incoming messages. |
| y | Allows all workstations on the local network the permission to send messages to your workstation. |

Exit Status

This command returns the following exit values:

Item	Description
-------------	--------------------

- | | |
|--------------|------------------------------------|
| 0 | Receiving messages is allowed. |
| 1 | Receiving messages is not allowed. |
| >1 | An error occurred. |

Examples

1. To allow only the root user the permission to send messages to your workstation, enter:

```
mesg n
```

2. To allow everyone the permission to send messages to your workstation, enter:

```
mesg y
```

3. To display what your current message-permission setting is, enter:

```
mesg
```

Information similar to the following is displayed:

```
is y
```

In the previous example, the current message-permission setting is y (allowing all users on the local network the permission to send messages to your workstation). If you change the message-permission setting to n (allowing only the root user the permission to send messages to your workstation), information similar to the following is displayed:

```
is n
```

Files

Item	Description
/dev/tty*	Supports the controlling terminal interface.
\$HOME/.profile	Controls startup processes and daemons.

mhl Command

Purpose

Produces formatted listings of messages.

Syntax

mhl [**-form** *FormFile*] [**-folder** *+Folder*] [**-moreproc** *Command* | **-nomoreproc** [**-bell** | **-nobell**] [**-clear** | **-noclear**]] [**-length** *Number*] [**-width** *Number*]

Description

The **mhl** command creates formatted lists of messages. The command is usually started through the `showproc`: profile entry or through the **-showproc** flag in other MH commands. When displaying messages, the **mhl** command uses the directions listed in the format file. If you specify more than one message, the **mhl** command provides a prompt before displaying each screen of messages.

If the **-nomoreproc** flag is specified, the **mhl** command prompts the user to press the Return key (the Ctrl-D key sequence is also acceptable) to see the next message. To stop the current message output and receive a prompt for the next message, press the Ctrl-D key sequence. Press the QUIT key sequence to stop the command output.

Note: To use the **mhl** command, you must make the folder you wish to work with the current directory.

Flags

Item	Description
-bell	Produces a bell at the end of each page. When the -nomoreproc flag is specified or the <code>moreproc</code> : profile entry is defined, but empty, the -bell flag is the default.
-clear	Clears the screen after each page when the output device is a display. The mhl command uses the \$TERM environment variable to determine the type of display. When the output device is not a display, the -clear flag inserts a form feed character at the end of each message. This flag affects the mhl command only if the <code>moreproc</code> : profile entry is defined and empty.
-folder <i>+Folder</i>	Identifies the folder to be used for the mhl.format file's MessageName: entry. The default is the value of the \$mhfolder environment variable.
-form <i>FormFile</i>	Specifies a file containing an alternate output format. The default format is described in the <i>UserMHDirectory/mhl.format</i> file. If this file does not exist, the mhl command uses the system default format described in the <i>/etc/mh/mhl.format</i> 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.
-length <i>Number</i>	Sets the screen length for the output. The default is the value indicated by the \$TERM environment variable. If that value is not appropriate, the default is 40 lines.
-moreproc <i>Command</i>	Uses the value of the <i>Command</i> variable instead of the value of the <code>moreproc</code> : entry specified in the \$HOME/.mh_profile file.
-nobell	Suppresses the bell at the end of each page. This flag affects the mhl command only if the output device is a display, the -nomoreproc flag is used, or the <code>moreproc</code> : profile entry is defined and empty.

Item	Description
-noclear	Prevents clearing of the screen at the end of each page when the output device is a display. When the output device is not a display, the -clear flag does not insert a form-feed character at the end of each message. This flag is the default when the -moreproc flag is used or the <code>moreproc :</code> entry is defined and is empty.
-nomoreproc	Sets the <code>moreproc :</code> entry as an empty value.
-width <i>Number</i>	Sets the screen width for the output. The default is the value indicated by the \$TERM environment variable. If that value is not appropriate, the default is 80 characters.

Profile Entries

The following entry is found in the *UserMHDDirectory/.mh_profile* file:

Item	Description
<code>moreproc :</code>	Specifies the interactive program for communicating with the user.

Examples

1. To list message 5 in the **inbox** folder, change the directory to **inbox**:

```
cd /home/mickey/Mail/inbox
```

Then enter:

```
/usr/lib/mh/mhl 5
```

A display similar to the following appears:

```
--- Using template MHL.FORMAT ---
Date:

To:
cc:

From:
Subject:

Message Text
```

2. To display more than one message, enter:

```
/usr/lib/mh/mhl 5 6 7
```

Files

Item	Description
\$HOME/.mh_profile	Contains the MH user profile.
/etc/mh/mhl.format	Defines the default MH message template.
<i>UserMHDirectory/mhl.format</i>	Specifies a user's default message template. (If it exists, it overrides the default MH message template.)
/usr/lib/mh/mhl	Contains the mhl command.

mhmail Command

Purpose

Sends or receives mail.

Syntax

```
mhmail User ... [ -cc User ... ] [ -from User ... ] [ -subject "String" ] [ -body "String" ]
```

Description

The **mhmail** command composes, sends, and files messages. To file a message, enter the **mhmail** command without any flags. The default folder is **\$HOME/inbox**.

If you specify one or more user addresses with the *User* parameter, the **mhmail** command accepts text from your terminal and composes a message. You can end the message text by pressing the Ctrl-D key sequence. The **mhmail** command sends a copy of the message to each specified address.

Flags

Item	Description
-body "String"	Sends a message with the specified string as the body. You must enclose the string in quotes. When you specify the -body flag, the mhmail command does not accept text from the terminal.
-cc User...	Sends a copy of the message to the specified users. The mhmail command puts the addresses in the cc : field.
-from User...	Places the specified user address in the From : field of 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.
-subject "String"	Places the specified text string in the Subject : field of the message.

Examples

1. To receive new mail and file it into the default mail folder, **\$USER/Mail/inbox**, enter:

```
mhmail
```

The system displays a message similar to the following:

```
Incorporating new mail into inbox...
65+ 04/08 jim@athena.a Meeting      <<The meeting will
66  04/08 jim@athena.a Schedule    <<Schedule change
```

In this example, two messages are filed in the inbox file. The subject of the first message is Meeting, and the first line starts with the words The meeting will. The subject of the second message is Schedule, and the first line starts with the words Schedule change.

2. To send a message regarding a schedule change to user jamie on system venus, enter:

```
mhmail jamie@venus -subject "Schedule Change"
```

The system waits for you to enter the text of the message. After completing the last line of the text, press the Enter key and then the Ctrl-D key sequence to send the message.

Files

Item	Description
<code>/var/spool/Mail/\$USER</code>	Defines the location of the mail drop.
<code>/usr/bin/mhmail</code>	Contains the mhmail command.

mhpath Command

Purpose

Prints full path names of messages and folders.

Syntax

```
mhpath [ +Folder ] [ Messages [ ,Messages ] ... ]
```

Description

The **mhpath** command lists the path names of folders and messages. By default, the command lists the path name of the current folder.

Flags

Item	Description
<code>+Folder</code>	Specifies which folder path to list.
<code>-help</code>	Lists the command syntax, available switches (toggles), and version information.

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

Item	Description
<i>Messages</i>	<p>Specifies the messages for which you want to list path names. The <i>Messages</i> parameter can specify several messages, a range of messages, or a single message. Use the following references to specify messages.</p> <p>Number Number of the message. When specifying multiple messages, separate each message number with a comma. When specifying a range of messages, separate the upper and lower ends of the range with a hyphen.</p> <p style="padding-left: 40px;">Note: You cannot use the new variable when specifying a range.</p> <p>Sequence A group of messages specified by the user. Recognized values include:</p> <p>all All the messages in a folder.</p> <p>cur or . (period) Current message.</p> <p>first First message in a folder.</p> <p>last Last message in a folder.</p> <p>new Path name that the system will assign to the next message that is incorporated.</p> <p>next Message following the current message.</p> <p>prev Message immediately before the current message.</p>

Profile Entries

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

Item	Description
Current-Folder:	Sets the default current folder.
Path:	Specifies a user's MH 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

1. To list the path name of the current folder, enter:

```
mhpath
```

The system responds with a message similar to the following:

```
/home/tom/Mail/inbox
```

2. To list the path names for messages 2 through 4 in the **source** folder, enter:

```
mhpath +source 2-4
```

The system responds with a message similar to the following:

```
/home/tom/Mail/source/2  
/home/tom/Mail/source/3  
/home/tom/Mail/source/4
```

3. To list the path name the system will assign to the next message added to the current folder, enter:

```
mhpath new
```

The system responds with a message similar to the following:

```
/home/tom/Mail/source/5
```

In this example, the next message will be message 5 in user tom's current folder, /home/tom/Mail/source.

Files

Item	Description
\$HOME/.mh_profile	Defines the user's MH profile.
/usr/bin/mhpath	Contains the mhpath command.

migratelp Command

Purpose

Moves allocated logical partition from one physical partition to another physical partition on a different physical volume.

Syntax

```
migratelp LVname/LPartnumber [ /Copynumber ] DestPV[/PPartNumber]
```

Description

The **migratelp** moves the specified logical partition *LPartnumber* of the logical volume *LVname* to the *DestPV* physical volume. If the destination physical partition *PPartNumber* is specified it will be used, otherwise a destination partition is selected using the intra region policy of the logical volume. By default the first mirror copy of the logical partition in question is migrated. A value of 1, 2 or 3 can be specified for *Copynumber* to migrate a particular mirror copy.

Note:

1. You must consider the partition usage, reported by **lvmstat**, on the other active concurrent nodes in case of a concurrent volume group.
2. Strictness and upper bound settings are not enforced when using **migratelp**.
3. Running this command on an active, firmware-assisted, dump logical volume temporarily changes the location of the dump device to **/dev/sysdumpnull**. After you have successfully migrated the logical volume, this command calls the following command to set the firmware-assisted, dump logical volume to the original logical volume.

```
sysdumpdev -P
```

The **migratelp** command fails to migrate partitions of striped logical volumes.

Security

To use **migratelp**, you must have root user authority.

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 move the first logical partitions of logical volume lv00 to hdisk1, type:

```
migratelp lv00/1 hdisk1
```

2. To move second mirror copy of the third logical partitions of logical volume hd2 to hdisk5, type:

```
migratelp hd2/3/2 hdisk5
```

3. To move third mirror copy of the 25th logical partitions of logical volume testlv to 100th partition of hdisk7, type:

```
migratelp testlv/25/3 hdisk7/100
```

Files

Item	Description
<code>/usr/sbin</code>	Directory where the migratelp resides.

migratepv Command

Purpose

Moves allocated physical partitions from one physical volume to one or more other physical volumes.

Syntax

```
migratepv [ -i ] [ -l LogicalVolume ] SourcePhysicalVolume DestinationPhysicalVolume...
```

Description

The **migratepv** command moves allocated physical partitions and the data they contain from the *SourcePhysicalVolume* to one or more other physical volumes. To limit the transfer to specific physical volumes, use the names of one or more physical volumes in the *DestinationPhysicalVolume* parameter; otherwise, all the physical volumes in the volume group are available for the transfer. All physical volumes must be within the same volume group. The specified source physical volume cannot be included in the list of *DestinationPhysicalVolume* parameters.

Note:

1. To use this command, you must either have root user authority or be a member of the **system** group.
2. The **migratepv** command is not allowed on a snapshot volume group or a volume group that has a snapshot volume group.
3. Running this command on a physical volume that has an active, firmware-assisted, dump logical volume temporarily changes the location of the dump device to **/dev/sysdumpnull**. After you have

successfully migrated the logical volume, this command calls the following command to set the firmware-assisted, dump logical volume to the original logical volume.

```
sysdumpdev -P
```

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

When you migrate a physical volume, the boot logical volume must remain intact. Two contiguous physical partitions and the new boot image must be built on the new boot logical volume.

If you specify a logical volume that contains the boot image, the **migratepv -l** command attempts to find enough contiguous partitions on one of the target physical volumes. If the migration is successful, the **migratepv** command prints a message that recommends the user run the **bosboot** command to indicate a change in the boot device. The attempted migration fails if the **migratepv -l** command is unable to find enough contiguous space to satisfy the request.

Note: All Logical Volume Manager migrate functions work by creating a mirror of the logical volumes involved, then resynchronizing the logical volumes. The original logical volume is then removed. If the **migratepv** command is used to move a logical volume containing the primary dump device, the system will not have an accessible primary dump device during the execution of the command. Therefore, a dump taken during this execution may fail. To avoid this, reassign the primary dump device using the **sysdumpdev** command or ensure there is a secondary dump device defined before using **migratepv**.

You can use the System Management Interface Tool (SMIT) **smit migratepv** fast path to run this command.

Flags

Item	Description
-i	Reads the <i>DestinationPhysicalVolume</i> parameter from standard input.
-l LogicalVolume	Moves only the physical partitions allocated to the specified logical volume and located on the specified source physical volume.

Examples

1. To move physical partitions from hdisk1 to hdisk6 and hdisk7, enter:

```
migratepv hdisk1 hdisk6 hdisk7
```

Physical partitions are moved from one physical volume to two others within the same volume group.

2. To move physical partitions in logical volume lv02 from hdisk1 to hdisk6, enter:

```
migratepv -l lv02 hdisk1 hdisk6
```

Only those physical partitions contained in lv02 are moved from one physical volume to another.

Files

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

migwpar Command

Purpose

After an operating system migration of a global system to version 7, the **migwpar** command is used to migrate a workload partition (WPAR) that was created on the version 6 global system, to version 7.

The **migwpar** command can also be used to migrate from a 5.2 or 5.3 versioned WPAR to a native version 7 WPAR.

The **migwpar** command can also be used to enable a 5.2 or 5.3 versioned WPAR to work after the global system is migrated to a new operating system level. The versioned WPAR remains at the previous level.

Syntax

```
migwpar [ -d<software_source> ] [-V]{ -A | -f <wparNamesFile> | [ -C ] wpar_name }
```

Description

After an operating system migration of a global system to version 7, the **migwpar** command is used to migrate a workload partition (WPAR) that was created on the version 6 global system, to version 7. Software that no longer exists (it might be replaced by a different software package) on the global system, is removed.

Ensure that all software on the global system has been migrated before you begin to migrate the WPARs. The use of the `pre_migration` script before the global system migration, and the `post_migration` script after the migration, will provide data that can be used to verify the migration, such as listing software that will be removed during the migration, and software that did not migrate.

The **-C** flag is used to migrate from a 5.2 or 5.3 versioned WPAR to a native version 7 WPAR. The **-C** flag is incompatible with the list flags (**-A** or **-f**). When you migrate a versioned WPAR, the **-d** `software_source` information is mandatory.

The **-V** flag can be used to enable a versioned WPAR after the global system is migrated. This option maintains the WPAR at its current level and allows the WPAR to work on the newly migrated global system. This functionality is available on IBM AIX 7.2 with Technology Level 2 or later. Only `rootvg` versioned WPARs require enablement by using the **-V** option.

Note: It is highly recommended to back up the WPAR before you begin to migrate.

A log of all actions of the **migwpar** command is saved in the `/var/adm/ras/migwpar.log` file. The output of the actual software migration of each WPAR is saved in `/var/adm/ras/devinst.log` within the WPAR.

Flags

Item	Description
-A	Migrates all migratable WPARs
-f <i>wparNamesFile</i>	Migrates the list of WPARs contained in the file <i>wparNamesFile</i> , one per line.
-C <i>wparName</i>	Migrates the specified 5.2 or 5.3 versioned WPAR.
-d <i>software_source</i>	Specifies the installation location that is used for the detached WPAR migration. Note: The install (or update) images in the specified location must be the same as the ones used to install (or update) the global system.
-V	Enables a WPAR after the global system is migrated. The -V option is not compatible with the -C or -d flags.

Security

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

Examples

1. After the base operating system of GLOBAL has been migrated to version 7 to migrate a single WPAR, wpar1 to version 7, enter the following command:

```
# migwpar wpar1
```

2. To migrate a detached WPAR, wpar2 by using install images from **/images**, enter the following command:

```
# migwpar -d /images wpar2
```

3. To migrate all shared WPARs, enter the following command:

```
migwpar -A
```

4. To migrate all detached WPARs by using install images in **/images**, enter the following command:

```
migwpar -A -d /images
```

5. To migrate a 5.2 versioned WPAR wpar_52 by using install images in **/images**, enter the following command:

```
migwpar -d /images -C wpar_52
```

6. To enable all versioned WPARs on a system after a global system is migrated, enter the following command:

```
migwpar -VA
```

mirrorvg Command

Purpose

Mirrors all the logical volumes that exist on a given volume group.

Syntax

```
mirrorvg [ -S | -s ] [ -Q ] [ -c copies ] [ -m ] [ -p copyn=mirrorpool ] volumegroup [ physicalvolume ... ]
```

Description

The **mirrorvg** command takes all the logical volumes on a given volume group and mirrors those logical volumes. This same functionality may also be accomplished manually if you execute the **mklvcopy** command for each individual logical volume in a volume group. As with **mklvcopy**, the target physical drives to be mirrored with data must already be members of the volume group. To add disks to a volume group, run the **extendvg** command.

By default, **mirrorvg** attempts to mirror the logical volumes onto any of the disks in a volume group. If you wish to control which drives are used for mirroring, you must include the list of disks in the input parameters, *physicalvolume*. Mirror strictness is enforced. Additionally, **mirrorvg** mirrors the logical volumes, using the default settings of the logical volume being mirrored. If you wish to violate mirror strictness or affect the policy by which the mirror is created, you must execute the mirroring of all logical volumes manually with the **mklvcopy** command.

When **mirrorvg** is executed, the default behavior of the command requires that the synchronization of the mirrors must complete before the command returns to the user. If you wish to avoid the delay, use the **-S** or **-s** option. Additionally, the default value of 2 copies is always used. To specify a value other than 2, use the **-c** option.

Restrictions:

- To use this command, you must either have root user authority or be a member of the **system** group.
- You cannot use the **mirrorvg** command on a snapshot volume group.
- You cannot use the **mirrorvg** command on a volume group that has an active firmware assisted dump logical volume.



Attention: The **mirrorvg** command may take a significant amount of time before completing because of complex error checking, the amount of logical volumes to mirror in a volume group, and the time it takes to synchronize the new mirrored logical volumes.

You can use the System Management Interface Tool (SMIT) **smit mirrorvg** fast path to run this command.

Flags

Item	Description
-c <i>copies</i>	Specifies the minimum number of copies that each logical volume must have after the mirrorvg command has finished executing. It may be possible, through the independent use of mklvcopy , that some logical volumes may have more than the minimum number specified after the mirrorvg command has executed. Minimum value is 2 and 3 is the maximum value. A value of 1 is ignored.
-m <i>exact map</i>	Allows mirroring of logical volumes in the exact physical partition order that the original copy is ordered. This option requires you to specify a <code>PhysicalVolume(s)</code> where the exact map copy should be placed. If the space is insufficient for an exact mapping, then the command will fail. You should add new drives or pick a different set of drives that will satisfy an exact logical volume mapping of the entire volume group. The designated disks must be equal to or exceed the size of the drives which are to be exactly mirrored, regardless of if the entire disk is used. Also, if any logical volume to be mirrored is already mirrored, this command will fail.
-p <i>copyn=mirrorpool</i>	Assigns mirror pools to the copies being created. A mirror pool is assigned to a copy using the <i>copyn=mirrorpool</i> parameter. Specify a mirror pool for each copy. To specify more than one <i>copyn=mirrorpool</i> pair, provide multiple -p <i>copyn=mirrorpool</i> flags.
-Q <i>Quorum Keep</i>	By default in mirrorvg , when a volume group's contents becomes mirrored, volume group quorum is disabled. If the user wishes to keep the volume group quorum requirement after mirroring is complete, this option should be used in the command. For later quorum changes, refer to the chvg command.
-S <i>Background Sync</i>	Returns the mirrorvg command immediately and starts a background syncvg of the volume group. With this option, it is not obvious when the mirrors have completely finished their synchronization. However, as portions of the mirrors become synchronized, they are immediately used by the operating system in mirror usage.
-s <i>Disable Sync</i>	Returns the mirrorvg command immediately without performing any type of mirror synchronization. If this option is used, the mirror may exist for a logical volume but is not used by the operating system until it has been synchronized with the syncvg command.

The following is a description of **rootvg**:

Item	Description
rootvg mirroring	<p>When the rootvg mirroring has completed, you must perform two additional tasks: bosboot and bootlist.</p> <p>The bosboot command is required to customize the bootrec of the newly mirrored drive. The bootlist command needs to be performed to instruct the system which disk and order you prefer the mirrored boot process to start.</p>
non-rootvg mirroring	<p>When this volume group has been mirrored, the default command causes Quorum to be deactivated.</p>
rootvg and non-rootvg mirroring	<p>The system dump devices, primary and secondary, should not be mirrored. In some systems, the paging device and the dump device are the same device. However, most users want the paging device mirrored. When mirrorvg detects that a dump device and the paging device are the same, the logical volume will be mirrored automatically.</p> <p>If mirrorvg detects that the dump and paging device are different logical volumes, the paging device is automatically mirrored, but the dump logical volume is not. The dump device can be queried and modified with the sysdumpdev command.</p>

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 triply mirror a volume group, enter:

```
mirrorvg -c 3 workvg
```

The logical partitions in the logical volumes held on workvg now have three copies.

2. To get default mirroring of rootvg, enter:

```
mirrorvg rootvg
```

rootvg now has two copies.

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.

4. To sync the newly created mirrors in the background, enter:

```
mirrorvg -S -c 3 workvg
```

5. To create a second and third copy of the logical volumes within `datavg`, where the physical partition maps on each disk match each other exactly, enter:

```
mirrorvg -m -c 3 datavg hdisk2 hdisk3
```

The logical partitions in the logical volumes held on `datavg` now have three copies.

Files

Item	Description
<code>/usr/sbin</code>	Directory where the mirrorvg command resides.

mirscan Command

Purpose

Search for and correct physical partitions that are stale or unable to perform I/O operations.

Syntax

```
mirscan -v vgname | -l lvname | -p pvname | -r reverse_pvname [ -a ] [ -o ] [ -q nblks ] [ -c lvcopy ]  
[ -s strictness ] [ -u upperbound ]
```

Description

The `mirscan` command examines each allocated partition on the specified device. A report is generated that lists whether the partition is stale or fresh, and lists whether it is capable of performing I/O operations. The LVM device driver is queried to determine whether the partition is stale or fresh. Regardless of whether the partition is stale or fresh, it is read to determine whether it is capable of performing I/O operations. By default the entire partition is read, but if the `-q` flag is specified, the *nblks* value determines how much of the partition will be read. If the `-a` flag is not specified, the report is printed and execution ends after all partitions are read.

If the `-a` flag is used, corrective action is taken after all the partitions have been examined. Stale partitions will be synced. If a partition is not capable of performing I/O, `mirscan` attempts to trigger bad block relocation or hardware relocation with a forced sync operation, which should write a good copy of the data to the block that is incapable of performing I/O operations. If the partition is still unreadable, the `mirscan` command attempts to migrate that partition to a new location. By default, the new location that is selected adheres to the strictness and upperbound policies for the logical volume that contains the partition. Using the `-s` flag causes the strictness value specified on the command line to override the natural strictness value of the logical volume that contains the partition. Similarly, using the `-u` flag causes the upperbound value specified on the command line to override the natural upperbound value of the logical volume that contains the partition.

The `mirscan` command prints (to standard output) a status report for the partitions scanned. If the `-a` flag is specified, the `mirscan` command also prints (to standard output) a status report containing each corrective action that is taken. If the `-o` flag is specified, the report will be in colon-separated output format. If the `-o` flag is not specified, the default behavior is to print the report in human-readable format.

Partitions on nonmirrored logical volumes are scanned and included in all reports, but no sync or migration operation is possible for such partitions. Partitions on striped logical volumes can be synced but cannot be migrated. Partitions on paging devices cannot be migrated, because this would result in a system hang if the `mirscan` process were to be paged out. Partitions on the boot logical volume cannot be migrated. Partitions on an active firmware-assisted dump logical volume cannot be migrated. An informative error message is generated in the corrective action report for each of the preceding cases.

By default, the `mirscan` command does not take any lock on the volume group. This should allow the `mirscan` command to run in the background without interfering with other `lvm` commands. If the `-a` flag is specified and there are partitions that need to be migrated, the volume group is locked, all the migration operations are performed, and the volume group lock is released. Therefore, if the `-a` flag is specified, the impact to other `lvm` commands is minimized because the volume group is only locked during the migration operations, which are all performed at once just before the end of execution.

Flags

Item	Description
-a	Specifies that corrective action should be taken.
-c <i>lvcopy</i>	Identifies a particular copy of the logical volume. The <code>-c</code> flag can only be specified in conjunction with the <code>-l</code> flag. The <code>-c</code> flag is ignored if it is used in conjunction with the <code>-p</code> , <code>-r</code> , or <code>-v</code> flag.
-l <i>lvname</i>	Specifies the logical volume to be scanned.
-o	Specifies colon-separated output format should be used for the report. If this option is not used, the default behavior is to print a report in human-readable format.
-p <i>pvname</i>	Specifies the physical volume to be scanned.
-q <i>nblks</i>	Specifies which portions of the partition should be read. If the <i>nblks</i> value is 0, only the first, middle, and last 512 bytes of each partition are read to determine whether the partition is capable of performing I/O operations. A nonzero <i>nblks</i> value indicates that only the first <i>nblks</i> 512 byte blocks of each partition should be read to determine whether the partition is capable of performing I/O operations. If the <code>-q</code> flag is not specified, the entire partition is read.
-r <i>reverse_pvname</i>	Specifies that any partitions in the volume group should be scanned if they do not reside on <i>pvname</i> but they do have a mirror copy on <i>pvname</i> . This could be run prior to removing <i>pvname</i> from the system, in case <i>pvname</i> somehow has the last good copy of a partition.
-s <i>strictness</i> (y, n, s)	Specifies a strictness value that should override the natural strictness value. Legal values are y, n, and s, where y enables strictness, n disables strictness, and s enables "superstrictness." By default, when <code>mirscan</code> has to perform a migration operation on a partition it will adhere to the natural strictness value of the logical volume that contains that partition. If the <code>-s</code> flag is used, the override strictness value will be used. If the <code>-s</code> flag is used in conjunction with the <code>-p</code> , <code>-r</code> , or <code>-v</code> flags, the override strictness value could override the natural strictness of multiple logical volumes.
-u <i>upperbound</i>	Specifies an upperbound value that should override the natural upperbound value. The upperbound value should be between 1 and the total number of physical volumes in the volume group. By default, when <code>mirscan</code> has to perform a migration operation on a partition it will adhere to the natural upperbound value of the logical volume that contains the partition. If the <code>-u</code> flag is used, the override upperbound value will be used. If the <code>-u</code> flag is used in conjunction with the <code>-p</code> , <code>-r</code> , or <code>-v</code> flags, the override upperbound value could override the natural upperbound value of multiple logical volumes.
-v <i>vgname</i>	Specifies the volume group to be scanned.

Exit Status

An exit code of 0 indicates that `mirscan` was able to complete its execution and was able to correct any error conditions that were encountered along the way. An exit code of 1 indicates that `mirscan` was able to complete its execution, but it was unable to correct every error that it found; further corrective action

is still required. For example, if corrective actions would be required but the `-a` flag was not specified, an exit code of 1 is used. An exit code of 2 indicates that `mirscan` was unable to complete its execution. For example, if the target device is not listed in the ODM, an exit code of 2 is used.

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 scan logical volume `lv33`, report the status of each partition, and have every block of each partition read to determine whether it is capable of performing I/O operations, type:

```
mirscan -l lv33
```

2. To scan logical volume `lv33`, report the status of each partition, and have only the first two blocks of each partition read to determine whether it is capable of performing I/O operations, type:

```
mirscan -l lv33 -q 2
```

3. To scan logical volume `lv33`, report the status of each partition, sync any stale partitions found, and migrate any partitions that are not capable of performing I/O operations, type:

```
mirscan -l lv33 -a
```

4. To scan every allocated logical partition on `hdisk4` and report the status of each partition, type:

```
mirscan -p hdisk4
```

5. To find every allocated partition in the volume group that resides on `hdisk4`, and scan and report the status of all partitions that do not reside on `hdisk4` but are mirror copies of a partition that resides on `hdisk4`, type:

```
mirscan -r hdisk4
```

This would be useful to run before removing `hdisk4` from the system.

6. To scan volume group `vg05`, report the status of each allocated partition, and have the first, middle, and last 512 bytes of each partition read to determine whether that partition is capable of performing I/O operations, type:

```
mirscan -v vg05 -q 0
```

Restrictions

Unmirrored partitions and striped partitions are not eligible for migration. Partitions on paging devices will not be migrated by mirror scan because it would result in a system hang if the `mirscan` process happened to get paged out. Partitions from the boot logical volume cannot be migrated.

Location

`/usr/sbin/mirscan`

Standard Output

Each line in the report corresponds to an operation on a physical partition. There are 4 types of operation that `mirscan` can perform. A *scan operation* determines whether the partition is synced and whether

it is capable of performing I/O operations. A *resync operation* is a corrective action performed on stale partitions that attempts to return them to synced state. A *force resync operation* is a corrective action performed on partitions that are not capable of performing I/O operations, in an attempt to trigger bad block relocation or hardware relocation. At the end of the force resync operation, the partition is read again to determine whether it is capable of performing I/O operations. A *migration operation* is a corrective action performed on partitions that are not capable of performing I/O operations, in an attempt to move the data to a physical location that is capable of performing I/O.

The default format for the reports contains the following column headings. If the `-o` flag is specified, no header is displayed and the output report is printed in colon-separated output format. The columns and their meanings are as follows:

Item	Description
OP	The valid values for this field are <code>s</code> , <code>r</code> , <code>f</code> , and <code>m</code> . A value of <code>s</code> signifies a scan operation. A value of <code>r</code> signifies a resync operation. A value of <code>f</code> signifies a force resync operation, which is performed in an effort to trigger bad block relocation or hardware relocation. A value of <code>m</code> signifies a migration operation.
STATUS	The valid values for this field are <code>SUCCESS</code> or <code>FAILURE</code> . For a scan operation, <code>FAILURE</code> is indicated if the partition being scanned is stale or incapable of performing I/O. For a resync operation, <code>FAILURE</code> is indicated if the partition was not synchronized. For a force resync operation, <code>FAILURE</code> is indicated if the partition is still incapable of performing I/O operations. For a migration operation, <code>FAILURE</code> is indicated if the migration operation was not completed.
PVNAME	Identifies the name of the physical volume where the partition being operated on resides. For a migration operation, <code>PVNAME</code> refers to the source physical volume and <code>TARGETPV</code> refers to the destination physical volume.
PP	Identifies the physical partition number of the partition being operated on. The first partition on a particular physical volume has a <code>PP</code> value of <code>1</code> , not <code>0</code> .
SYNC	The valid values for this field are <code>synced</code> or <code>stale</code> . The value indicated refers to the state of the partition after the operation has been completed. For example, if a resync operation succeeds, a value of <code>synced</code> will be displayed.
IOFAIL	The valid values for this field are <code>yes</code> or <code>no</code> . The value indicated refers to the state of the partition after the operation has been completed. For example, if a migration operation succeeds then a value of <code>no</code> is displayed to indicate that the partition no longer has a problem performing I/O operations.
LVNAME	Identifies the name of the logical volume where the partition being operated on resides.
LP	Identifies the logical partition number of the partition being operated on. The first partition on a particular logical volume has an <code>LP</code> value of <code>1</code> , not <code>0</code> .
CP	Identifies the logical copy number of the partition being operated on. The first logical copy of a logical volume has a <code>CP</code> value of <code>1</code> , not <code>0</code> .
TARGETPV	Identifies the name of the physical volume that was used as the target for a migration operation. For any type of operation other than a migration operation, this field is left blank.
TARGETPP	Identifies the physical partition number of the partition that was used as the target for a migration operation. For any type of operation other than a migration operation, this field is left blank. The first partition on a particular physical volume has a <code>TARGETPP</code> value of <code>1</code> , not <code>0</code> .

mkauth Command

Purpose

Creates a new user-defined authorization.

Syntax

mkauth [-R *load_module*] [*Attribute = Value ...*] *Name*

Description

The **mkauth** command creates a new user-defined authorization in the authorization database. You can create authorization hierarchies by using a dot (.) in the *Name* parameter to create an authorization of the form *ParentAuth.SubParentAuth.SubSubParentAuth....* All parent elements in the *Name* parameter must already exist in the authorization database before the new authorization is created. The maximum number of parent elements that you can use to create an authorization is 8.

If the system is configured to use multiple domains for the authorization database, the new authorization is created in the first domain specified by the **secorder** attribute in the authorizations stanza of the **/etc/nscontrol.conf** file. Use the **-R** flag to create an authorization in a specific domain.

Authorization attributes can be set at creation time through the *Attribute = Value* parameter. Every authorization that you create must have a value for the **id** authorization attribute. If you do not specify the value using the **mkauth** command, the command automatically generates a unique ID for the authorization. If you specify an ID, the value must be unique and greater than 10000.

Restriction: Authorization IDs less than 10000 are reserved for system-defined authorizations

.

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. Authorizations created in the authorization database can be assigned to roles immediately, but do not take effect until the kernel security tables is updated.

Flags

Item	Description
-R <i>load_module</i>	Specifies the loadable module to use for authorization creation.

Parameters

Item	Description
<i>Attribute = Value</i>	Initializes an authorization attribute. Refer to the chauth command for valid attributes and values.

Item	Description
<i>Name</i>	<p>Specifies a unique authorization name string.</p> <p>Restrictions on Creating Authorization Names:</p> <p>The <i>Name</i> parameter that you specify must be unique, and can be a maximum of 63 single-byte printable characters. Although the mkauth command supports multibyte authorization names, authorization names to characters are restricted within the POSIX portable file name character set. The authorization name that you specify cannot begin with <code>aix.</code> because that is the designated top-level parent for system-defined authorizations and the mkauth command only creates user-defined authorizations.</p> <p>Authorization names must not begin with a dash (-), a plus sign (+), an at sign (@), a tilde (~), or contain any space, tab or newline characters. You cannot use the keywords ALL, default, ALLOW_OWNER, ALLOW_GROUP, ALLOW_ALL, or an asterisk (*) as an authorization name. Additionally, do not use any of the following characters within an authorization string:</p> <ul style="list-style-type: none"> • : (colon) • " (quotation mark) • # (number sign) • , (comma) • = (equal sign) • \ (backslash) • / (forward slash) • ? (question mark) • ' (single quotation mark) • ` (grave accent)

Security

The **mkauth** command is a privileged command. You must assume a role that has the following authorization to run the command successfully.

Item	Description
aix.security.auth.create	Required to run the command.

Files Accessed

Item	Description
File	Mode
/etc/security/authorizations	rw

Examples

1. To create a top-level authorization `custom` and have the **mkauth** command assign an appropriate ID value, use the following command:

```
mkauth custom
```

2. To create a custom.test child authorization and assign an ID and default description, use the following command:

```
mkauth id=12000 dfltmsg="Test Authorization" custom.test
```

3. To create the custom authorization in LDAP, use the following command:

```
mkauth -R LDAP custom
```

mkboot Command

Purpose

Creates the boot image, the boot record, and the service record. This command is *not* a user-level command and is NOT supported.

Syntax

```
mkboot -d Device [ -b ] [ -D ] [ -c ] [ -h ] [ -i ] [ -I ] [ -l LVDev ] { -k Kernel | -e Expander } [ -L ] [ -s ] [ -r ] [ -p Offset ] [ -w ] -f FileSystem
```

Description

The **mkboot** command combines a kernel and file system into a boot image. The resulting image is written to standard out. It is copied to a boot device with the appropriate boot record information. The boot image can be made compressed or uncompressed and with or without a boot record at the beginning of the image. An image created for a tape is compressed with the boot record at the start of the image file. A disk boot image may be created without compression and has no boot record. The boot record is written to the first sector of the disk. The record contains information about the size and location of the image after it is written to the boot logical volume on that disk.

If the boot logical volume is mirrored, the **mkboot** command not only writes the boot image to each copy of the boot logical volume but also writes a boot record to each physical disk comprising the mirror. As long as the **mkboot** command is able to update at least one of the copies of a mirrored boot logical volume, no error is returned. To enable booting from each copy of a mirrored boot logical volume, each of the physical disks must be specified using the **bootlist** command.

The **mkboot** command is usually called by the **bosboot** command. However, you can run the **mkboot** command a second time to put expand code at the beginning of a compressed boot image.

Flags

Item	Description
-b	Zeros out save-base fields. This flag is optional.
-d <i>Device</i>	Specifies the device required for the IPL record. This flag is required.
-c	Zeros out the boot record on the device. This flag is optional.
-D	Loads the low-level debugger at boot time.
-e <i>Expander</i>	Specifies kernel expansion code to create a compressed boot image file. Either the -e flag or the -k flag must be specified.
-f <i>FileSystem</i>	Specifies the boot file system. This flag is required.
-h	Prevents the mkboot command from updating the boot header. This flag is optional.

Item	Description
-i	Writes the normal portion of the boot record.
-I (uppercase i)	Invokes the low-level debugger at boot time.
-k <i>Kernel</i>	Specifies the kernel in the boot image. Either the -k flag or the -e flag must be specified.
-l (lowercase L) <i>LVDev</i>	Specifies the logical volume device that contains the loadable boot code.
-L	Enables lock instrumentation for MP systems. This flag has no effect on systems that are not using the MP kernel.
-p <i>Offset</i>	Specifies the address to use as <code>boot_pr_start</code> field in the boot record. This flag is used in creating the CD-ROM boot image. This flag is optional.
-r	Creates an image that is read-only storage (ROS) emulation code.
-s	Writes the service portion of the boot record.
-w	Outputs first two blocks of boot logical volume before the boot image. This flag is applicable to disk boot image only.

Security

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

Examples

1. To create an uncompressed boot image, using the kernel `/usr/lib/boot/unix` and the `/tmp/bootfs` file system for the device `/dev/hdisk0`, enter

```
mkboot -d /dev/hdisk0 -k /usr/lib/boot/unix -f /tmp/bootfs \
-b -i -s > /tmp/boot.image
```

2. To clear the boot record but leave the PVID for disk `hdisk0`, enter:

```
mkboot -d /dev/hdisk0 -c
```

3. Although the `mkboot` command combines a kernel and a random access memory (RAM) file system to create one boot image, you can run the `mkboot` command a second time to put expand code at the beginning of a compressed boot image. For example, enter:

```
mkboot -b -d /dev/rmt0 -k unix -f ramfs | compress > /tmp/image
mkboot -b -i -s -d /dev/rmt0 -k bootexpand -f /tmp/image \
> bootfile
```

for a bootable tape where:

Item	Description
unix	Specifies the kernel.
ramfs	Specifies the RAM disk file system.
compress	Specifies the compression or compact routine.
bootexpand	Specifies the expansion or kernel uncompact routine.

Files

Item	Description
<code>/usr/include/sys/bootrecord.h</code>	Specifies the structure of the boot record.

mkC2admin Command

Purpose

Configure a system to operate in C2 Security Mode.

Syntax

```
mkC2admin { [ -m ] | [ -a address ] hostname }
```

Description

The **mkC2admin** command initializes the security directories for use in a C2 System configuration. The distributed database directories are created and symbolic links initialized. When a system is being configured as the Administrative Host (using the **-m** flag), an additional file system is created to hold the master copies of the administrative database files. Those files are stored in the directory **/etc/data.master** which has a logical volume name of **hd10sec**.

The administrative database files are divided into three categories. Those files that must be shared, those files that optionally may be shared, and those files that may not be shared. Optionally sharable files are described in the file **/etc/security/files.config**. That file consists of multiple lines of the format:

```
[y|n] | filename
```

and is editable by the administrator. To select an optionally sharable filename, the administrator sets the first field to the value **y**. To make an optionally sharable file be unshared, the field is set to the value **n**. All hosts in the C2 System must have an identical **/etc/security/files.config** file.

The system *hostname* must be defined in the **/etc/hosts** file at the time this command is run. If not, the IP address of the new C2 System Administrative Host may be provided with the **-a** option, and an entry will be added to **/etc/hosts**.

Flags

Item	Description
-a <i>address</i>	Use <i>address</i> as the IP address of <i>hostname</i> .
-m	Configure the host as the administrative master.

Parameters

Item	Description
<i>hostname</i>	Specifies the hostname.

Exit Status

- 0** The system has been properly configured to operate in the C2 mode.
- 1** The system was not installed with the C2 option.

2

The system could not be successfully configured to operate in C2 mode.

3

The system was previously configured to operate in C2 mode without having first been unconfigured.

Files

Item	Description
<code>/usr/sbin/mkC2admin</code>	Contains the mkC2admin command.

mkcatdefs Command

Purpose

Preprocesses a message source file.

Syntax

`mkcatdefs SymbolName SourceFile ... [-h]`

Description

The **mkcatdefs** command preprocesses a message source file for input to the **gencat** command.

The *SourceFile* message file contains symbolic identifiers. The **mkcatdefs** command produces the *SymbolName_msg.h* file, containing statements that equate symbolic identifiers with the set numbers and message ID numbers assigned by the **mkcatdefs** command.

The **mkcatdefs** command creates two outputs. The first is a header file called *SymbolName_msg.h*. You must include this *SymbolName_msg.h* file in your application program to associate the symbolic names to the set and message numbers assigned by the **mkcatdefs** command.

The **mkcatdefs** command sends message source data, with numbers instead of symbolic identifiers, to standard output. This output is suitable as input to the **gencat** command. You can use the **mkcatdefs** command output as input to the **gencat** command in the following ways:

- Use the **mkcatdefs** command with a **>** (redirection symbol) to write the new message source to a file. Use this file as input to the **gencat** command.
- Pipe the **mkcatdefs** command output file directly to the **gencat** command.
- Use the **runcat** command rather than the **mkcatdefs** command. The **runcat** command automatically sends the message source file through the **mkcatdefs** command and then pipes the file to the **gencat** command.

After running the **mkcatdefs** command, you can use symbolic names in an application to refer to messages.

Flags

Item	Description
-h	Suppresses the generation of a <i>SymbolName_msg.h</i> file. This flag must be the last argument to the mkcatdefs command.

Examples

To process the `symb.msg` message source file and redirect the output to the `symb.src` file, enter:

```
mkcatdefs symb symb.msg > symb.src
```

The generated `symb_msg.h` file looks similar to the following:

```
#ifndef _H_SYMB_MSG
#define _H_SYMB_MSG
#include <limits.h>
#include <nl_types.h>
#define MF_SYMB "symb.cat"
/* The following was generated from symb.src. */
/* definitions for set MSFAC */
#define SYM_FORM 1
#define SYM_LEN 2
#define MSG_H 6
#endif
```

The **mkcatdefs** command also creates the `symb.src` message catalog source file for the **gencat** command with numbers assigned to the symbolic identifiers:

```
$quote " Use double quotation marks to delimit message text
$delset 1
$set 1
1 "Symbolic identifiers can only contain alphanumeric \
characters or the _ (underscore character)\n"
2 "Symbolic identifiers cannot be more than 65 \
characters long\n"
5 "You can mix symbolic identifiers and numbers\n"
$quote
6 remember to include the "msg_h" file in your program
```

The assigned message numbers are noncontiguous because the source file contained a specific number. The **mkcatdefs** program always assigns the previous number plus 1 to a symbolic identifier.

Note: The **mkcatdefs** command inserts a **\$delset** command before a **\$set** command in the output message source file. This means you cannot add, delete, or replace single messages in an existing catalog when piping to the **gencat** command. You must enter all messages in the set.

Files

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

mkCCadmin Command

Purpose

Configure a system to operate in Common Criteria enabled Security Mode.

Syntax

```
mkCCadmin { [-m] | [-a address] hostname }
```

Description

The **mkCCadmin** command initializes the security directories for use in a Common Criteria enabled System configuration. The distributed database directories are created and symbolic links initialized. When a system is being configured as the Administrative Host (using the **-m** flag), an additional file

system is created to hold the master copies of the administrative database files. Those files are stored in the directory **/etc/data.master** which has a logical volume name of **hd10sec**.

The administrative database files are divided into three categories. Those files that must be shared, those files that optionally may be shared, and those files that may not be shared. Optionally sharable files are described in the file **/etc/security/files.config**. That file consists of multiple lines of the format:

```
[y|n]|filename
```

and is editable by the administrator. To select an optionally sharable filename, the administrator sets the first field to the value **y**. To make an optionally sharable file be unshared, the field is set to the value **n**. All hosts in the Common Criteria enabled System must have an identical **/etc/security/files.config** file.

The system *hostname* must be defined in the **/etc/hosts** file at the time this command is run. If not, the IP address of the new Common Criteria enabled System Administrative Host may be provided with the **-a** option, and an entry will be added to **/etc/hosts**.

Flags

Item	Description
-a <i>address</i>	Use <i>address</i> as the IP address of <i>hostname</i> .
-m	Configure the host as the administrative master.

Parameters

Item	Description
<i>hostname</i>	Specifies the hostname.

Exit Status

- 0** The system has been properly configured to operate in the Common Criteria enabled mode.
- 1** The system was not installed with the Common Criteria enabled option.
- 2** The system could not be successfully configured to operate in Common Criteria enabled mode.
- 3** The system was previously configured to operate in Common Criteria enabled mode without having first been unconfigured.

Files

Item	Description
/usr/sbin/mkCCadmin	Contains the mkCCadmin command.

mkcd Command

Purpose

Creates multi-volume CDs from a **mksysb**, **savevg**, or **savewpar** backup image.

Syntax

mkcd **-r** *directory* | **-d** *cd_device* | **-S** [**-m** *mksysb_image* | **-M** *mksysb_target* | **-s** *savevg_image* | **-v** *savevg_volume_group* | **-w** *savewpar_image* | **-W** *wparname*] [**-C** *cd_fs_dir*] [**-I** *cd_image_dir*] [**-V** *cdfs_volume_group*] [**-B**] [**-p** *pkg_source_dir*] [**-R** | **-S**] [**-i** *image.data*] [**-u** *bosinst.data*] [**-f** *wparspecificationfile*] [**-e**] [**-P**] [**-l** *package_list*] [**-L**] [**-b** *bundle_file*] [**-z** *custom_file*] [**-D**] [**-U**] [**-Y**] [**-n**] [**-a**] [**-A**] [**-c**] [**-Z**] [**-G**] [**-N**] [**-x** *file*] [**-T**]

Description

The **mkcd** command creates a system backup image (**mksysb**) to CD-Recordable (CD-R) or DVD-Recordable (DVD-R, DVD-RAM) from the system **rootvg** or from a previously created **mksysb** image. It creates a volume group backup image (**savevg**) to CD-R from a user-specified volume group or from a previously created **savevg** image. It also creates the backup image of a workload partition (**savewpar**) to CD or DVD from a user-specified workload partition or from a previously created **savewpar** image.

Note: If the system has a **multibos** environment where both instances are mounted, the only way to restore the backup is by using the **alt_disk_mksysb** command.

For DVD media, system backups that are made with the **mkcd** command have a limitation in that they expect the media to be 4.7 GB or larger per side. The **mkcd** command does not process the next volume until it writes over 4 GB on the current volume, thus the use of smaller media would result in corruption when you go beyond the capacity of the media.

When a bootable backup of a root volume group is created, the boot image reflects the currently running kernel. If the current kernel is the 64-bit kernel, the backup boot image is also 64 bit, and it boots 64-bit systems only. If the current kernel is a 32-bit kernel, the backup boot image is 32 bit, and it can boot both 32-bit and 64-bit systems.

With the **mkcd** command, you can create bootable and non-bootable CDs in Rock Ridge (ISO9660) or UDF (Universal Disk Format) format.

See the **-L** flag for details about creating DVD-sized images. What applies to CDs also applies to DVDs, except where noted.

Note: The functionality that is required to create Rock Ridge format CD images and to write the CD image to the CD-R, DVD-R or DVD-RAM device is not part of the **mkcd** command. You must supply additional code to **mkcd** command to do these tasks. The code is called by using shell scripts and then linked to **/usr/sbin/mkrr_fs** (for creating the Rock Ridge format image) and **/usr/sbin/burn_cd** (for writing to the CD-R device). Both links are called from the **mkcd** command.

Some sample shell scripts are included for different vendor-specific routines. You can find these scripts in **/usr/samples/oem_cdwriters**.

If you do not supply any file systems or directories as command parameters, the **mkcd** command creates the necessary file systems and removes them when the command finishes executing. File systems that you supply are checked for adequate space and write access.

Note:

1. While the **mkcd** command is running, ensure that system activity is minimal.
2. If the **mkcd** command creates file systems in the backup volume group, they are excluded from the backup.

If you want to create multi-volume CDs because the volume group image does not fit on one CD, the **mkcd** command gives instructions for CD replacement and removal until all the volumes have been created.

Flags

Item	Description
-a	Does not back up extended attributes or NFS4 ACLs.
-A	Backs up DMAPI file system files.

Item	Description
-b <i>bundle_file</i>	Gives the full path name of the file that contains a list of filesets to be installed after the mksysb is restored. This file is copied to ./usr/sys/inst.data/user_bundles/bundle_file in the CD file system and also copied to RAM in case the CD is unmounted. The file would be listed as BUNDLES=../usr/sys/inst.data/user_bundles/bundle_file in the bosinst.data file.
-B	Prevents the mkcd command from adding boot images (non-bootable CD) to the CD. Use this flag if you want to create a mksysb CD that you will not boot. Before you install the non-bootable mksysb CD, you must boot a same level (V.R.M.) product CD. The mkcd command defaults to creating a bootable CD for the machine type of the source system. See Notes for details.
-c	Does not compress or pack files as they are backed up.
-C <i>cd_fs_dir</i>	<p>Specifies the file system that is used to create the CD file system structure, which must have at least 645 MB of available disk space (up to 4.38 GB for DVD sized images). The CD image consumes only as much room as is necessary to contain all the data on the CD.</p> <p>If you do not specify the -C flag and the /mkcd/cd_fs directory exists, the mkcd command uses that directory. If you do not specify the -C flag and the /mkcd/cd_fs directory does not exist, the mkcd command creates the file system /mkcd/cd_fs and removes it when the command finishes running. The command creates the file system in the volume group that is indicated with the -V flag, or rootvg if that flag is not used. Each time that you invoke the mkcd command, a unique subdirectory (that uses the process id) is created under the /mkcd/cd_fs directory, or in the directory that is specified with the -C flag.</p> <p>Note: If performing DVD sized backups, the file systems must be <i>large file enabled</i>. This type of backup also requires setting the file ulimit size to <i>unlimited</i>.</p>
-d <i>cd_device</i>	Indicates the CD-R, DVD-R, or DVD-RAM device (/dev/cd1 , for instance). This flag is required unless you use the -S flag.
-D	Turns on the debug output information feature. The default is no debug output.
-e	Excludes the files and directories from the backup image that is listed in the /etc/exclude.volume_group file. You cannot use this flag with the -m or -s flags.
-f <i>wparspecificationfile</i>	Specifies the user-supplied WPAR specification file. This specification file of workload partition takes precedence over the wpar.spec file in the savewpar image. If you do not use the -f flag, the mkcd command restores the wpar.spec file from the specified savewpar image, or generates a new wpar.spec file during the creation of savewpar .

Item	Description
-i <i>image.data</i>	<p>Specifies the user-supplied <i>image.data</i> file. This data file takes precedence over the image.data file in the mksysb image. If you do not give the -i flag, then mkcd restores the image.data from the specified mksysb image, or generates a new image.data file during the creation of mksysb.</p> <p>Note: The -i flag cannot be used to specify a user-supplied <i>vgname.data</i> file for use with a savevg image.</p>
-I <i>cd_image_dir</i>	<p>Specifies the directory or file system where the final CD images are stored before they are written to the CD-R, DVD-R, or DVD-RAM device. If this flag is not used, mkcd uses the /mkcd/cd_images directory if it already exists. If not, the command creates the /mkcd/cd_images file system in the volume group that is given with the -V flag, or in rootvg if that flag is not used.</p> <p>If the mkcd command creates the file system, it is removed upon command completion, unless either the -R or -S flag is used. If the -R or -S flag is used, consideration must be made for adequate file system, directory, or disk space, especially when you create multi-volume CDs. The CD image consumes only as much room as is necessary to contain all the data on the CD.</p> <p>Note: If performing DVD sized backups, the file systems must be <i>large file enabled</i>. This type of backup also requires setting the file ulimit size to <i>unlimited</i>.</p>
-l <i>package_list</i>	<p>Specifies the file that contains a list of additional packages you want copied to the ./usr/lpp/inst.images directory of the CD file system. The images are copied from the location that is named with the -p flag. If you use the -l flag, you must also use the -p flag.</p>
-L	<p>Creates final CD images that are DVD sized (up to 4.38 GB).</p>
-m <i>mksysb_image</i>	<p>Specifies a previously created mksysb image. If you do not specify the -m flag, the mkcd command calls mksysb. For more information about where the mksysb image is placed, see the -M flag.</p>
-M <i>mksysb_target</i>	<p>States the directory or file system where the mksysb or savevg image is stored if a previously created backup is not given with the -m or -s flags. If the -M flag is not used and a mksysb or savevg image is not provided, the mkcd command verifies that /mkcd/mksysb_image exists. If the directory does not exist, then the mkcd command creates a separate file system, /mkcd/mksysb_image, where the mksysb or savevg images are temporarily stored. The command creates the file system in the volume group that is given with the -V flag, or in rootvg if that flag is not used.</p> <p>Note: If performing DVD sized backups, the file systems must be <i>large file enabled</i>. This also requires setting the file ulimit size to <i>unlimited</i>.</p>

Item	Description
-n	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 flag does not back up user data files. This backup can be used to create a user volume group without restoring user data files. This action cannot be done to rootvg.
-N	Includes file systems that belong to a workload partition (WPAR) in the defined state in the system backup. Note: To be included in the backup, all file systems that belong to a WPAR in the defined state must be in the rootvg volume group.
-p <i>pkg_source_dir</i>	Names the directory or device that contains device and kernel package images. The device must be a CD device (for example, /dev/cd0). If you use the same CD-R, DVD-R, or DVD-RAM device that you gave with the -d flag, the product CD media must be inserted into the CD-R drive first. The mkcd command then prompts you to insert the writeable CD before the actual CD creation.
-P	Creates physical partition mapping during the mksysb or savevg creation. You cannot use this flag with the -m or -s flags.
-r <i>directory</i>	Indicates existing directory structure to burn onto a CD or DVD. This flag makes a CD image that is a copy of the specified directory structure.
-R	Prevents the mkcd command from removing the final CD images. mkcd defaults by removing everything that it creates when it finishes running. The -R flag allows multiple CD image sets to be stored, or for CD creation (burn) to occur on another system. If multiple volumes are needed, the final images are uniquely named by using the process ID and volume suffixes.
-s <i>savevg_image</i>	Indicates a previously created savevg image. All savevg backup images are nonbootable. See Notes for details.
-S	Stops the mkcd command before it writes to the CD-R, DVD-R, or DVD-RAM without removing the final CD images. The -S flag allows multiple CD sets to be created, or for CDs to be created on another system. The images remain in the directory marked by the -I flag, or in the /mkcd/cd_images directory if the -I flag is not used. If multiple volumes are required, the final images are uniquely named by using the process ID and volume suffixes.

Item	Description
-T	<p>Creates backup by using snapshots. This command applies only to JFS2 file systems.</p> <p>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 back up 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 thus, 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.</p>
-u <i>bosinst.data</i>	<p>Specifies the user-supplied <i>bosinst.data</i> file. This data file takes precedence over the bosinst.data file in the mksysb image. If you do not give the -u flag, then the mkcd command restores bosinst.data from the specified mksysb image, or generates a new bosinst.data file during the creation of mksysb.</p>
-U	<p>Creates a UDF (Universal Disk Format) file system on DVD-RAM media. It does not require the amount of free space that is needed to create Rock Ridge format backups. It does not need the /mkcd/cd_fs and /mkcd/cd_images file systems. Therefore, the only temporary disk space it needs is to create the backup image that will be copied to the media. This means that the -I and -C flags do not apply to the -U flag. Because the backup is copied to the media, images cannot be created and burned later or on another system. So, the -R flag and -S flag do not apply when you use the -U flag. You must specify a device to write to with the -d flag. The -U flag does not use the /usr/sbin/mkrr_fs or /usr/sbin/burn_cd file systems.</p>
-v <i>savevg_volume_group</i>	<p>Denotes the volume group to be backed up using the savevg command. All savevg backup images are nonbootable. See Notes for details. For more information about where the savevg image is placed, see the -M flag.</p>
-V <i>cdfs_volume_group</i>	<p>Indicates the volume group that is used when you create the file systems needed for the mkcd command. If the -V flag is not given and a file system is needed but not there (because it was not supplied with other flags), then rootvg is the default volume group for creating the file systems. If the mkcd command creates the file systems in the backup volume group, those file systems are not included as part of the backup image. mkcd-created file systems are removed upon completion of the command.</p>
-w <i>savewpar_image</i>	<p>Indicates a previously created savewpar image.</p>
-W <i>wparname</i>	<p>Denotes the workload partition to be backed up using the savewpar command.</p>
-Y	<p>Accepts licenses.</p>

Item	Description
-z <i>custom_file</i>	States the full path name of the file to be copied to the root directory of the CD file system. This file could be a customization script that is specified in the bosinst.data file, such as CUSTOMIZATION_FILE=filename. For example: If the file my_script is in /tmp on the machine where the mkcd command is running, then enter -z/tmp/my_script and specify CUSTOMIZATION_FILE=my_script. The code copies the script to the root directory of the RAM file system before it runs.
-Z	Specifies that the Encrypted file system (EFS) information for all the files, directories, and file systems is not backed up.
-G	Excludes WPAR file systems from the system backup. The flag is not valid with -N flag.
-x <i>file</i>	Excludes the file systems that are listed in the file from the system backup. File system mount points must be listed one per line.

Note: Use care when you exclude file systems as a resulting backup can be unusable for system restoration.

Note:

1. If you are creating a non-bootable CD (by using the **-B** flag), you cannot use the **-p** or **-l** flags.
2. If you are creating a non-bootable CD with a **savevg** image (by using the **-s** or **-v** flags), you cannot use the **-p**, **-l**, **-u**, **-i**, **-z**, or **-b** flags.

Examples

1. To generate a bootable system backup to the CD-R device named `/dev/cd1`, enter the following command:

```
mkcd -d /dev/cd1
```

2. To generate a system backup to the DVD-R or DVD-RAM device named `/dev/cd1`, enter the following command:

```
mkcd -d /dev/cd1 -L
```

3. To generate a non-bootable volume group backup of the volume group `myvg` to `/dev/cd1`, enter the following command:

```
mkcd -d /dev/cd1 -v myv
```

Note: All **savevg** backup images are non-bootable.

4. To generate a non-bootable backup of the workload partition `mywpar` to **`/dev/cd1`**, enter the following command:

```
mkcd -d /dev/cd1 -W mywpar
```

Note: All **savewpar** backup images are not bootable.

5. To generate a non-bootable backup of the workload partition `mywpar` to **`/dev/cd1`** from the previously generated **savewpar** image **`/wparbackups/mywpar.bff`**, enter the following command:

```
mkcd -d /dev/cd1 -w /wparbackups/mywpar.bff
```

6. To create a CD or DVD that duplicates an existing directory structure such as:

```
/mycd/a  
/mycd/b/d  
/mycd/c/f/g
```

enter the following command:

```
mkcd -r /mycd -d /dev/cd1
```

After you mount with `mount -o ro /dev/cd1 /mnt, cd to /mnt; a find . -print command displays:`

```
./a  
./b  
./b/d  
./c  
./c/f  
./c/f/g
```

Files

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

mkcfsmnt Command

Purpose

Mounts a CacheFS directory.

Syntax

```
mkcfsmnt -d PathName -t { nfs | cdrom } [ -h RemoteHost ] [ -p { RemoteDirectory | LocalDeviceName } ]  
[ -c CacheDirectory ] [ -o MountOptions ] [ -b BackingFileSystem ] [ -I | -B | -N ]
```

Description

The **mkcfsmnt** command constructs an entry that will be appended to the `/etc/filesystems` file, thus making a file system available for use as a cache file system. If the mount is to be permanent, this entry will remain. If the mount is temporary, the flags will be used directly for the **mount** command. CacheFS file systems are used to cache accesses to backing file systems. Backing file systems are generally NFS mounts.

Flags

Item	Description
<code>-d</code> <i>PathName</i>	Specifies the mount point for the cache directory.
<code>-t</code>	Selects file systems to be cached. nfs Specifies that the CacheFS file system is backed by an NFS mount. cdrom Specifies that the CacheFS file system is backed by a CDROM file system. (Currently not supported.)

Item	Description
-h <i>RemoteHost</i>	Specifies the NFS server that is exporting the directory.
-p <i>RemoteDirectory</i>	Specifies the directory that is mounted on the path name specified. This is commonly a remote file system that will be mounted via NFS or a local device name in the case of CDROM (Currently not supported.)
-c <i>CacheDirectory</i>	Specifies the location of the CacheFS file system. This must have been previously created by execution of the cfsadmin command.
-d <i>RemoteDirectory</i>	Specifies the directory that is mounted on the path name specified.
-o <i>MountOptions</i>	Specifies a comma-separated string of mount options that are dependent on the backing file system type. For instance, if it is NFS, the options would be those typically specified by the -o Options string to mount. See the mount command documentation for the acceptable values.
-b <i>BackingFileSystem</i>	Specifies a backing file system if it is already mounted. If this is not specified, then the command will do the mount itself on a temporary mount point. If this is not specified, then RemoteHost and RemoteDirectory must be specified.
-I	Causes an entry to be added to the /etc/filesystems file. The directory is not mounted.
-B	Adds an entry to the /etc/filesystems file and attempts to mount the file system. This flag is the default.
-N	Mounts the directory with the options specified, but does not modify the /etc/filesystems file.

Examples

To specify a CacheFS mount, type:

```
/usr/sbin/mkcfsmnt -t nfs -d /usr/share/man -p /usr/share/man -h host1 -c
/cache/cache1 -o ro, intr -N
```

In this example, the **mkcfsmnt** command caches the remote directory **/usr/share/man** that resides on **host1** on the local **/usr/share/man** directory. The cache is kept in **/cache/cache1**, which was created with the **cfsadmin** command. CacheFS takes care of doing the NFS backing mount, because the **-b** flag has not been specified.

```
/usr/sbin/mkcfsmnt -t nfs -d /usr/share/man -p /usr/share/man -h host1 -c /cache/cache1
-b /backs/man -o ro, intr -N
```

In this example, the **mkcfsmnt** command caches the remote directory **/usr/share/man** residing on **host1** on the local **/usr/share/man** directory. The cache is kept in **/cache/cache1**, which was created with the **cfsadmin** command. The backing file system has already been mounted on **/backs/man**.

Files

Item	Description
<u>/etc/filesystems</u>	Lists the remote file systems to be mounted during the system restart.

mkcifscred Command

Purpose

Adds CIFS credentials to the `/etc/cifs_fs/cifscred` file to allow future mounting of CIFS shares with stored credentials.

Syntax

```
mkcifscred -h RemoteHost -u user [-p password]
```

Description

The `mkcifscred` command takes a server and user name as input, and prompts for a password. The password is encrypted, and the credentials are stored in the `cifscred` file. If the password is not passed in with the `-p` option when mounting to a CIFS server, the credentials are either retrieved from the `cifscred` file, or, if the credentials do not exist in `cifscred`, the password is prompted for and read in as hidden input.

The credentials are stored as a `server/user/password` set. Multiple sets of credentials for the same server are permitted with different user names. Multiple sets with the same user name on different servers are also permitted.

Flags

Item	Description
<code>-h <i>RemoteHost</i></code>	Specifies the name of the remote host (CIFS server). This can be provided as a host name, an IP address, or a fully qualified domain name.
<code>-p <i>password</i></code>	Specifies the password for a particular user on a particular remote host.
<code>-u <i>user</i></code>	Specifies the user name whose credentials are being defined for access to the given remote host.

Exit Status

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

Examples

1. To add credentials for `user1` to mount on `server1`, enter:

```
mkcifscred -h server1 -u user1
```

Location

`/usr/sbin/mkcifscred`

Files

Item	Description
<code>/etc/cifs_fs/cifscred</code>	Stores the CIFS credentials.

mkcifsmt Command

Purpose

Adds a CIFS mount to the `/etc/filesystems` file and performs the mount.

Syntax

```
mkcifsmt -f MountPoint -d RemoteShare -h RemoteHost -c user [-p password] [-m MountTypeName]  
[-A|-a] [-I|-B|-N] [-t {rw|ro}] [-u uid] [-g gid] [-x fmode] [-w wrkgrp]
```

Description

The `mkcifsmt` command constructs a CIFS entry that is appended to the `/etc/filesystems` file. It then attempts to mount the CIFS file system. Its options are parsed and prepared to be passed into the `cifs` command, which actually adds the CIFS entry to `/etc/filesystems`.

Flags

Item	Description
-a	Specifies that the <code>/etc/filesystems</code> entry for this file system should not be automatically mounted at system restart. This is the default.
-A	Specifies that the <code>/etc/filesystems</code> entry for this file system should be automatically mounted at system restart.
-B	Specifies that the entry should be added to the <code>/etc/filesystems</code> and that it should be mounted at system restart.
-c <i>user</i>	Specifies user name used to gain access to the CIFS share.
-d <i>RemoteShare</i>	Specifies the share name on the CIFS server that should be mounted.
-f <i>MountPoint</i>	Specifies the path name over which the CIFS share should be mounted.
-g <i>gid</i>	Specifies the GID that is assigned to files in the mount. The default is 0.
-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.
-I	Specifies that the entry should be added to the <code>/etc/filesystems</code> file, but should not be mounted.
-m <i>MountTypeName</i>	Defines the mount type that will be added to the <code>/etc/filesystems</code> file, which allows for mounting all file systems of a specific type using the <code>-t</code> option of the mount command. By default, no type value will be added to <code>/etc/filesystems</code> .
-N	Mounts the CIFS share with the options specified, but does not modify the <code>/etc/filesystems</code> file.

Item	Description
-p <i>password</i>	Specifies the password used to grant access to the specific user on the specific server. The specific credentials (server/user/password) are added to the <code>cifscred</code> file (the password will be encrypted). If the <code>-p</code> option is not specified, and the credentials do not already exist in the <code>cifscred</code> file, the command line prompts the user to provide the password, and the credentials will be added to the <code>cifscred</code> file. If the server/user credentials already exist in the <code>cifscred</code> file, this option is ignored, and the existing credentials are used for mounting.
-t {rw ro}	Specifies whether file system should be mounted as read-only. The default is read-write (rw).
-u <i>uid</i>	Specifies the UID that is assigned to files in the mount. The default is 0.
-x <i>fmode</i>	Specifies the owner, group, and other permission bits assigned to files in the mount. The default is 755.
-w <i>wrkgrp</i>	Specifies the domain that should be used to authenticate the user during mount. If this option is not used, authentication is handled locally by the CIFS server.

Exit Status

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

Security

You must have root authority to run this command.

Examples

1. To add a mount over `/mnt` to `share1` on `server1`, and then authenticate as `user1`, enter:

```
mkcifsmt -f /mnt -d share1 -h server1 -c user1
```

Location

`/usr/sbin/mkcifsmt`

Files

Item	Description
<code>/etc/cifs_fs/cifscred</code>	Stores the CIFS credentials.
<code>/etc/filesystems</code>	Stores the CIFS entry.

mkcimreg Command

Purpose

Registers Common Information Model (CIM) classes and Common Manageability Programming Interface (CMPI) providers with RMC.

Syntax

To register a class:

```
mkcimreg [-I include_directory...] [-f] [-h] definition_file...
```

To register a provider:

```
mkcimreg [-I include_directory...] [-p provider_directory] [-h] registration_file...
```

To compile the CIM schema:

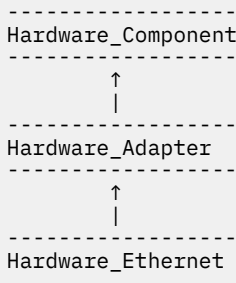
```
mkcimreg [-I include_directory...] -b schema_path [-h]
```

Description

The `mkcimreg` command registers Common Information Model (CIM) classes and Common Manageability Programming Interface (CMPI) providers with the resource monitoring and control (RMC) subsystem. You can specify one or more class definition files or provider registration files with this command. Use the `-I` flag to add directories to the search path. The output from `mkcimreg` includes the names of the files that the CIM resource manager needs for working with CIM classes.

Registering classes

If you upgrade a class using the `-f` flag (that is, if the class definition has changed somehow), you must re-register all classes that are subclasses of the upgraded class so that the changes introduced into the new class propagate to its subclasses. This must be done in "descending" order, because changes propagate from parent to child. The hierarchy is:



If, for example, `Hardware_Component` is upgraded using `mkcimreg -f`, `Hardware_Adapter` and then `Hardware_Ethernet` must both be registered afterward, in that order.

After you register any classes:

You must restart RMC.

Restarting RMC

As the final step in the CIM class registration process, the RMC subsystem must be restarted. The sequence of commands to run follows:

1. To shut down the RMC subsystem, enter:

```
/opt/rsct/bin/rmcctl -k
```

When you shut down RMC:

Any RMC-dependent resource monitoring that is in place at the time of shutdown is deactivated. Environments that rely on RMC or any of its resource managers for high availability or other critical system functions may become temporarily disabled.

2. Wait until the following command lists the status of `ctrmc` as "inoperative":

```
lssrc -s ctrmc
```

3. Shut down the CIM resource manager and confirm it has been stopped:

```
stopsrc -s IBM.CIMRM  
lssrc -s IBM.CIMRM
```

4. To restart the RMC subsystem, enter:

```
/opt/rsct/bin/rmcctrl -A
```

Registering providers

The `-p` flag indicates that the registration file on the command line contains provider registration information. The provider library's directory is expected as this flag's parameter. Provider library names follow the CMPI/Pegasus convention of appending `lib` to the beginning of the `ProviderName` property. For example, the provider with the property `ProviderName=Linux_Processor` is searched for in the `ProviderDirectory` under the name `libLinux_Processor.so`. Auxiliary libraries required by providers that are not explicitly declared in the registration file must be either in the directory supplied on the command line, or in a standard system directory such as `/usr/lib` or `/lib`.

Compiling a schema

Version 2.9 of the CIM schema is shipped with the CIM resource manager. Use the `-b` flag if you want to upgrade to a higher version. The schema file (`CIM_Schemaversion.mof`) must be passed as the parameter to this flag. This file contains the entire CIM schema, usually in the form of a series of `#include` statements that bring in other schema MOF files.

After a CIM schema is compiled with the `-b` flag, `mkcimreg` will not need further access to the schema managed object format (MOF) files. User classes that are registered by `mkcimreg` against previous versions of the CIM schema need to be re-registered, so changes from the new version of the schema are reflected in any derived classes.

Flags

-I include_directory...

Specifies one or more additional directories to be searched.

-f

Overwrites any existing class registration data with the definitions that are provided in the class definition files.

-p provider_directory

Specifies a path to the provider library.

-b schema_path

Compiles the CIM schema file.

-h

Writes the command's usage statement to standard output.

Parameters

definition_file...

Specifies one or more class definition files.

registration_file...

Specifies one or more provider registration files.

Security

This command requires root authority.

Exit Status

0

The command has run successfully.

1

An internal command error occurred.

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

A class registration error occurred.

Restrictions

You cannot register a class that derives from a class that has not yet been registered.

Implementation Specifics

This command is part of the `rsct.exp.cimim` fileset, in the `rsct.exp` package on the AIX Expansion Pack.

Standard Output

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

Standard Error

When the `-T` flag is specified, this command's trace messages are written to standard error.

Examples

1. To register the `Linux_ComputerSystem` CIM class if the class definition file is located in the `$CIMDEFS` directory, enter:

```
mkcimreg $CIMDEFS/Linux_ComputerSystem.mof
```

You must also register the CMPI provider for this class.

2. To register a CMPI provider when the registration file is located in the `$CIMDEFS` directory and the provider library is in the `$CMPIHOME` directory, enter:

```
mkcimreg -p $CMPIHOME $CIMDEFS/Linux_ComputerSystemRegistration.mof
```

3. To compile Version 2.12 of the CIM schema, enter:

```
mkcimreg -I $SCHEMA_DIR -b CIM_Schema2.12.mof
```

`$SCHEMA_DIR`, which indicates a search path for schema MOF files, is not required, but could help `mkcimreg` find the required MOF files if they are not in the current working directory from which the command is run.

Location

`/opt/rsct/bin/mkcimreg`

mkclass Command

Purpose

Create a Workload Management class.

Syntax

```
mkclass [ -a Attribute=Value ... ] [ -c | -m | -b | -v | -C | -B | -P | -T | -V | -L | -A KeyWord=Value ] [ -d Config_Dir ] [ -S SuperClass ] Name
```

Description

The **mkclass** command creates a superclass or a subclass identified by the *Name* parameter. The class must not already exist. The *Name* parameter can contain only uppercase and lowercase letters, numbers, and underscores. The name is in the format *supername* or *subname* (with the **-S** *supername* flag) or *supername.subname*. The *supername* and *subname* parameters are each limited to 16 characters in length. The names **Default**, **System**, and **Shared** are reserved. They refer to predefined classes. Any *Attribute=Value* or *KeyWord=Value* argument initializes the specified attribute or resource limit. See “Attributes” on page 538 for more information. To set the process total limits (the limits that apply to each process of the class), use one or more of the options **-C** (totalCPU), **-B** (totalDiskIO), **-A** (totalConnectTime), or **-v** (totalVirtualMemoryLimit), with the keyword value of `hardmax`. To set the class total limits (the limits that apply to the whole class), use one or more of the options **-P** (totalProcesses), **-T** (totalThreads), **-L** (totalLogins), or **-V** (totalVirtualMemoryLimit) with the keyword value of `hardmax`. To reset any total limit, use **-** for *Value*. Process, class, or both total limits may be disabled when starting or updating the WLM (see **wlmcntrl** command).

Normally, **mkclass** adds the class and its attributes in the relevant WLM property files, and the modifications is 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 created only in the WLM in-core data structures, and no property file is updated, making the new class temporary (the change is lost if WLM is stopped and restarted or the system is rebooted).

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.

Attributes

The following attributes can be changed:

Class properties:

Item	Description
tier	Specifies the tier value. The tier value for a class is the position of the class in the hierarchy of resource limitation desirability for all classes. A class with a lower tier value is more favored. The tier value ranges from 0 through 9 (the default is 0).

Item	Description
inheritance	If the inheritance attribute is set to yes , the children of processes in this class remain in the class upon exec regardless of the automatic assignment rules in effect. If the inheritance attribute is set to no , the assignment rules apply normally. The default if not specified is no .
localshm	Indicates whether memory segments that are accessed by processes in different classes remain local to the class they were initially assigned to or if they go to the Shared class. You can specify a value of Yes or No . If not specified, the default is No .
authuser	Specifies the user name of the user who is allowed to assign processes to this class. The default when the attribute is not specified is root .
authgroup	Specifies the group name of the group of users that is allowed to assign processes to this class. There is no default value.
rset	Specifies the name of a resource set that the processes in the class have access to. By default, the class has access to all resources on the system.
vmenforce	Specifies whether all processes or only the offending processes in the class need to be terminated when the class hits the maximum VM limit. You can specify the value of class or proc . The default value is proc .
delshm	Specifies whether the shared segments will be deleted when the last process referencing them ends because virtual memory is exceeded. You can specify the value of yes or no . The default value is no .
adminuser	Specifies the user name of the user who is allowed to administer the subclasses of this superclass. This attribute is valid only for superclasses. The default, when the attribute is not specified, is a null string, and in this case, only root users can administer the subclasses. Note: If the adminuser or admingroup attribute is changed for a superclass that belongs to the running configuration (or to a configuration of the running set), a global WLM update should be performed to reflect these changes to the in-core configuration, elsewhere, updates that are restricted to superclass by such a user might fail due to lack of authority.
admingroup	Specifies the group name of the group of users that is allowed to administer the subclasses of this superclass. This attribute is valid only for superclasses. The default value, when the attribute is not specified, is a null string, meaning that no group can administer the subclasses. Note: If the adminuser or admingroup attribute is changed for a superclass that belongs to the running configuration (or to a configuration of the running set), a global WLM update should be performed to reflect these changes to the in-core configuration, elsewhere, updates that are restricted to superclass by such a user might fail due to lack of authority.

Class limits and shares for CPU, memory, or disk I/O resource:

Item	Description
min	Specifies the minimum percentage of the resource that must be made available when requested, expressed as a percentage of the total resource available in the system. Possible values range from 0 through 100 (the default is 0).

Item	Description
shares	Specifies the maximum ratio of the resource that can be made available if there is contention. This parameter is expressed in shares of the total resource available in the system. The actual ratio of the resource is dynamically computed, proportionally to the shares of all active classes. If a class has no running process, its shares are excluded from the computation. The shares are arbitrary numbers ranging from 1 through 65535. If shares is specified as a hyphen (-), the class is always considered on target and its utilization for this resource is not regulated by WLM, but the minimum and maximum limits if any still apply. This is the default if the shares for a resource are not specified.
softmax	Specifies the maximum percentage of the resource that can be made available, when there is contention. Possible values range from 1 through 100 (the default is 100). A class can exceed its soft maximum for a given resource if there is no contention on the resource.
hardmax	Specifies the maximum percentage of the resource that can be made available, even if there is no contention. Possible values range from 1 through 100 (the default is 100). Specifying a value different from the default value of 100 for memory can result in some memory pages remaining unused, while some processes in the class use more.
max	Specifies the maximum percentage of the resource that can be made available, even if there is no contention. Possible values range from 1 through 100 (the default is 100). Specifying a value different from the default value of 100 for memory can result in some memory pages remaining unused, while some processes in the class use more.

Note: The default values for a class can be read using the **lsclass -D** command and can be changed by manually editing the property files **classes**, **shares**, or **limits** to add a default stanza. For more information about these files, see the *Files Reference*.

Class description:

Item	Description
description	The class description text can be composed of any ASCII character, except colons (:) and commas (,).

Note: This command is not supported when executed within a workload partition.

Flags

Item	Description
-A hardmax=Value	Sets the maximum amount of time a login session in the class can stay active. Value is specified as an integer, possibly appending the unit (s for seconds, m for minutes, h for hours, d for days, and w for weeks, default is seconds). As a user approaches this connection time limit, WLM will send a warning message to the session terminal. When the limit is reached, the user will be notified and the session leader will be sent the SIGTERM signal, and after a short grace period, the session will be terminated (SIGKILL).
-B hardmax=Value	Sets the total amount of disk I/Os allowed for each process in the class. Value is specified as an integer, possibly appending the unit (KB for kilobytes, MB for megabytes, TB for terabytes, PB for petabytes, and EB for exabytes, default is kilobytes). After a process has used this amount of disk I/Os, the process will be sent the SIGTERM signal, and after a grace period, it will be killed (SIGKILL).
-C hardmax=Value	Sets the total amount of CPU time allowed for each process in the class. Value is specified as an integer, possibly appending the unit (s for seconds, m for minutes, h for hours, d for days, and w for weeks, default is seconds). After a process has used this amount of time, the process will be sent the SIGTERM signal, and after a grace period, it will be killed (SIGKILL).

Item	Description
-d <i>Config_Dir</i>	Use <code>/etc/wlm/Config_Dir</code> as an alternate directory for the properties files. When this flag is not used, mkclass uses the configuration files in the directory pointed to by <code>/etc/wlm/current</code> . If an empty string is passed as the configuration name (<code>-d ""</code>) the new class is created only in the WLM in-core data structures and no configuration file is modified.
-L hardmax=Value	Sets the total number of login sessions simultaneously available in the class. If a user tries to log onto the system and the login shell would end up in a class that has reached the total logins limit, the login operation will fail.
-P hardmax=Value	Sets the maximum number of processes allowed in the class. If an operation would result in a new process entering the class when the class has this many processes in it, the operation will fail.
-S SuperClass	Specifies the name of the superclass when creating a subclass. There are two ways of creating the subclass Sub of superclass Super : <ol style="list-style-type: none"> 1. Specify the full name of the subclass as Super.Sub for <i>Name</i> and not use -S 2. Specify the -S flag to give the superclass name and use the short name for the subclass: <pre style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;">mkclass options -S Super Sub</pre>
-T hardmax=Value	Sets the maximum number of threads allowed in the class. If an operation would result in a new thread entering the class when the class has this many processes in it, the operation will fail. The total thread limit must be at least as large as the total process limit for a class. If a class has a total thread limit but no total process limit specified, the total process limit will be set to the total thread limit.
-v hardmax=Value	Specifies the virtual memory limit allowed per process in the specified class. The maximum amount of virtual memory allowed per process is $(2^{31})-1$ for 32-bit kernels and $(2^{63})-1$ for 64-bit kernels.
-V hardmax=Value	Specifies the virtual memory allowed for the specified class. The maximum amount of virtual memory allowed per process is $(2^{31})-1$ for 32-bit kernels and $(2^{63})-1$ for 64-bit kernels.

Security

Access control: Only the root user can create 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 create a subclass of this superclass.

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 each class.
shares	Contains the resource shares attributed to each class.

mkclient Command

Purpose

Uncomments the entry in the `/etc/rc.nfs` file for the **ypbind** daemon and starts the **ypbind** daemon to configure a client.

Syntax

```
/usr/sbin/mkclient [ -I | -B | -N ] [ -S server]
```

Description

The **mkclient** command uncomments the entry to the `/etc/rc.nfs` file to start the **ypbind** daemon to configure a client. The **mkclient** command starts the **ypbind** daemon by using the appropriate System Resource Controller (SRC) command.

You can use the System Management Interface Tool (SMIT) **smit mkclient** fast path to run this command.

Flags

Item	Description
------	-------------

- | | |
|-----------|--|
| -I | Uncomments the entry for starting the ypbind daemon to the <code>/etc/rc.nfs</code> file. This entry causes the ypbind daemon to start during the next system restart. |
| -B | Uncomments the entry to the <code>/etc/rc.nfs</code> file and starts the ypbind daemon. This flag is the default. |
| -N | Causes the startsrc command to start the ypbind daemon. This flag does not affect the <code>/etc/rc.nfs</code> file. |
| -S | Specifies which NIS <i>server</i> to use instead of broadcasting for one. This option must be used when no NIS server exists on the networks directly connected to the client machine. |

Examples

To modify the `/etc/rc.nfs` file so that the **ypbind** daemon is started on the next system restart, enter:

```
mkclient -I
```

Files

Item	Description
<code>/var/yp/domainname</code> directory	Contains the NIS maps for the NIS domain.
<code>/etc/rc.nfs</code>	Contains the startup script for the NFS and NIS daemons.

mkcluster Command

Purpose

To create a single-site cluster.

Syntax

```
mkcluster -r reposdev [ -S sitename { [ cle_uid=UUID,cle_globid=id,cle_prio=prio ] } ] [ -m node { [ cle_ip=addr,cle_uid=UUID,cle_globid=id ] } [,...] [ -d shareddisk [,...] ] [ -n clustername ] [ -s multi_cast_addr ] [ -c capability [,...] ] [ -v ] [ -b backupdisk [,...] ] [ -p comdisk ]
```

Description

The **mkcluster** command creates a cluster. A cluster is a collection of nodes and disks.

Each node that is added to the cluster must have common storage area network (SAN) storage devices that are zoned appropriately. The SAN storage devices are used for the cluster repository disk and for any clustered shared disks.

A multicast address is used for cluster communications between the nodes in the cluster. If any network considerations need to be reviewed before creating a cluster, consult your cluster systems administrator.

Flags

Item	Description
-b <i>backupdisk</i> [,...]	Specifies a comma-separated list of SAN shared storage device such as <code>hdisk5</code> and <code>hdisk6</code> . These disks are used as the backup for the central repository of the cluster. When the central repository is inaccessible, the disk from the list is used as a replacement. These devices must be accessible from all nodes in the site.
-c <i>capability</i> [,...]	Specifies a comma-separated list of capabilities that the cluster requires upon creation. If no capabilities are specified, the mkcluster command allows for the possibility that some nodes have older AIX software that is not capable of supporting newer CAA capabilities. In that case, the cluster is created in such a way that it is compatible with nodes that run older AIX software. After the cluster is created, and it is determined that all nodes can support newer CAA capabilities, the cluster automatically allows those capabilities to be used. The -c flag merely allows the specified capabilities to be used without first determining that all nodes are able to support it. In some situations, it is necessary for newer CAA capabilities to be enabled immediately. For example, if one or more nodes do not have IPv4 connectivity to all of the other cluster nodes, then it is necessary to specify that IPv6 capability must be enabled during cluster creation, to allow those nodes that have only IPv6 connectivity to join the cluster. All nodes must be online and able to join the cluster, to determine that the cluster can support newer capabilities. If a cluster must be created while one or more of the nodes are powered off, then specifying that the capability is needed during cluster creation allows the capability to be used before all of the nodes are able to join the cluster. However, you must make sure that all of the nodes have an AIX software level that can support that capability. Otherwise, any nodes that have older AIX software that cannot support the capability are not allowed to join the cluster.

The **-c** flag supports the following capability keywords:

- **ipv6**: IPv6 connectivity is required because some nodes do not have IPv4 connectivity to the rest of the cluster, or IPv6 functionality is needed before all nodes are able to join the cluster.
- **site**: The cluster needs to allow one or more sites to be defined before all nodes are able to join the cluster.
- **auto_repos_replace**: The created cluster can process and maintain the backup repository disks.
- **4kdisk**: The cluster must be able to support a 4k-block disk as repository and backup repository disk.
- **>|pvm_wdog**: The created cluster can process the POWER® hypervisor (PHYP) virtual machine watchdog (PVM WD) tunable parameter. |<

Item	Description
-c <i>unicast</i> or <i>multicast</i>	<p>Specifies the type of communication mode that is used by CAA to transfer CAA heartbeats and other protocol messages.</p> <p>If the <i>unicast</i> option is specified, CAA uses unicasting to transfer the protocol messages.</p> <p>If the <i>multicast</i> option is specified, CAA uses multicasting to transfer the protocol messages.</p> <p>If no option is specified, CAA uses the default multicast communication mode.</p>
-p <i>comdisk</i>	<p>Specifies a SAN shared storage device such as <i>hdisk5</i> and <i>hdisk6</i>. These disks are used by the shared storage pool cluster for inter-node communication when the network is down.</p>
-r <i>reposdev</i>	<p>Specifies the name of the SAN shared storage device that is used as the central repository for the cluster configuration data, such as <i>hdisk10</i>. This device must be accessible from all gateway nodes in the site. It is required that this device is a minimum of 1 GB, and is backed up by a redundant and highly available SAN configuration.</p>
-S <i>sitename</i>	<p>Specifies the name of the local site. If not specified, a default site with the name LOCAL is created. Currently, a cluster can support only 2 sites. To create a second site, use the chcluster command.</p> <p>The following site information can be specified:</p> <ul style="list-style-type: none"> • <i>cle_uuid</i>: The site UUID, which is acknowledged as unique across the cluster. If not specified, the site UUID is automatically generated. • <i>cle_globid</i>: The short ID of site, which must be a unique unsigned number greater than zero. If not specified, the site short ID is automatically generated. <p>The following site attribute can be specified:</p> <ul style="list-style-type: none"> • <i>cle_prio</i>: The priority of a site. A lower value indicates a higher priority. <p>The priority is used in the context of synchronizing the repository metadata.</p> <p>If two sites split and the repository data becomes out of sync, then the data from the site with higher priority must be copied over to the site with lower priority.</p>
-m <i>node[,...]</i>	<p>Lists the comma-separated host names or IP addresses for nodes that are members of the cluster. The local host must be included in the list. If the -m flag is not used, the local host is implied, causing a one-node local cluster to be created.</p> <p>The following node information can be specified:</p> <ul style="list-style-type: none"> • <i>cle_uuid</i>: The node UUID, which is acknowledged as unique across the cluster. If not specified, the node UUID is automatically generated. • <i>cle_globid</i>: The short ID of node, which must be a unique unsigned number greater than zero. If not specified, the node short ID is automatically generated. <p>The following node attributes can be specified:</p> <ul style="list-style-type: none"> • <i>cle_ip</i>: The nodes gateway address (in case the cluster spans across multiple sites). Typically, this attribute is an address through which this node can be reached from an external node. This address can be specified in either IP version 4 or version 6 format.

Item	Description
-d <i>shareddisk[,...]</i>	Specifies a comma-separated list of shared storage area network (SAN) devices, such as <i>hdisk12</i> , <i>hdisk34</i> , to be incorporated into the cluster configuration. Specified devices must not be open when the mkcluster command is run.
-n <i>clustername</i>	Sets the name of the cluster that is being created. If no name is specified when you run the mkcluster command, a default of CL_hostname is used, where <i>hostname</i> is the name of the local host. You can retrieve the name of the local host by running the gethostname() function.
-s <i>multi_cast_addr</i>	Sets the multicast address of the cluster that is created. This address is used for internal communication within the cluster. Only a multicast address in IPv4 format is accepted. If an IPv6 multicast address is needed, it must be generated from the IPv4 address. If the -s flag is not specified when you first run the mkcluster command, the necessary multicast addresses are automatically generated.
-v	Specifies the verbose mode.

Examples

1. To create a cluster of one node and use the default values, enter the following command:

```
mkcluster -r hdisk
```

The output is a cluster that is name *CL_myhostname* with a single node in the cluster. The multicast address is automatically generated and no shared disks are created for this cluster. The

```
mkcluster -r hdisk1
```

repository is set up on *hdisk1* and this disk cannot be used by the node for any other purpose. The repository device is dedicated as the cluster repository disk.

2. To create a multinode cluster, enter the following command:

```
mkcluster -n mycluster -m nodeA,nodeB,nodeC
```

The output is a cluster of three nodes and uses the default values. The output also creates a cluster with the specified name and the multicast address is automatically created. Three disks are created as shared clustered disks for this cluster. The repository device is set up on *hdisk1*, and it cannot be used by any of the nodes for any other purpose. The repository device is now dedicated to being the cluster repository disk. A volume group of *cvg* is created for the cluster repository disk and these logical volumes are used exclusively by the clustering subsystem.

3. To create a cluster that is capable of IPv6 and sites, enter the following command:

```
mkcluster -n mycluster -m nodeA,nodeB,nodeC -r hdisk1 -c ipv6, site
```

This command creates a cluster of three nodes that are immediately capable of using IPv6 networks and having sites that are defined. The cluster is named *mycluster*, and the local site and multicast address are automatically created with default values. If any IPv6 networks are configured, they can be used for cluster communication. The repository device is set up on *hdisk1*, and it cannot be used by any of the nodes for any other purpose. The repository device is now dedicated to being a cluster repository disk.

4. To create a cluster with one site named *mysite*, enter the following command:

```
mkcluster -n mycluster -S mysite -m nodeA,nodeB,nodeC -r hdisk1 -d hdisk10,hdisk11,hdisk12
```

The output is a single-site cluster of three nodes that uses the default attribute values for all sites and all nodes. The cluster is named *mycluster*, and the local site is named *mysite*. The multicast address is automatically created. Three disks are created as shared clustered disks for the local site. The

repository device is set up on *hdisk1*, and it cannot be used by any of the nodes for any other purpose. The repository device is now dedicated to being a cluster repository disk.

5. To create a cluster with one site named *mysite*, specifying site and node information, enter the following command:

```
mkcluster -n mycluster -S mysite{cle_uuid=0551c722-92fe-11e1-97b0-1aae1ed14715,
cle_globid=5,cle_prio=2}
-m nodeA,nodeB,nodeC{cle_uuid=e4ad47bc-92fd-11e1-8486-1aae1ed14715}
-r hdisk1 -d hdisk10,hdisk11,hdisk12
```

The output is a single-site cluster of three nodes. Nodes *nodeA* and *nodeB* have automatically generated UUIDs, while *nodeC* has a UUID of *e4ad47bc-92fd-11e1-8486-1aae1ed14715*. The cluster is named *mycluster*, and the local site is named *mysite* and has a UUID of *0551c722-92fe-11e1-97b0-1aae1ed14715*, a short ID of 5, and a priority of 2. The multicast address is automatically created. Three disks are created as shared clustered disks for the local site. The repository device is set up on *hdisk1*, and it cannot be used by any of the nodes for any other purpose. The repository device is now dedicated to being a cluster repository disk.

6. To create a multinode unicast cluster with one site named *mycluster*, append **-c unicast** to the **mkcluster** command. The multicast cluster in example 2 can be made unicast by entering the following command:

```
mkcluster -r hdisk10 -m nodeA,nodeB,nodeC -n mycluster -r hdisk1
-d hdisk10,hdisk11,hdisk12 -c unicast
```

7. To create a cluster that uses backup disks that are added later, enter the following command:

```
mkcluster -n mycluster -n nodeA,nodeB -r hdisk1 -c auto_repos_replace
```

8. To create a cluster and populate the backup repository disk list, enter the following command:

```
mkcluster -n mycluster -n nodeA,nodeB -r hdisk1 -b hdisk5,hdisk6
```

Messages

CLUST_LVL_NO_4K

"1035-346 %1\$s: The current effective cluster level does not support 4k-block disks.\n"

DISK_INFO_ERR

"1035-347 %1\$s: Unable to get disk information for %2\$s.\n"

mkcomg Command

Purpose

Creates a new communication group definition for a peer domain.

Syntax

```
mkcomg [-s sensitivity] [-p period] [-g grace] [-t priority] [-x b | r | br] [-N
UseForNodeMembership] [-e NIM_path] [-m NIM_parameters] [-M media_type] [-i {h |
n}:interface1[:node1],[interface2[:node2]]...] [-S {h | n}:"interface_selection_string"] [-6] [-h] [-TV]
communication_group
```

Description

The **mkcomg** command creates a new communication group definition for an online peer domain with the name specified by the *communication_group* parameter. 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. There can be more than one communication group in a peer domain.

The `mkcomg` command must be run on a node that is currently online in the peer domain where the communication group is to be defined. More than half of the nodes must be online to create a new communication group for the domain.

The `-e` and `-m` flags are used to set the network interface module (NIM) path and parameters. The NIM path is the path to the NIM that supports the adapter types used in the communication group. The NIM parameters are passed to NIM when it is started. If `-m` is not specified, the parameters predefined by topology services are used.

The communication group can be assigned to one or more interface resources. Use the `-i` flag to assign the communication group to a specific interface resource name. The interface resource can be limited to one on a particular node. An interface resource can also be specified using the `-S` flag and a selection string. This is used when specifying the interface resource name is not sufficient. The `-i` and `-S` flags cannot be used together. The `chcomg` command can also be used to assign a communication group to an interface resource.

Flags

-s sensitivity

Specifies the heartbeat sensitivity. This is the number of missed heartbeats that constitute a failure. The sensitivity value is an integer greater than or equal to 2. The default value is 4.

-p period

Specifies the amount of time between heartbeats. The period is specified in seconds and is significant to milliseconds. It can be specified as an integer or as a floating-point number.

-g grace

Specifies the grace period that is used when heartbeats are no longer received. When a heartbeat is missed, an Internet Control Message Protocol (ICMP) echo packet is sent to the failed node. If the echo is returned, the grace period is initiated.

The grace period is specified in seconds and is significant to milliseconds. It can be specified as an integer, a floating-point number, or one of these values:

0

Specifies that the grace period is disabled.

-1 | D

Specifies that the topology services subsystem controls the grace period. This is the default.

-t priority

Specifies the priority. This value indicates the importance of this communication group with respect to others. It is used to order the heartbeat rings. The lower the number means the higher the priority. The highest priority is 1. The default value is 1 for IP networks and 255 for RS232 networks.

-x b | r | br

Excludes controls for heartbeat mechanisms. This flag indicates that one or more controls for heartbeat mechanisms should not be used even if the underlying media support it. The following features can be excluded:

b

Specifies that the broadcast feature should not be used even if the underlying media support it. If `-x b` is not specified, the broadcast feature will be used if the underlying media support it.

r

Specifies that the source routing feature should not be used even if the underlying media support it. If `-x r` is not specified, the source routing feature will be used if the underlying media support it.

To exclude more than one control, specify the feature characters consecutively: `-x br`.

-N UseForNodeMembership

Specifies whether group services will use the communication group in calculating node membership. Sets the **UseForNodeMembership** persistent resource attribute for the communication group resource. Valid values are:

0

Indicates that, regardless of the results of liveness checks run on **NetworkInterface** resources that are members of this communication group, group services will not use those results in calculating whether the node owning the interfaces is online.

1

Indicates that group services will use the results of liveness checks run on the **NetworkInterface** resources in calculating the online state of their owning nodes.

-e NIM_path

Specifies the network interface module (NIM) path name. This character string specifies the path name to the NIM that supports the adapter types in the communication group.

-m NIM_parameters

Specifies the NIM start parameters. This character string is passed to the NIM when starting it.

-M media_type

Specifies the type of interfaces that make up *communication_group*. Valid values are:

0

Indicates that *communication_group* consists of interface resources other than IP or disk.

1

Indicates that *communication_group* consists of IPv4 or IPv6 interface resources.

If the **-M** flag is not specified, this is the default.

2

Indicates that *communication_group* consists of disk interface resources.

-i {h | n}:interface1[:node1] [,interface2[:node2]]...

Assigns *communication_group* to one or more heartbeat or network interface resources and, optionally, to the nodes where these resources can be found. Specify **-i h** for heartbeat interface resources or **-i n** for network interface resources.

By default, the **-i n** flag adds network interface resources that have IPv4 addresses to *communication_group*. If the **-6** flag is specified, the **-i n** flag will add network interface resources that have IPv6 addresses to *communication_group*.

If **-i** is specified, **-S** cannot be specified.

-S {h | n}:"network_selection_string"

Assigns *communication_group* to the heartbeat or network interface that is specified by *interface_selection_string*. Specify **-S h** for heartbeat interfaces or **-S n** for network interfaces.

By default, the **-S n** flag adds network interface resources that have IPv4 addresses to *communication_group*. If the **-6** flag is specified, the **-S n** flag will add network interface resources that have IPv6 addresses to *communication_group*.

If **-S** is specified, **-i** cannot be specified.

-6

Specifies that IPv6 addresses represented as resources on each interface have their communication group changed to the one specified. IPv4 addresses represented as resources on the interfaces would be unaffected.

By default (without **-6** specified), the inverse is true. Only IPv4 addresses represented as resources on the interface would have their communication group changed.

-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 new communication group that is to be created for the online peer domain. The name can contain any printable character.

Security

The user of the `mkcomg` command needs write permission for the `IBM.CommunicationGroup` resource class. Write permission for the `IBM.NetworkInterface` resource class is required to set the communication group for a network interface resource. By default, `root` on any node in the peer domain has read and write access to these resource classes 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 defined and online to the peer domain where the communication group is to be defined.

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

1. To define the communication group `ComGrp1` for the peer domain `App1Domain` and `nodeA` is defined and online to `App1Domain`, run this command on `nodeA`:

```
mkcomg ComGrp1
```

2. To define the communication group `ComGrp1` for the peer domain `App1Domain`, using a sensitivity of 1 and period of 3, and `nodeA` is defined and online to `App1Domain`, run this command on `nodeA`:

```
mkcomg -s 1 -p 3 ComGrp1
```

3. To define the communication group `ComGrp1` for the peer domain `App1Domain`, not using broadcast, using a priority of 3, and `nodeA` is defined and online to `App1Domain`, run this command on `nodeA`:

```
mkcomg -x b -t 3 ComGrp1
```

4. To define the communication group `ComGrp1` for the peer domain `App1Domain`, not using broadcast, not using source routing, and `nodeA` is defined and online to `App1Domain`, run the following command on `nodeA`:

```
mkcomg -x br ComGrp1
```

5. To define the communication group `ComGrp1` for the peer domain `App1Domain`, using a NIM path of `/opt/rsct/bin/hats_nim`, NIM parameters `-l 5` to set the logging level, and `nodeA` is defined and online to `App1Domain`, run this command on `nodeA`:

```
mkcomg -e /opt/rsct/bin/hats_nim -m "-l 5" ComGrp1
```

6. To define the communication group **ComGrp1** for **App1Domain** and assign **ComGrp1** to the heartbeat interface resource named **hbi0** on **nodeC**, run this command on **nodeA**:

```
mkcomg -i h:hbi0:nodeC ComGrp1
```

7. To define the communication group `ComGrp1` for the peer domain `App1Domain`, assign `ComGrp1` to the network interface resource named `eth0` on `nodeB`, and `nodeA` is defined and online to `App1Domain`, run this command on `nodeA`:

```
mkcomg -i n:eth0:nodeB ComGrp1
```

8. To define the communication group **ComGrp1** for **App1Domain** and assign **ComGrp1** to the heartbeat interface resource that uses the subnet `9.345.67.812`, run this command on **nodeA**:

```
mkcomg -S h:"Subnet == 9.345.67.812" ComGrp1
```

9. To define the communication group `ComGrp1` for the peer domain `App1Domain`, assign `ComGrp1` to the network interface resource that uses the subnet `9.123.45.678`, and `nodeA` is defined and online to `App1Domain`, run this command on `nodeA`:

```
mkcomg -S n:"Subnet == 9.123.45.678" ComGrp1
```

10. To define the communication group **ComGrp1** for **ApplDomain**, using a period of 500 milliseconds, run this command on **nodeA**:

```
mkcomg -p 0.5 ComGrp1
```

Location

`/opt/rsct/bin/mkcomg`

mkcondition Command

Purpose

Creates a new condition definition which can be monitored.

Syntax

```
mkcondition -r resource_class -e "event_expression" [ -E "rearm_expression" ] [ -d "event_description" ] [ -D "rearm_description" ] [ -b interval [, max_events] [, retention_period] [, max_totalsize] ] [ -m l | m | p ] [ -n node_name1 [, node_name2...] ] [ -p node_name ] [ --qnotoggle | --qtoggle ] [ -s "selection_string" ] [ -S c | w | i ] [ -g 0 | 1 | 2 ] [ -h ] [ -TV ] condition
```

```
mkcondition -c existing_condition [:node_name] [ -r resource_class ] [ -e "event_expression" ] [ -E "rearm_expression" ] [ -d "event_description" ] [ -D "rearm_description" ] [ -b interval [, max_events] [, retention_period] [, max_totalsize] ] [ -m l | m | p ] [ -n node_name1 [, node_name2...] ] [ -p node_name ] [ --qnotoggle | --qtoggle ] [ -s "selection_string" ] [ -S c | w | i ] [ -g 0 | 1 | 2 ] [ -h ] [ -TV ] condition
```

Description

The **mkcondition** command creates a new condition with the name specified by the condition parameter. The condition is used to monitor a resource for the occurrence of the condition (or event). Use the **mkresponse** command to define one or more responses to an event. You can then link the conditions to the responses using the **mkcondresp** command, or you can use the **startcondresp** command to link the responses and start monitoring.

Using the **-b** flag, multiple events can be batched or grouped together and passed to a response. The grouping of events is by the time span in which they occur. In addition, the grouping can be done such that a specified maximum number of events are grouped within the time span. A response that handles batched events must be defined as supporting batched events.

In a cluster environment, use the **-p** flag to specify the node in the domain that is to contain the condition definition. If you are using **mkcondition** on the management server and you want the condition to be defined on the management server, do *not* specify the **-p** flag. If the **-p** flag is not specified, the condition is defined on the local node. If the node where the condition will be defined is:

- in a cluster of nodes, the condition can monitor resources on more than one node. Use the **-n** flag to specify the nodes on which the condition will be monitored.
- the management server in a management domain, a management scope (**-m**) of local (**l**) or management domain (**m**) can be specified to indicate how the condition applies. The selection string will be evaluated using the entire management domain when management scope is set to the management domain and the node is the management server.
- a managed node in a management domain, only a management scope (**-m**) of local (**l**) can be used.
- in a peer domain, a management scope (**-m**) of peer domain (**p**) or local (**l**) can be used to indicate how the condition and the selection string apply.
- in both a management domain and a peer domain, a management scope (**-m**) of management domain (**m**), peer domain (**p**), or local (**l**) can be used to indicate how the condition and its selection string apply.

To lock a condition so it cannot be modified or removed, use the `chcondition` command (with its `-L` flag).

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

-b *interval* [, *max_events*] [, *retention_period*] [, *max_totalsize*]

Specifies one or more batching-related attributes. Use commas to separate the attribute values. Do not insert any spaces between the values or the commas.

interval specifies that the events are to be batched together for the indicated interval. Batching continues until no events are generated for an interval. Use an interval of 0 to turn batching off.

max_events specifies that the events are to be batched together until the *max_events* number of events are generated. The interval restarts if the *max_events* number of events is reached before the interval expires.

retention_period specifies the retention period in hours. The batched event file is saved for the time specified as the retention period. Once this time is reached, the file is automatically deleted.

max_totalsize specifies the total size for the batched event file in megabytes (MB). The batched event file is saved until this size is reached. Once the size is reached, the file is automatically deleted.

max_events, *retention_period*, and *max_totalsize* cannot be specified unless *interval* is greater than 0.

When *interval* is greater than 0 and *max_events* is 0, no maximum number of events is used.

If *retention_period* and *max_totalsize* are both specified, the batched event file is saved until the specified time or size is reached, whichever occurs first.

If you want to change one, two, or three attribute values, you must specify a valid value or an empty field for any attributes that precede the value you want to change. You do not have to specify any values for attributes that follow the value you want to change. For example, if you only need to change the retention period, you need to specify values for *interval* and *max_events* as well. You can provide an empty field if an attribute does not need to be changed. To change the retention period to 36 hours without changing the values of *interval* and *max_events*, enter:

```
mkcondition -c existing_condition -b ,,36
```

-c *existing_condition* [: *node_name*]

Copies an existing condition. The existing condition is defined on *node_name*. 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. If any other flags are specified, update the new condition as indicated by the flags. Links with responses are not copied.

-d "*event_description*"

Describes the event expression.

-D "*rearm_description*"

Describes the rearm expression.

-e "*event_expression*"

Specifies an *event expression*, which determines when an event occurs. An event expression consists of a dynamic attribute or a persistent attribute of *resource_class*, a mathematical comparison symbol (or `<`, for example), and a constant. When this expression evaluates to TRUE, an event is generated.

-E "*rearm_expression*"

Specifies a rearm expression. After *event_expression* has evaluated to True and an event is generated, the rearm expression determines when monitoring for the event expression will begin again. Typically, the rearm expression prevents multiple events from being generated for the same event evaluation. The rearm expression consists of dynamic attributes or persistent attributes of *resource_class*,

mathematical comparison symbols (> or <, for example), logical operators (|| or &&), constants, and an optional qualifier.

--g 0 | 1 | 2

Specifies granularity levels that control audit logging for the condition. The levels of granularity are:

0

Enables audit logging. ERRM writes all activities to the audit log. This is the default.

1

Enables error logging only. ERRM writes only in case of errors to the audit log.

2

Disables audit logging. ERRM does not write any records to the audit log.

-m l | m | p

Specifies the management scope to which the condition applies. The management scope determines how the condition is registered and how the selection string is evaluated. The scope can be different from the current configuration, but monitoring cannot be started until an appropriate scope is selected. The valid values are:

l

Specifies *local* scope. This is the default. The condition applies only to the local node (the node where the condition is defined; see the -p flag). Only the local node is used in evaluating the selection string.

m

Specifies *management domain* scope. The condition applies to the management domain in which the node where the condition is defined belongs (see the -p flag). All nodes in the management domain are used in evaluating the selection string. The node where the condition is defined must be the management server in order to use management domain scope.

p

Specifies *peer domain* scope. The condition applies to the peer domain in which the node where the condition is defined belongs (see the -p flag). All nodes in the peer domain are used in evaluating the selection string.

-n node_name1[,node_name2...]

Specifies the host name for a node (or a list of host names separated by commas for multiple nodes) where this condition will be monitored. Node group names can also be specified, which are expanded into a list of node names.

You must specify the -m flag with a value of **m** or **p** if you want to use the -n flag. This way, you can monitor conditions on specific nodes instead of the entire domain.

The host name does not have to be online in the current configuration, but once the condition is monitored, the condition will be in error if the node does not exist. The condition will remain in error until the node is valid.

-p node_name

Specifies the name of the node where the condition is defined. This is used in a cluster environment and the node name is the name by which the node is known in the domain. The default *node_name* is the local node on which the command runs. *node_name* is a node within the scope determined by the CT_MANAGEMENT_SCOPE environment variable.

If you are using mkcondition on the management server and you want the condition to be defined on the management server, do *not* specify the -p flag.

--qnotoggle

Specifies that monitoring does not toggle between the event expression and the rearm expression, but instead the event expression is always evaluated.

--qtoggle

Specifies that monitoring toggles between the event expression and the rearm expression.

-r *resource_class*

Specifies the resource class to be monitored by this condition. You can display the resource class names using the `lsrsrcdef` command.

-s "*selection_string*"

Specifies a selection string that is applied to all of the *resource_class* attributes to determine which resources should be monitored by the *event_expression*. The default is to monitor all resources within the *resource_class*. The resources used to evaluate the selection string is determined by the management scope (the `-m` flag). The selection string must be enclosed within double or single quotation marks. For information on how to specify selection strings, see the *RSCT: Administration Guide*.

-S c | w | i

Specifies the severity of the event:

c

Critical

w

Warning

i

Informational (the default)

-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

The *condition* name is a character string that identifies the condition. If the name contains spaces, it must be enclosed in quotation marks. A name cannot consist of all spaces, be null, or contain embedded double quotation marks.

Security

The user needs write permission for the IBM.Condition resource class to run `mkcondition`. 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) filesset 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 define a condition with the name "FileSystem space used" to check for percentage of space used greater than 90% and to rearm when the percentage is back down below 85%, enter:

```
mkcondition -r IBM.FileSystem \  
-e "PercentTotUsed > 90" -E "PercentTotUsed < 85" \  
"FileSystem space used"
```

2. To define a condition with the name "tmp space used" to check for percentage of space used greater than 90% for `/tmp` and to rearm when the percentage is back down below 85%, including comments, enter:

```
mkcondition -r IBM.FileSystem \  
-e "PercentTotUsed > 90" -E "PercentTotUsed < 85" \  
-d "Generate event when tmp > 90% full" \  
"tmp space used"
```

```
-D "Restart monitoring tmp again after back down < 85% full" \
-s 'Name=="/tmp"' "tmp space used"
```

3. To define a condition with the name "Space used" as a copy of "FileSystem space used", enter:

```
mkcondition -c "FileSystem space used" "Space used"
```

4. To define a condition with the name "var space used" as a copy of "tmp space used", but change the selection to **/var**, enter:

```
mkcondition -c "tmp space used" -s 'Name=="/var"' \
"var space used"
```

5. To define a condition with the name "vmstat is running" to monitor when user joe is running the vmstat program in a 64-bit environment, enter:

```
mkcondition -r "IBM.Program" \
-e "Processes.CurPidCount > 0" -E "Processes.CurPidCount <= 0" \
-d "Generate event when user starts vmstat" \
-D "Restart monitoring when vmstat is terminated" \
-s ProgramName == \"vmstat64\" && Filter==\"ruser==\\\"joe\\\"\" \
-S "i" -m "l" "vmstat is running"
```

6. To define a condition with the name "myscript terminated" to monitor when a script has ended, enter:

```
mkcondition -r "IBM.Program" \
-e "Processes.CurPidCount <= 0" -E "Processes.CurPidCount > 0" \
-d "Generate event when myscript is down" \
-D "Rearm the event when myscript is running" \
-s ProgramName == \"ksh\" && Filter == 'args[1]==\"/home/joe/myscript\"' \
-m "l" "myscript terminated"
```

In this example, `args` represents the array of argument strings that was passed to `main`. Because this is an array, `args[1]` references the first argument after the program name. Use the `ps -e1` command to determine the `ProgramName`. See the `lsrsrcdef` command for more information.

7. To batch together a maximum of 20 events at a time that come from a sensor named **DBInit** in 60-second intervals, enter:

```
mkcondition -r "IBM.Sensor" \
-e "Int32 < 0" -E "Int32 > 0" -b 60,20 \
-s "Name == \"DBInit\" \"DBInit Sensor"
```

8. To define a condition with the name `tmp space used` to check for percentage of space used greater than 90% for **/tmp** for at least seven out of the last 10 observations, including comments, enter:

```
mkcondition -r IBM.FileSystem \
-e "PercentTotUsed > 90 __QUAL_COUNT(7,10)" \
-d "Generate event when tmp > 90% full for 7 out of 10 last \
observations" \ -s 'Name=="/tmp"' "tmp space used"
```

9. To define a condition with the name `adapter stability` to check for adapter status that has changed four times within one minute, including comments, enter:

```
mkcondition -r IBM.NetworkInterface \
-e "OpState != OpState@P __QUAL_RATE(4,60)" \
-d "Generate event when OpState is changed 4 times within 1 minute" \
"adapter stability"
```

10. To define a condition for a batched event called `tmp space used` to check the percentage of space used by **/tmp** that is greater than 90%, with a batch interval of 5 and a batch event file retention period of 72 hours, enter:

```
mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" -b 5,,72 "tmp space used"
```

11. To define a condition called `tmp space used` to check that percentage of space used by **/tmp** that is greater than 90%, with audit logging enabled only in case of errors, enter:

```
mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" -g 1 "tmp space used"
```

These examples apply to management domains:

1. To define a condition with the name "FileSystem space used" to check for percentage of space used greater than 90%, to rearm when the percentage is back down below 85%, and to monitor all nodes in the domain, run this command on the management server:

```
mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \  
-E "PercentTotUsed < 85" -m d "FileSystem space used"
```

2. To define a condition with the name "FileSystem space used" to check for percentage of space used greater than 90%, to rearm when the percentage is back down below 85%, and to monitor nodes nodeA and nodeB in the domain, run this command on the management server:

```
mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \  
-E "PercentTotUsed < 85" -n nodeA,nodeB -m p \  
"FileSystem space used"
```

3. To define a condition with the name "nodeB FileSystem space used" on nodeB to check for percentage of space used greater than 90%, to rearm when the percentage is back down below 85%, and to monitor the condition with local scope, run this command on the management server:

```
mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \  
-E "PercentTotUsed < 85" -m l -p nodeB \  
"nodeB FileSystem space used"
```

4. To define a condition with the name "local FileSystem space used" to check for percentage of space used greater than 90%, to rearm when the percentage is back down below 85%, and to monitor the local node, run this command on a managed node:

```
mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \  
-E "PercentTotUsed < 85" -m l "local FileSystem space used"
```

These examples apply to peer domains:

1. To define a condition on nodeA with the name "FileSystem space used" to check for percentage of space used greater than 90%, to rearm when the percentage is back down below 85%, and to monitor all nodes in the domain, run this command:

```
mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \  
-E "PercentTotUsed < 85" -m p -p nodeA "FileSystem space used"
```

2. To define a condition on nodeC with the name "FileSystem space used" to check for percentage of space used greater than 90%, to rearm when the percentage is back down below 85%, and to monitor nodes nodeA and nodeB in the domain, run this command:

```
mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \  
-E "PercentTotUsed < 85" -n nodeA,nodeB -m p -p nodeC \  
"FileSystem space used"
```

3. To define a condition with the name "local FileSystem space used" on nodeB to check for percentage of space used greater than 90%, to rearm when the percentage is back down below 85%, and to monitor the local node only, run this command:

```
mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \  
-E "PercentTotUsed < 85" -m l -p nodeB "local FileSystem space used"
```

Location

/opt/rsct/bin/mkcondition

mkcondresp Command

Purpose

Creates a link between a condition and one or more responses.

Syntax

```
mkcondresp [-h] [-TV] condition[:node_name] response1 [response2...]
```

Description

The `mkcondresp` command creates a link between a condition and one or more responses. A link between a condition and a response is called a *condition/response association*. This command creates one or more condition/response associations; it does not start monitoring. In a cluster environment, the condition and the response must be defined on the same node. You can start monitoring for this condition and its linked responses later using the `startcondresp` command.

To lock a condition/response association, use the `-L` flag of the `rmcondresp`, `startcondresp`, or `stopcondresp` command.

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

condition

Specifies the name of the condition to be 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.

response1 [*response2*...]

Specifies one or more response names. All responses are linked to *condition*.

Security

The user needs write permission for the IBM.Association resource class to run `mkcondresp`. 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 link the condition "FileSystem space used" to the response "Broadcast event on-shift", run this command:

```
mkcondresp "FileSystem space used" "Broadcast event on-  
shift"
```

2. To link the condition "FileSystem space used" to the responses "Broadcast event on-shift" and "E-mail root anytime", run this command:

```
mkcondresp "FileSystem space used" "Broadcast event on-shift" "E-mail root anytime"
```

These examples apply to management domains:

1. To link the condition "FileSystem space used" on the management server to the response "Broadcast event on-shift" (also on the management server), run this command on the management server:

```
mkcondresp "FileSystem space used" "Broadcast event on-shift"
```

2. To link the condition "FileSystem space used" on the management server to the response "Broadcastevent on-shift", run this command on one of the nodes in the domain:

```
mkcondresp "FileSystem space used":nodeA "Broadcast event on-shift"
```

This example applies to peer domains:

1. To link the condition "FileSystem space used" on node nodeA to the response "Broadcastevent on-shift" (also on nodeA), run this command on one of the nodes in the domain:

```
mkcondresp "FileSystem space used":nodeA "Broadcast event on-shift"
```

Location

/opt/rsct/bin/mkcondresp

mkcosi Command

Purpose

Makes a Common Operating System Image (COSI) for use with thin servers.

Syntax

```
mkcosi -s Source [-l Location] [-S Server] [-v] COSI
```

Description

The `mkcosi` command creates a Common Operating System Image (COSI). A COSI is a repository that contains all the necessary software to bring a thin server up to a functional state. The `mkcosi` command takes a source (`-s Source`) containing installable images and attempts to install those software images into a specific location (`-l Location`). If the `-S Server` is specified, the COSI image is stored on that particular server. The result is an OS image that can be used by thin servers as its boot image and operating system.

This command is dependent upon the `bos.sysmgt.nim.master` fileset being present on the system. When this command is executed for the first time, the machine executing the command is configured as a NIM master. The `mkcosi` command uses the `nim_master_setup` command to configure the machine as a NIM master. The `-S` parameter must point to a machine that is managed by the caller of the `mkcosi` command.

Flags

Item	Description
-l	Specifies the full path name to a location for storing the COSI.
-S <i>Server</i>	Specifies the name of the machine where the COSI image resides.
-s <i>Source</i>	Specifies the source of installable images to be used in creating the COSI. The source can be an <code>lpp_source</code> , a device with installable media, a directory to installable images, or a remote location to installable images.
-v	Enables verbose debug output when the <code>mkcosi</code> 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 `mkcosi` command.

Examples

1. To define a COSI named `cosi1` from a CD-ROM `cd0`, and to store it at the location `/export/cosi1`, enter the following command:

```
mkcosi -s cd0 -l /export/cosi1 cosi1
```

Location

`/usr/sbin/mkcosi`

Files

Item	Description
<code>/etc/niminfo</code>	Contains variables used by NIM.

mkdev Command

Purpose

Adds a device to the system.

Syntax

```
mkdev { -c Class -s Subclass -t Type } [ -l Name ] [ -a Attribute=Value ] ... [ -d | -S | -R ] [ -f File ] [ -h ] [ -p ParentName ] [ -q ] [ -w ConnectionLocation ]
```

mkdev -l *Name* [-h] [-q] [-S]

Description



Attention: To protect the Configuration Database, the **mkdev** command is not interruptible. Stopping this command before it is complete could result in a corrupted database.

The **mkdev** command performs the following:

- Defines and makes available a device with the given device class (**-c** *Class* flag), type (**-t** *Type* flag), subclass (**-s** *Subclass* flag), connection location (**-w** *ConnectionLocation* flag), and the device logical name of the parent (**-p** *ParentName* flag)
- Makes available the previously defined device specified by the given device logical name (**-l** *Name* flag).

You can use any combination of the **-c**, **-s**, and **-t** flags you need to uniquely identify the predefined device.

If you specify the **-d** flag, the **mkdev** command only defines the device. If you specify the **-S** flag, the **mkdev** command brings the device to the Stopped state, if this state is supported, and does not make the device available. If you do not specify either the **-d** flag or the **-S** flag, the **mkdev** command makes the device available.

If you specify the **-R** flag, the **mkdev** command configures any previously-defined parents of the specified device that are not already configured. The **-R** flag is not compatible with the **-d** and **-S** flags.

By using the **-l** flag with the **-c**, **-s**, and **-t** flags, you can specify the name of the device. If you do not use the **-l** flag, a name will be automatically generated and assigned. Not all devices support user-supplied names.

Note: Queue device names must begin with an alphabetic character.

When using the **mkdev** command, you can supply the flags either on the command line or in the specified **-f** *File* flag.

You can use the System Management Interface Tool (SMIT) **smit mkdev** fast path to run this command.

Flags

Item	Description
-a <i>Attribute=Value</i>	Specifies the device attribute-value pairs to be used instead of the defaults. The <i>Attribute=Value</i> variable can be used to specify one attribute value pair or multiple attribute value pairs for one -a flag. Multiple attribute-value pairs must be enclosed in quotation marks with a blank space between the pairs. For example, entering -a Attribute=Value lists one attribute value pair per flag, while entering -a 'Attribute1=Value1 Attribute2=Value2' lists more than one attribute value pair. This flag cannot be used with the -l flag unless the -c , -s , and -t flags are also used.
-c <i>Class</i>	Specifies the device class.
-d	Defines the device in the Customized Devices object class. If you specify the -d flag, the mkdev command does not make the device available. This flag cannot be used with the -S flag.
-f <i>File</i>	Reads the necessary flags from the <i>File</i> parameter.
-h	Displays the command usage message.
-l <i>Name</i>	Specifies the predefined device, indicated by the <i>Name</i> variable, in the Customized Devices object class when not used with the -c , -s , and -t flags. The -a , -p , and -w flags cannot be used in this case. Queue device names must begin with an alphabetic character.

Item	Description
-p <i>ParentName</i>	Specifies the device name, indicated by the <i>ParentName</i> variable, that you want assigned to the device when it is used with the -c , -s , and -t flags. Not all devices support this feature. This flag cannot be used with the -l flag unless the -c , -s , and -t flags are also used.
-q	Suppresses the command output messages from standard output and standard error.
-R	Configures any parents of the device that are not already configured. This flag cannot be used with the -d and -S flags.
-S	Prevents the device from being set to the Available state. This flag is only meaningful for those devices that support the Stopped state. This flag cannot be used with the -d flag.
-s <i>Subclass</i>	Specifies the subclass, indicated by the <i>Subclass</i> variable, of the device.
-t <i>Type</i>	Specifies the device type from the Predefined Devices object class.
-w <i>ConnectionLocation</i>	Specifies the connection location, indicated by the <i>ConnectionLocation</i> variable, on the parent. This flag cannot be used with the -l flag unless the -c , -s , and -t flags are also used.

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.

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:

Event	Information
DEV_Create	Method name, parameters
DEV_Configure	Errors
DEV_Start	Device name
DEV_Change	Parameters

Examples

1. To define (but not configure) a 4.0 GB 4mm Tape Drive connected to the scsi0 SCSI adapter and using SCSI ID 5 and LUN of 0, type the following:

```
mkdev -d -c tape -t4mm2gb -s scsi -p scsi0 -w 5,0
```

The system displays a message similar to the following:

```
rmt4 defined
```

2. To make the predefined `rmt0` tape device available to use, type the following:

```
mkdev -l rmt0
```

The system displays a message similar to the following:

```
rmt0 available
```

3. To define and configure an RS-232 tty device connected to port 0 on the IBM 8-Port EIA-232/RS-422A (PCI) Adapter with the speed attribute set to 19200, and other attributes set from the `foo` file, type the following:

```
mkdev -t tty -s rs232 -p sa3 -w 0 -a speed=19200 -f foo
```

The system displays a message similar to the following:

```
tty0 available
```

Files

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

mkdir Command

Purpose

Creates one or more new directories.

Syntax

```
mkdir [-e] [ -m Mode ] [ -p ] Directory ...
```

Description

The **mkdir** command creates one or more new directories specified by the *Directory* parameter. Each new directory contains the standard entries `.` (dot) and `..` (dot-dot). You can specify the permissions for the new directories with the **-m** *Mode* flag. You can use the **umask** subroutine to set the default mode for the **mkdir** command.

The owner-ID and group-ID of the new directories are set to the process's effective user-ID and group-ID, respectively. The setgid bit setting is inherited from the parent directory. To change the setgid bit, you can either specify the **-m** *Mode* flag or issue the **chmod** command after the creation of the directory.

Note: To make a new directory you must have write permission in the parent directory.

Flags

Item	Description
-e	Creates directories with encryption inheritance.

Item	Description
-m Mode	<p>Sets the permission bits for the newly-created directories to the value specified by the <i>Mode</i> variable. The <i>Mode</i> variable takes the same values as the <i>Mode</i> parameter for the chmod command, either in symbolic or numeric form.</p> <p>When you specify the -m flag using symbolic format, the op characters + (plus) and - (minus) are interpreted relative to the assumed permission setting a=rwx. The + adds permissions to the default mode, and the - deletes permissions from the default mode. Refer to the chmod command for a complete description of permission bits and formats.</p>
-p	<p>Creates missing intermediate path name directories. If the -p flag is not specified, the parent directory of each-newly created directory must already exist.</p> <p>Intermediate directories are created through the automatic invocation of the following mkdir commands:</p> <pre style="background-color: #f0f0f0; padding: 5px;">mkdir -p -m \$(umask -S),u+wx \$(dirname Directory) && mkdir [-m Mode] Directory</pre> <p>where the [-m Mode] represents any option supplied with your original invocation of the mkdir command.</p> <p>The mkdir command ignores any <i>Directory</i> parameter that names an existing directory. No error is issued.</p>

Exit Status

This command returns the following exit values:

Item	Description
0	All the specified directories were created successfully, or the -p option was specified and all the specified directories now exist.
>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 a new directory called Test in the current working directory, enter:

```
mkdir Test
```

The Test directory is created with default permissions.

2. To create a new directory called Test with rwxr-xr-x permissions in the previously created /home/demo/sub1 directory, enter:

```
mkdir -m 755 /home/demo/sub1/Test
```

3. To create a new directory called Test with default permissions in the /home/demo/sub2 directory, enter:

```
mkdir -p /home/demo/sub2/Test
```

The **-p** flag creates the /home, /home/demo, and /home/demo/sub2 directories if they do not already exist.

Files

Item	Description
/usr/bin/mkdir	Contains the mkdir command.

mkdirhier Command

Purpose

Creates a hierarchy of directories or a single directory.

Syntax

```
mkdirhier Directory ...
```

Description

The **mkdirhier** command creates the specified directories. Unlike the **mkdir** command, if any of the parent directories of the specified directory do not exist, the **mkdirhier** command creates those directories as well as the specified directory.

Example

To create a directory named **foo2** or to create a hierarchy of directories named **foo**, **foo1**, and **foo2**, enter:

```
mkdirhier ~/foo/foo1/foo2
```

If **foo** and **foo1** already exist then the command creates **foo2**. However, if none of them exist then the command creates all three new directories.

mkdom Command

Purpose

Creates a new domain.

Syntax

```
mkdom [ -R load_module ] [Attribute = Value ...] Name
```

Description

The **mkdom** command creates a new domain in the domain database. The domain attributes can be set during the domain creation phase by using the *Attribute = Value* parameter.

When the system is operating in the enhanced Role Based Access Control (RBAC) mode, modifications made to the domain database are not used for security considerations until the database is sent to the kernel security tables by using the **setkst** command.

Note: The domain id value can be lesser than or equal to 1024. The **mkdom** command enables you to create 1024 domains on the system.

If the system is configured to use multiple authentication load modules for the role-based access control (RBAC) domain database, the new RBAC domain is created in the first load module specified by the `secorder` attribute in the domains stanza of the `/etc/nscontrol.conf` file. Use the **-R** flag to create an RBAC domain in a specific authentication load module.

Flags

Item	Description
-R <i>load_module</i>	Specifies the loadable module that is to be used when you create an RBAC domain.

Parameters

Item	Description
<i>Attribute = Value</i>	Initializes a domain attribute. See the chdom command for valid attributes and values.
<i>Name</i>	Specifies a unique domain name string.

Restrictions on creating domain names: The *Name* parameter specified must be unique and is limited to a 63 single-byte printable character. While the **mkdom** command supports multibyte domain names, it is recommended that you restrict domain names to characters within the POSIX portable file name character set. Domain names must not begin with a - (dash), + (plus sign), @ (at sign), or ~ (tilde) and must not contain any space, tab, or new-line characters. You cannot use the keywords ALL, default, ALLOW_OWNER, ALLOW_GROUP, ALLOW_ALL or * as a domain name. Additionally, do not use any of the following characters within a domain string:

Item	Description
:	Colon
"	Double quotation mark
#	Number sign
,	Comma
=	Equal sign
\	Backslash
/	Forward slash
?	Question mark
'	Single quotation marks
`	Grave accent

Security

The **mkdom** command is a privileged command. Callers of the command must have activated a role that has the following authorization to run the command successfully.

Item	Description
aix.security.domains.create	Required to run the command.

Files Accessed

Item	Description
File	Mode
<code>/etc/security/domains</code>	rw

Examples

1. To create a domain `h1rdom` and to have the `mkdom` command assign an appropriate ID value, enter the following command:

```
mkdom h1rdom
```

2. To create a custom domain in Lightweight Directory Access Protocol (LDAP), enter the following command:

```
mkdom -R LDAP custom
```

mkdvd Command

Purpose

Creates multi-volume DVDs from a `mksysb`, `savevg`, or `savewpar` backup image.

Syntax

```
mkdvd -r directory | -d dvddevice | -S [ -m mksysbimage | -M mksysbtarget | -s savevgimage | -v savevgvolumegroup | -w savewpar_image | -W wparname ] [ -C cdfsdir ] [ -I cdimagedir ] [ -V dvdsvolumegroup ] [ -B ] [ -p pkgsourcedir ] [ -R | -S ] [ -i image.data ] [ -u bosinst.data ] [ -f wparspecificationfile ] [ -e ] [ -P ] [ -l packagelist ] [ -b bundlefile ] [ -z customfile ] [ -D ] [ -U ] [ -Y ] [ -n ] [ -a ] [ -A ] [ -c ] [ -Z ] [ -G | -N ] [ -x file ] [ -T ]
```

Description

The `mkdvd` command creates a system backup image (`mksysb`) to DVD-Recordable (DVD-R, DVD-RAM) from the system `rootvg` or from a previously created `mksysb` image. It creates a volume group backup image (`savevg`) to DVD from a user-specified volume group or from a previously created `savevg` image. It also creates the backup image of a workload partition (`savewpar`) to DVD from a user-specified workload partition or from a previously created `savewpar` image.

Note: If the system has a `multibos` environment where both instances are mounted, you can restore the backup only by using the `alt_disk_mksysb` command.

For DVD media, system backups that are made with the `mkdvd` command have a limitation in that they expect the media to be 4.7 GB or larger per side. The `mkdvd` command does not process the next volume until it writes over 4 GB on the current volume, thus the use of smaller media would result in corruption when you go beyond the media capacity.

When a bootable backup of a root volume group is created, the boot image reflects the currently running kernel. If the current kernel is the 64-bit kernel, the backup boot image is also 64 bit, and it boots 64-bit systems only. If the current kernel is a 32-bit kernel, the backup boot image is 32 bit, and it can boot both 32-bit and 64-bit systems.

With the `mkdvd` command, you can create bootable and non-bootable DVDs in Rock Ridge (ISO9660) or UDF (Universal Disk Format) format.

Note: The functionality that is required to create Rock Ridge format DVD images and to write the DVD image to the DVD-RAM device is not part of the `mkdvd` command. You must supply additional code to

the **mkdvd** command to do these tasks. You can call the code by using shell scripts and then link it to **/usr/sbin/mkrr_fs** (for creating the Rock Ridge format image) and **/usr/sbin/burn_cd** (for writing to the DVD device). Both links are called from the **mkdvd** command.

Some sample shell scripts are included for different vendor-specific routines. You can find these scripts in **/usr/samples/oem_cdwriters**.

If you do not supply any file systems or directories as command parameters, the **mkdvd** command creates the necessary file systems and removes them when the command finishes running. File systems that you supply are checked for adequate space and write access.

Note: If the **mkdvd** command creates file systems in the backup volume group, they are excluded from the backup.

If you must create multi-volume DVDs because the volume group image does not fit on one DVD, the **mkdvd** command provides instructions for DVD replacement and removal until all the volumes have been created.

Flags

Item	Description
-a	Does not back up extended attributes or NFS4 ACLs.
-A	Backs up DMAPI file system files.
-b <i>bundlefile</i>	Gives the full path name of the file that contains a list of filesets to be installed after the mksysb is restored. This file is copied to ./usr/sys/inst.data/user_bundles/bundle_file in the DVD file system and also copied to RAM in case the DVD is unmounted. The file would be listed as BUNDLES=../usr/sys/inst.data/user_bundles/bundlefile in the bosinst.data file.
-B	Prevents mkdvd from adding boot images (non-bootable DVD) to the DVD. Use this flag if you create a mksysb DVD that you will not boot. Before you install the non-bootable mksysb DVD, you must boot a same level (V.R.M.) product media. The mkdvd command defaults to creating a bootable DVD for the machine type of the source system. For more information, see the Notes section.
-c	Does not compress or pack files as they are backed up.
-C <i>cdfsdir</i>	Specifies the file system that is used to create the DVD file system structure, which must have up to 4.38 GB for DVD sized images. The DVD image consumes only as much room as is necessary to contain all the data on the DVD. If you do not specify the -C flag and the /mkcd/cd_fs directory exists, the mkdvd command uses that directory. If you do not specify the -C flag and the /mkcd/cd_fs directory does not exist, the mkdvd command creates the file system /mkcd/cd_fs and removes it when the command finishes running. The command creates the file system in the volume group that is indicated with the -V flag, or rootvg if that flag is not used. Each time that you invoke the mkdvd command, a unique subdirectory (by using the process id) is created under the /mkcd/cd_fs directory, or in the directory that is specified with the -C flag. Note: If performing DVD sized backups, the file systems must be <i>large file enabled</i> . This type of backup also requires setting the file ulimit size to <i>unlimited</i> .
-d <i>vddevice</i>	Indicates the DVD-R or DVD-RAM device (/dev/cd1 , for instance). This flag is required unless you use the -S flag.

Item	Description
-D	Turns on the debug output information feature. The default is no debug output.
-e	Excludes the files and directories from the backup image that is listed in the <i>/etc/exclude.volume_group</i> file. You cannot use this flag with the -m or -s flags.
-f wparspecificationfile	Specifies the user-supplied WPAR specification file. This specification file of workload partition takes precedence over the wpar.spec file in the savewpar image. If you do not use the -f flag, the mkdvd command restores the wpar.spec from the specified savewpar image, or generates a new wpar.spec file during the creation of savewpar .
-i image.data	Specifies the user-supplied <i>image.data</i> file. This data file takes precedence over the image.data file in the mksysb image. If you do not specify the -i flag, then the mkdvd command restores the image.data from the given mksysb image, or generates a new image.data file during the creation of mksysb . Note: The -i flag cannot be used to specify a user-supplied <i>vgname.data</i> file for use with a savevg image.
-I cdimagesdir	Specifies the directory or file system where the final DVD images are stored before they are written to the DVD-R or DVD-RAM device. If this flag is not used, the mkdvd command uses the /mkcd/cd_images directory if it already exists. If not, the command creates the /mkcd/cd_images file system in the volume group that is given with the -V flag, or in rootvg if that flag is not used. If the mkdvd command creates the file system, it is removed upon command completion, unless either the -R or -S flag is used. If the -R or -S flag is used, consideration must be made for adequate file system, directory, or disk space, especially when you create multi-volume DVDs. The DVD image consumes only as much room as is necessary to contain all the data on the DVD. Note: If performing DVD sized backups, the file systems must be <i>large file enabled</i> . This type of backup also requires setting the file ulimit size to <i>unlimited</i> .
-l packagelist	Specifies the file that contains a list of additional packages you want copied to the ./usr/lpp/inst.images directory of the DVD file system. The images are copied from the location that is named with the -p flag. If you use the -l flag, you must also use the -p flag.
-m mksysbimage	Specifies a previously created mksysb image. If you do not specify the -m flag, the mkdvd command calls mksysb . For more information about where the mksysb image is placed, see the -M flag.

Item	Description
-M <i>mksysbtarget</i>	<p>States the directory or file system where the mksysb or savevg image is stored if a previously created backup is not given with the -m or -s flags. If the -M flag is not used and a mksysb or savevg image is not provided, the mkdvd command verifies that /mkcd/mksysbimage exists. If the directory does not exist, then the mkdvd command creates a separate file system, /mkcd/mksysbimage, where the mksysb or savevg images are temporarily stored. The command creates the file system in the volume group that is given with the -V flag, or in rootvg if that flag is not used.</p> <p>Note: If performing DVD sized backups, the file systems must be <i>large file enabled</i>. This type of backup also requires setting the file ulimit size to <i>unlimited</i>.</p>
-n	<p>Backs up user volume group information and administration data files. This flag backs up files such as /tmp/vgdata/vgname/vgname.data and map files, if any exist. This flag does not back up user data files. This backup can be used to create a user volume group without restoring user data files. This action cannot be done to rootvg.</p>
-N	<p>Includes file systems that belong to a workload partition (WPAR) in the defined state in the system backup.</p> <p>Note: To be included in the backup, all file systems that belong to a WPAR in the defined state must be in the rootvg volume group.</p>
-p <i>pkgsourcedir</i>	<p>Names the directory or device that contains device and kernel package images. The device must be a CD or DVD device (for example, /dev/cd0). If you use the same DVD-R or DVD-RAM device that you gave with the -d flag, the product media must be inserted into the drive first. The mkdvd command then prompts you to insert the writable DVD before the actual DVD creation.</p>
-P	<p>Creates physical partition mapping during the mksysb or savevg creation. You cannot use this flag with the -m or -s flags.</p>
-r <i>directory</i>	<p>Indicates existing directory structure to burn onto a DVD. This flag makes a DVD image that is a copy of the specified directory structure.</p>
-R	<p>Prevents the mkdvd command from removing the final DVD images. The mkdvd command defaults by removing everything that it creates when it finishes running. The -R flag allows multiple DVD image sets to be stored, or for DVD creation (burn) to occur on another system. If multiple volumes are needed, the final images are uniquely named by using the process ID and volume suffixes.</p>
-s <i>savevgimage</i>	<p>Indicates a previously created savevg image. See Notes for details.</p>
-S	<p>Stops the mkdvd command before it writes to the DVD-R or DVD-RAM without removing the final DVD images. The -S flag allows multiple DVD sets to be created, or for DVDs to be created on another system. The images remain in the directory marked by the -I flag, or in the /mkcd/cd_images directory if the -I flag is not used. If multiple volumes are required, the final images are uniquely named by using the process ID and volume suffixes.</p>

Item	Description
-T	<p>Creates backup by using snapshots. This command applies only to JFS2 file systems.</p> <p>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 back up 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 thus, 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.</p>
-u <i>bosinst.data</i>	<p>Specifies the user-supplied <i>bosinst.data</i> file. This data file takes precedence over the bosinst.data file in the mksysb image. If you do not specify the -u flag, then the mkdvd command restores bosinst.data from the specified mksysb image, or generates a new bosinst.data file during the creation of mksysb.</p>
-U	<p>Creates a UDF (Universal Disk Format) file system on DVD-RAM media. It does not require the amount of free space that is needed to create Rock Ridge format backups. It does not need the /mkcd/cd_fs and /mkcd/cd_images file systems. Therefore, the only temporary disk space it needs is to create the backup image that will be copied to the media. This means that the -I and -C flags do not apply to the -U flag. Because the backup is copied to the media, images cannot be created and burned later or on another system. So, the -R flag and -S flag do not apply when you use the -U flag. You must specify a device to write to with the -d flag. The -U flag does not use the /usr/sbin/mkrr_fs or /usr/sbin/burn_cd file systems.</p>
-v <i>savevgvolumegroup</i>	<p>Denotes the volume group to be backed up using the savevg command. See Notes for details. For more information about where the savevg image is placed, see the -M flag.</p>
-V <i>dvdsvolumegroup</i>	<p>Indicates the volume group that is used when you create the file systems needed for the mkdvd command. If the -V flag is not given and a file system is needed but not there (because it was not supplied with other flags), then rootvg is the default volume group for creating the file systems. If the mkdvd command creates the file systems in the backup volume group, those file systems are not included as part of the backup image. The mkdvd-created file systems are removed upon the command's completion.</p>
-w <i>savewparimage</i>	<p>Indicates a previously created savewpar image.</p>
-W <i>wparname</i>	<p>Denotes the workload partition to be backed up using the savewpar command.</p>
-Y	<p>Accepts licenses.</p>

Item	Description
-z <i>customsfile</i>	States the full path name of the file to be copied to the root directory of the DVD file system. This file can be a customization script that is specified in the bosinst.data file, such as CUSTOMIZATION_FILE=filename. For example: If the file my_script is in /tmp on the machine where mkdvd is running, then enter -z/tmp/my_script and specify CUSTOMIZATION_FILE=my_script. The code copies the script to the root directory of the RAM file system before it runs.
-Z	Specifies that the Encrypted file system (EFS) information for all the files, directories, and file systems is not backed up.
-G	Excludes WPAR file systems from the system backup. This flag is not valid with -N flag.
-x <i>file</i>	Excludes the file systems that are listed in the file from the system backup. File system mount points must be listed one per line.

Note: Use care when you exclude file systems as a resulting backup can be unusable for system restoration.

Note:

- If you are creating a non-bootable DVD (by using the **-B** flag), you cannot use the **-p** or **-l** flags.
- If you are creating a non-bootable DVD with a **savevg** image (by using the **-s** or **-v** flags), you cannot use the **-p**, **-l**, **-u**, **-i**, **-z**, or **-b** flags.

Examples

1. To generate a bootable system backup to the DVD-R device named `/dev/cd1`, enter the following command:

```
mkdvd -d /dev/cd1
```

2. To generate a system backup to the DVD-R or DVD-RAM device named `/dev/cd1`, enter the following command:

```
mkdvd -d /dev/cd1
```

3. To generate a non-bootable volume group backup of the volume group `myvg` to `/dev/cd1`, enter the following command:

```
mkdvd -d /dev/cd1 -v myvg
```

Note: All **savevg** backup images are non-bootable.

4. To generate a non-bootable backup of the workload partition `mywpar` to `/dev/cd1`, enter the following command:

```
mkdvd -d /dev/cd1 -W mywpar
```

Note: All **savewpar** backup images are not bootable.

5. To generate a non-bootable backup of the workload partition `mywpar` to **`/dev/cd1`** from the previously generated **savewpar** image **`/wparbackups/mywpar.bff`**, enter the following command:

```
mkdvd -d /dev/cd1 -w /wparbackups/mywpar.bff
```

6. To create a DVD or DVD that duplicates an existing directory structure such as the following example:

```
/mycd/a
/mycd/b/d
/mycd/c/£/g
```

enter the following command:

```
mkdvd -r /mycd -d /dev/cd1
```

After you mount with `mount -o ro /dev/cd1 /mnt, cd to /mnt; a find . -print command displays:`

```
./a
./b
./b/d
./c
./c/£
./c/£/g
```

Files

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

mkfifo Command

Purpose

Makes first-in-first-out (FIFO) special files.

Syntax

```
mkfifo [ -m Mode ] File ...
```

Description

The **mkfifo** command creates FIFO special files specified by the *File* parameter, in the order specified. If the **-m** *Mode* flag is not specified, the file mode of the FIFO file is the bitwise inclusive OR of the **S_IRUSR**, **S_IWUSR**, **S_IRGRP**, **S_IWGRP**, **S_IROTH**, and **S_IWOTH** permissions as modified by the file mode creation (see the **umask** command).

The **mkfifo** command functions similarly to the **mkfifo** subroutine.

Flags

Item	Description
-m <i>Mode</i>	Sets the file permission bits of the newly created FIFO file to the specified mode values. The <i>Mode</i> variable is the same as the mode operand defined for the chmod command. The characters + (plus sign) and - (minus sign), if used, are interpreted relative to the initial value a=rw (that is, having permissions of rw-rw-rw-).

Exit Status

This command returns the following exit values:

Item Description

- 0** All the specified FIFO special files were created successfully.
- >0** An error occurred.

Examples

1. To create a FIFO special file with permissions `prw-r--`, enter:

```
mkfifo -m 644 /tmp/myfifo
```

This command creates the `/tmp/myfifo` file with read/write permissions for the owner and read permission for the group and for others.

2. To create a FIFO special file using the `-` (minus sign) operand to set permissions of `prw-r---`, enter:

```
mkfifo -m g-w,o-rw /tmp/fifo2
```

This command creates the `/tmp/fifo2` file, removing write permission for the group and all permissions for others.

Note: If more than one file is created using the `-` (minus sign) operand, separate each mode specifier with a comma and no spaces.

Files

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

mkfilt Command

Purpose

Activates or deactivates the filter rules.

Syntax

```
mkfilt -v 4 | 6 [ -d ] [ -u ] [ -z P | D ] [ -g start | stop ] [ -i ]
```

Description

Use the **mkfilt** command to activate or deactivate the filter rules. This command can also be used to control the filter logging function. IPsec filter rules for this command can be configured using the `genfilt` command or IPsec `smit` (IP version 4 or IP version 6).

Flags

Item	Description
-v	IP version of the rules you want to activate. The value of 4 specifies IP version 4 and the value of 6 specifies IP version 6. The default (when this flag is not used) is to activate both IP version 4 and IP version 6. All the filter rules defined in the filter rule table for the IP version(s) will be activated or deactivated.
-d	Deactivates the active filter rules. This flag cannot be used with the -u flag.

Item	Description
-u	Activates the filter rules in the filter rule table. This flag cannot be used with the -d flag.
-z	Sets the action of the default filter rule to Permit (P) or Deny (D). The default filter rule is the last rule in the filter rule table that will apply to traffic that does not apply to any other filter rules in the table. Setting the action of this rule to Permit will allow all traffic that does not apply to any other filter rules. Setting this action to Deny will not allow traffic that does not apply to any other filter rules.
-g	This flag is used to either start (start) or stop (stop) the log functionality of the filter rule module.
-i	Initialization flag. This flag only applies when the -u flag is also used. If the -i flag is used, all the filter rules with an "active" status will be activated. If not used, all the filter rules in the filter rule table will be activated.

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.

mkfont Command

Purpose

Adds a font path name to the Object Data Manager (ODM) that is loaded by the low function terminal (LFT) at boot time.

Syntax

mkfont [*FontPathName*]

Description

The **mkfont** command adds a fully qualified font file path name to the ODM. At boot time, the LFT loads the new font and any other fonts found in the ODM. The list of font information acquired by the LFT is passed to the default display device driver. The display driver selects from this list the font that best fits the display. If a default font was selected using the **chfont** command, the device driver uses that font.

Note: This command can be run only from an LFT.

You can use the System Management Interface Tool (SMIT) **smit mkfont** fast path to run this command.

Parameter

Item	Description
<i>FontPathName</i>	The fully qualified pathname of a font file.

Security

The user must have root authority to execute this command.

Example

To add the font file **/usr/lpp/fonts/Rom10.snf**, enter the following command:

```
mkfont /usr/lpp/fonts/Rom10.snf
```

Files

Item	Description
<code>/bin/mkfont</code>	Contains the mkfont command.
<code>/usr/lpp/fonts</code>	Contains the font directory.

mkfontdir Command

Purpose

Creates a **fonts.dir** file from a directory of font files.

Syntax

```
mkfontdir [ DirectoryName ... ]
```

Description

The **mkfontdir** command creates a **fonts.dir** file from a directory of font files. For each directory argument, the **mkfontdir** command reads all of the bitmapped font files in the directory, searching for properties named FONT or the name of the file stripped of its suffix. These are used as font names, which are written to the **fonts.dir** file in the directory along with the name of the font file. The **fonts.dir** file is then used by the X server and the Font server to determine which fonts are available.

The kinds of font files read by the **mkfontdir** command depend upon the configuration parameters and typically include the following formats:

Item	Description
Portable Compile Format	(suffix .pcf)
Compressed PCF	(suffix .pcf.Z)
Server Natural Format	(suffix .snf)
Compressed SNF	(suffix .snf.Z)
Bitmap Distribution Format	(suffix .bdf)
Compressed BDF	(suffix .bdf.Z)

If a font exists in multiple formats, the most efficient format is used (PCF format before SNF then BDF formats).

Scalable fonts are not automatically recognized by **mkfontdir**. You can construct a **fonts.scale** file (the format is identical to that in the **fonts.dir** file) containing entries for scalable fonts. Then, when you run **mkfontdir** on a directory, it copies entries from the **fonts.scale** file in that directory into the **fonts.dir** file it constructs in that directory.

You can create the **fonts.alias** file, which can be put in any directory of the font path, to map new names to existing fonts. This file should be edited by hand. The format is two columns separated by white space, with the first column containing aliases and the second column containing font-name patterns.

When a font alias is used by an X client, the X server searches for the name it references by looking through each font directory in turn. Therefore, the aliases and the font files do not need to be in the same directory.

To embed white space in aliases or font-name patterns, enclose them in double-quotation marks. To embed double-quotation marks, or any other characters, precede each character with a \ (backslash).

```
"magic-alias with spaces" "\"font\name\"with quotes"  
regular-alias                fixed
```

If the character string **FILE_NAMES_ALIASES** stands alone on a line, each file name in the directory when stripped of its suffix (such as **.pcf** or **.pcf.Z**) is used as an alias for that font.

The X server and the Font Server look for **fonts.dir** and **fonts.alias** files in each directory in the font path each time the font path is set.

Examples

To create a **fonts.dir** file from a directory of font files, enter:

```
mkfontdir DirectoryName
```

If no directory name is specified, the **mkfontdir** command reads the current directory.

Files

Item	Description
<code>/usr/lib/X11/fonts</code>	Is the directory containing font files, fonts.dir and fonts.alias files.

mkfs Command

Purpose

Makes a file system.

Syntax

```
mkfs [ -b Boot ] [ -l Label ] [ -i i-Nodes ] [ -o Options ] [ -p Prototype ] [ -s Size ] [ -v VolumeLabel ] [ -V VfsName ] Device
```

Description

The **mkfs** command makes a new file system on a specified device. The **mkfs** command initializes the volume label, file system label, and startup block.

The *Device* parameter specifies a block device name, raw device name, or file system name. If the parameter specifies a file system name, the **mkfs** command uses this name to obtain the following parameters from the applicable stanza in the [/etc/filesystems](#) file, unless these parameters are entered with the **mkfs** command:

Item	Description
dev	Device name
vol	Volume ID
size	File system size
boot	Program to be installed in the startup block
vfs	Definition of the virtual file system
options	File-system implementation-specific options of the form <i>Keyword, Keyword=Value</i>

Note:

1. The file system is created with the setgid (set group ID) bit enabled. The setgid bit determines the default group permissions. All directories created under the new file system have the same default group permissions.
2. The **mkfs** command does not alter anything in a mounted file system, including the file system label. The file system label changes when you change the mount point, unless the file system is mounted.
3. For information about creating a file system on a striped logical volume, refer to [mklv.dita#mklv/mklv_description_fs_on_stripped_lv](#) in the **mklv** documentation.
4. To create a JFS2 file system on a logical volume, the minor number of the logical volume must be greater than 3071.

Flags

Item	Description
-b <i>Boot</i>	Names the program to be installed in block 0 of the new file system.
-i <i>i-Nodes</i>	Specifies the initial number of i-nodes on the file system. This flag is ignored when creating a journaled file system.
-l <i>Label</i>	Specifies the file system label for the new file system.
-o <i>Options</i>	Specifies a comma-separated list of virtual file system implementation-specific options.

The following options are specific to the Journaled File System (JFS):

Item	Description
-o ag ={ 8 16 32 64 }	Specifies the allocation group size in megabytes. An allocation group is a grouping of i-nodes and disk blocks similar to BSD cylinder groups. The default ag value is 8.
-o bf ={ true false }	Specifies a large file enabled file system. See JFS and large files for more information. If you do not need a large file enabled file system, set this option to false; this is the default. Specifying bf=true requires a fragment size of 4096 and compress=no .
-o frag ={ 512 1024 2048 4096 }	Specifies the JFS fragment size in bytes. A file system fragment is the smallest unit of disk storage that can be allocated to a file. The default fragment size is 4096 bytes.
-o compress ={ no LZ }	Specifies data compression. If you do not want data to be compressed, set this option to no. Selecting compression requires a fragment size of 2048 or less.
-o nbpi ={ 512 1024 2048 4096 8192 16384 32768 65536 131072 }	Specifies the number of bytes per i-node (nbpi). The nbpi is the ratio of file system size in bytes to the total number of i-nodes. The default nbpi value is 4096 bytes.

Notes:

- The **ag**, **bf**, **compress**, **frag**, and **nbpi** attributes are set at file system creation and cannot be changed after the file system is successfully created. The **size** attribute defines the minimum file system size, and you cannot decrease it after the file system is created.
- The root file system (/) cannot be compressed.
- Some **nbpi** values and allocation group sizes are mutually exclusive. See "Understanding JFS Size Limitations" for information.

The following options are specific to the Enhanced Journaled File System:

Item	Description
-o agblksize={ 512 1024 2048 4096 }	Specifies the Enhanced Journaled File System (JFS2) block size in bytes. A file system block is the smallest unit of disk storage that can be allocated to a file. The default block size is 4096 bytes.
-o isnAPSHOT={yes no}	Specifies whether the file system can support internal snapshots. Specifying yes enables the file system to support internal snapshots and v2 extended attributes. The resulting file system is not compatible with releases earlier than AIX 6.1.
-o name=mountpoint	Specifies the mount point for the file system.
-o log=LVName INLINE OUTLINE	<p>> You can set one of the following values for the log attribute:</p> <p>LVName Specifies the log logical volume name. The specified logical volume is the logging device for the new JFS2.</p> <p>INLINE Places the log device in the logical volume with the JFS2 file system. The log device defaults to 0.4% of the logical volume size if the logsize attribute is not specified. This value is set by default.</p> <p>OUTLINE Uses the existing logging device in the target volume group if available.</p> <p><</p>
-o logsize=Value	Specifies the size for an INLINE log in MB. Ignored if INLINE log not being used. Cannot be greater than 2047 MB and cannot be greater than 10% of the size of the file system.
-o ea={v1 v2}	Specifies the format to be used to store named extended attributes in the JFS2 file system. The v2 format provides support for scalable named extended attributes as well as support for NFS4 ACLs. The v1 format is compatible with prior releases of AIX. The default format is v1 .
-o efs={yes/no}	<p>Specifies encryption. Specifying yes enables encryption for the JFS2 file system.</p> <ul style="list-style-type: none"> • If the efs attribute is set to yes, the mkfs command automatically creates the JFS2 file system with the extended attribute format set to v2. The ea attribute is not required. • If the efs attribute is set to no, the mkfs command creates a file system that is not encrypted.
-o vix={yes/no}	<p>Specifies whether the file system can allocate i-node extents smaller than the default of 16 KB, if there are no contiguous 16 KB extents free in the file system. After a file system is enabled for small free extents, the file system cannot be accessed on AIX 5.1 or earlier releases.</p> <p>yes The file system can allocate variable-length i-node extents. This is the default value beginning with AIX 6.1.</p> <p>no The file system must use the default size of 16 KB for i-node extents. This has no effect if the file system already contains variable-length i-node extents.</p>

Item	Description
-o maxext= <i>Value</i>	Specifies the maximum size of a file extent in file system blocks. A zero value implies that the JFS2 default maximum should be used. Values less than 0 or exceeding maximum supported extent size of 16777215 are invalid.
> -o lff={yes/no} <	Specifies whether the file system supports large files and large file systems. By default, the maximum size of the JFS2 file system is 32 TB, and the maximum size of the file is 16 TB. If the value of the lff attribute is set to yes , the maximum potential size of the file system is 4 PB, and the maximum potential file size is 4 PB. Additionally, if the lff attribute is set to yes , the file system can only be mounted on AIX 7.3, or later.
	Notes:
	1. You cannot change the value of the lff attribute after you set it to yes .
	2. The lff attribute is only supported on a file system with an aggregate block size of 4096 bytes.

Note: The **agblksize** attribute is set at file system creation and cannot be changed after the file system is successfully created.

The **ea** attribute format is set at file system creation. The **chfs** command can be used to convert the extended attribute format from **v1** to **v2**, but the format cannot be converted back. The conversion is done in an on-demand manner such that any extended attribute or ACL writes cause the conversion for that file object to occur.

Item	Description
-p <i>Prototype</i>	Specifies the name of the prototype file when you create a JFS file system. Options specified on the command line override attributes in the prototype file.
-s <i>Size</i>	> Specifies the size of the file system. The value of the <i>Size</i> attribute can be specified in units of 512-byte blocks, megabytes (suffix M must be used), gigabytes (suffix G must be used), terabytes (suffix T must be used), or petabytes (suffix P must be used). See JFS and JFS2 for more information.
	Notes:
	1. To use the file system with size greater than 32 TB, the value of the lff attribute must be set to yes .
	2. The lff attribute is only supported on a file system with an aggregate block size of 4096 bytes.

|<

Notes:

- The volume group in which the file system resides defines a maximum logical volume size and also limits the file system size.
- The **-s** *Size* flag specifies the minimum file size and cannot be decreased after the file system has been successfully created.
- The maxext attribute is ignored in older releases even if the filesystem was created with it on a later release .

Item	Description
-v <i>VolumeLabel</i>	Specifies the volume label for the new file system.
-V <i>VfsName</i>	Specifies the virtual file system (VFS) type. The VFS must have an entry in the /etc/vfs file.

Restriction: The **mkfs** command prevents EFS File System enablement of the following File Systems (mount points) because the security infrastructure (kernel extensions, libraries, and so on) are not available when you start the system. The following list is of known File Systems (mount points) that you cannot use:

```
"/"  
"/usr"  
"/var"  
"/opt"
```

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 that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To specify the volume and file system name for a new file system, type:

```
mkfs -lworks -vvo1001 /dev/hd3
```

This command creates an empty file system on the `/dev/hd3` device, giving it the volume serial number `vo1001` and file system name `works`. The new file system occupies the entire device. The file system has a default fragment size (4096 bytes) and a default nbpi ratio (4096).

2. To create a file system with nondefault attributes, type:

```
mkfs -s 8192 -o nbpi=2048,frag=512 /dev/lv01
```

This command creates an empty 4 MB file system on the `/dev/lv01` device with 512-byte fragments and 1 i-node for each 2048 bytes.

3. To create a large file enabled file system, type:

```
mkfs -V jfs -o nbpi=131072,bf=true,ag=64 /dev/lv01
```

This creates a large file enabled JFS file system with an allocation group size of 64 megabytes and 1 i-node for every 131072 bytes of disk. The size of the file system will be the size of the logical volume `lv01`.

4. To create a file system with nondefault attributes, type:

```
mkfs -s 4M -o nbpi=2048, frag=512 /dev/lv01
```

This command creates an empty 4 MB file system on the `/dev/lv01` device with 512-byte fragments and one i-node for each 2048 bytes.

5. To create a JFS2 file system which can support NFS4 ACLs, type:

```
mkfs -V jfs2 -o ea=v2 /dev/lv01
```

This command creates an empty file system on the `/dev/lv01` device with **v2** format for extended attributes.

Files

Item	Description
<code>/etc/vfs</code>	Contains descriptions of virtual file system types.

Item	Description
<code>/etc/filesystems</code>	Lists the known file systems and defines their characteristics.

mkgroup Command

Purpose

Creates a new group.

Syntax

```
mkgroup [ -R load_module ] [ -a ] [ -A ] [ Attribute=Value ... ] Group
```

Description

The **mkgroup** command creates a new group. The *Group* parameter must be a unique string (whose length is administrator-configurable by way of the `chdev` command) and cannot be the **ALL** or **default** keywords. By default, the **mkgroup** command creates a standard group. To create an administrative group, specify the **-a** flag. You must be the root user or a user with GroupAdmin authorization to create an administrative group.

To create a group with an alternate Identification and Authentication (I&A) mechanism, the **-R** flag can be used to specify the I&A load module used to create the group. Load modules are defined in the `/usr/lib/security/methods.cfg` file.

You can use the System Management Interface Tool (SMIT) **smit mkgroups** fast path to run this command.

The **mkgroup** command always checks the target group registry to make sure the ID for the new account is unique to the target registry. The **mkgroup** command can also be configured to check all group 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 non-target 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 **mkgroup** command is unspecified when creating a new account in a third registry using the colliding ID values. The new account creation might succeed or fail depending 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 **mkgroup** command bypasses a remote registry if the remote registry is not reachable at the time the command is run.

If Encrypted File System (EFS) is enabled on the system, the **mkgroup** command updates the `/etc/security/group` file with EFS attributes (default values are added if you do not specify the attributes on

the command line). If you do not specify `efs_keystore_access=none`, the **mkgroup** command creates the group keystore if at least one of the users has a keystore.

If the **mkgroup** command returns with the return code of 3, the keystore for the group is not created, but the **mkgroup** command creates the group.

Note: You can later create the group keystore by using the **efskeymgr** command.

Restrictions on Creating Group Names

To prevent login inconsistencies, you should avoid composing group names entirely of uppercase alphabetic characters. While the **mkgroup** command supports multibyte group names, it is recommended that you restrict group names to characters with the POSIX portable filename character set.

To ensure that your user database remains uncorrupted, you must be careful when naming groups. Group names must not begin with a - (dash), + (plus sign), @ (at sign), or ~ (tilde). You cannot use the keywords **ALL** or **default** in a group name. Additionally, do not use any of the following characters within a group-name string:

Item	Description
:	Colon
"	Double quote
#	Pound sign
,	Comma
=	Equal sign
\	Back slash
/	Slash
?	Question mark
'	Single quote
`	Back quote

Finally, the *Name* parameter cannot contain any space, tab, or new-line characters.

Flags

Item	Description
-a	Creates an administrative group. Only the root user can use this flag.
-A	Sets the group administrator to the person who invoked the mkgroup command.
-R <i>load_module</i>	Specifies the loadable I&A module used to create the user.
<i>Attribute=Value</i>	Initializes a group with a specific attribute. See the chgroup command for more information about the group attributes.

Exit Status

This command returns the following exit values:

Item	Description
0	The command runs successfully and all requested changes are made.
>0	An error occurred. The printed error message lists 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.

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 must also have the **aix.security.group.change** authorization. If Encrypted File System (EFS) is enabled on the system, the role must also have the **aix.security.efs** authorization to create the group keystore.

Files Accessed:

Mode	File
rw	/etc/passwd
rw	/etc/security/user
rw	/etc/security/limits
rw	/etc/security/environ
rw	/etc/group
rw	/etc/security/group
r	/usr/lib/security/mkuser.default
x	/usr/lib/security/mkuser.sys

Auditing Events:

Event	Information
USER_Create	user

Limitations

Creating a group may not be supported by all loadable I&A modules. If the loadable I&A module does not support creating a group, an error is reported.

Examples

1. To create a new group account called `finance`, type:

```
mkgroup finance
```

2. To create a new administrative group account called `payroll`, type:

```
mkgroup -a payroll
```

Only the root user can issue this command.

3. To create a new group account called `managers` and set yourself as the administrator, type:

```
mkgroup -A managers
```

4. To create a new group account called `managers` and set the list of administrators to `steve` and `mike`, type:

```
mkgroup adms=steve,mike managers
```

The users `steve` and `mike` must already exist on the system.

5. To create a new group that is a LDAP I&A loadable module user, type:

```
mkgroup -R LDAP monsters
```

Files

Item	Description
<u>/usr/bin/mkgroup</u>	Contains the mkgroup command.
<u>/etc/group</u>	Contains the basic attributes of groups.
<u>/etc/security/group</u>	Contains the extended attributes of groups.
<u>/etc/passwd</u>	Contains basic user information.
<u>/etc/security/passwd</u>	Contains password information.

mkhosts Command

Purpose

Generates the host table file.

Syntax

```
/usr/sbin/mkhosts [ -v ] HostFile
```

Description

The **mkhosts** command can be used to generate a hashed host database, using the filename specified by the *HostFile* parameter. It is not used if name resolution is performed by the **named** daemon. The host file is usually the **/etc/hosts** file, and in any case must be in the same format as the **/etc/hosts** file.

The **mkhosts** command generates database files named **hostfile.pag** and **hostfile.dir**. Updates to these files are built in a set of temporary files named **hostfile.new.pag** and **hostfile.new.dir**. The temporary files are copied into the database files only if the **hostfile.new.pag** and **hostfile.new.dir** files are built without errors.

The host file is used by one version of the **gethostbyaddr** and **gethostbyname** library routines for name resolution.

Note: The version of the **gethostbyaddr** and **gethostbyname** library routines on this operating system do not support the **hostfile.pag** and **hostfile.dir** files.

After creating the host file, you can edit it to include the desired host entries.

Flags

It	Description
----	-------------

m	
----------	--

-v	Lists each host as it is added to the host file specified by the <i>HostFile</i> parameter.
-----------	---

Examples

Use the following command to generate the **/etc/hosts.pag** and **/etc/hosts.dir** files:


```
mkhosts /etc/hosts
```

This command creates two host files called **/etc/hosts.pag** and **/etc/hosts.dir**.

Files

Item	Description
hostfile.pag	One of two files containing the real database for name resolution.
hostfile.dir	One of two files containing the real database for name resolution.
hostfile.new.pag	One of two files containing the temporary database for name resolution.
hostfile.new.dir	One of two files containing the temporary database for name resolution.

mkiba Command

Purpose

Configures an IPv4 address or IPv6 address over the InfiniBand interface.

Syntax

```
mkiba { -i Interface -a address | -v address6 -A ib_adapter -p ib_port [ -P P_KEY ] [ -V ipv6prefix ] [ -m subnet_mask ] [ -S state ] [ -M mtu ] [ -q queue_pair_size ] [ -Q Q_KEY ] [ -k superpacket ] }
```

Description

The **mkiba** command sets the minimal values required for using the IPv4 address or IPv6 address over the InfiniBand interface on a host machine. These values are written to the configuration database. This command also parses the information and verifies if the parameters are correct. If the interface is not defined, this command defines it, and calls the **chdev** command to configure it.

The following list details the functions of the **mkiba** command:

- Defining the interface name in the configuration database if it is not already defined
- Setting the host name in both the configuration database and the running machine
- Setting the IP address of the interface in the configuration database
- Setting the subnetwork mask, if applicable
- Setting the prefix length, if applicable
- Setting InfiniBand-specific parameters (such as host channel adapter (HCA), port, and so on)

You can use the **smit mkinetib** fast path in the System Management Interface Tool (SMIT) to run this command.

At least one IPv4 address or IPv6 address must be included to configure the interface.

Use the **ifconfig** *ibX* to check the state of the interface after the configuration.

The **ifconfig** command indicates that if the InfiniBand multicast group is pending, then there is an adapter malfunction or the physical port is down. If there is an adapter malfunction, you must perform manual HCA device driver reconfiguration to recover the InfiniBand stack. If the port is down, you must check the cabling and switches. If multicast is pending, check the InfiniBand Subnet Manager for errors. To troubleshoot the InfiniBand interface issues, use the **ibstat** command.

Flags

Item	Description
-A <i>ib_adapter</i>	Specifies the HCA. For example, iba0.
-a <i>address</i>	Sets the Internet address of the host. Specify the address in dotted decimal notation. Each network interface on the host must have a unique Internet address. For example, a standard format for setting the Internet address is 127.10.31.2.
-i <i>Interface</i>	Specifies a particular InfiniBand interface. For example, ib0.
-k <i>superpacket</i>	Superpacket is a proprietary algorithm that allows the interface to receive large MTU (Maximum Transmit Unit) packets and fragment them in the interface layer to fit in the maximum MTU of the adapter. Enabling the superpacket automatically changes the <i>tcp_sendspace</i> attribute which specifies the number of bytes of data that the sending application can buffer in the kernel before the application is blocked on a send call, and the <i>tcp_recvspace</i> attribute which specifies the number of bytes of data that the receiving system can buffer in the kernel on the receiving sockets queue, with optimized values for the superpacket use.
-M <i>mtu</i>	Specifies the MTU for the interface. The IPv4 address or IPv6 address uses this MTU to fragment the packets.
-m <i>subnet_mask</i>	Specifies the mask that the gateway must use to determine the appropriate subnetwork for routing. The subnet mask is a set of 4 bytes. The subnet mask consists of high bits (1s) corresponding to the bit positions of the network and subnetwork address, and low bits (0s) corresponding to the bit positions of the host address.
-P <i>P_KEY</i>	Specifies the partition key. The common partition keys are 0xFFFF and 0x7FFF.
-p <i>ib_port</i>	Specifies the HCA ports that you must use to configure the InfiniBand interface.
-q <i>queue_pair_size</i>	Specifies the size of the software queue. The range is from 256 to 32000.
-Q <i>Q_KEY</i>	Specifies the multicast qkey used to create a broadcast multicast group if there is no group previously created in the Subnet Manager. The common keys are 1, 0, and 0x1E.
-S <i>state</i>	Specifies whether the interface is active. When an interface is marked as inactive, any attempt to transmit messages through that interface fails. This action does not automatically disable routes using the interface.
-V <i>ipv6prefix</i>	Specifies the number of high-order bits used by routing protocols. The prefix is usually denoted following the IPv6 address and a slash (/). For example, the notation ff12::/16 represents a 16-bit prefix with a value of 1111111100010010.
-v <i>address6</i>	Specifies the IPv6 address. The address is a 128-bit address represented as eight 16-bit integers separated by colons. Each integer is represented by 4 hex digits. Leading zeros can be skipped, and consecutive null 16-bit integers can be replaced by two colons (one per address). For example, fe80:abcd:0000:0000:0000:0000:0260:8c2e:00a4.

Example

1. To set the required values to configure the IPv4 address over the InfiniBand interface, enter:

```
mkiba -a 192.9.200.9 -i ib0 -A iba0 -p 1 -P -1 -q 4000 -M 2044 -m 255.255.255.0
```

2. To set the required values to configure the IPv6 address over the InfiniBand interface, enter:

```
mkiba -v fe80::2:c903:1:1b40 -i ib0 -A iba0 -p 1 -P -1 -q 4000 -M 2044
```

mkinstall Command

Purpose

Creates software packages in **installp** format.

Syntax

```
mkinstall [ -d BaseDirectory ] [ -T TemplateFile ]
```

Description

The **mkinstall** command allows users to create their own software packages for AIX. Packages created with the **mkinstall** command are in **installp** format and can be installed or removed with the **installp** command.

Files to be packaged by the **mkinstall** command must be in a directory structure such that the location of the file relative to the root build directory is the same as the destination of the file after installation. For example, if **/usr/bin/somecommand** is to be installed through a **mkinstall** package, the *somecommand* parameter must be in the *buildroot/usr/bin* directory when the **mkinstall** command is run.

After the contents of a package are located in the correct directory structure, the **mkinstall** command prompts for basic package data. This data includes the package name, requisites, descriptions of files to be packaged, and more. The **mkinstall** command will then generate a template file based on responses given by the user. To prevent command-line prompting, template files can be created and edited directly by the user and passed to the **mkinstall** command with the **-T** flag.

Flags

Item	Description
-d <i>BaseDirectory</i>	Specifies the root build directory containing the files to be packaged. If not specified, the current working directory is used.
-T <i>TemplateFile</i>	Specifies the full path name of the template file to be passed to the mkinstall command. If not specified, the mkinstall command prompts for package information and creates a new template file based on user responses.

Note: Do not use the **.info** directory located in the *BaseDirectory* to store a template file. The template file may be removed when you run the **mkinstall** command.

Examples

This example demonstrates how to package the file **/usr/bin/foo** using the **/tmp/packages** directory as the root build directory.

First, create the directory structure by typing the following at the command line:

```
mkdir -p /tmp/packages/usr/bin
```

Then, type the following to create the file **/usr/bin/foo**:

```
touch /tmp/packages/usr/bin/foo
```

Then, type the following to create the package using the **mkinstallp** command:

```
mkinstallp -d /tmp/packages
```

For more examples, see the **/usr/lpp/bos/README.MKINSTALLP** file.

Files

Item	Description
/usr/sbin/mkinstallp	Contains the mkinstallp command.

mkiscsi Command

Purpose

Adds iSCSI target data.

Syntax

```
mkiscsi -l AdapterName -g static -t TargetName -n PortNumber -i IPaddress [-p password] [-u UserName]
```

```
mkiscsi -l AdapterName -g auto -t TargetName -p password [-u UserName]
```

```
mkiscsi -l AdapterName -g group -f FileName
```

Description

The **mkiscsi** command adds 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	Description
-f FileName	Specifies the filename from which iSCSI target information will be read and then placed into ODM.
-g group	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.

Item	Description
-l <i>AdapterName</i>	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 <i>NewPortNumber</i>	Specifies the port number on which the iSCSI target is accessed. The default port number is 3260.
-p <i>password</i>	Specifies the new password for this iSCSI target.
-t <i>TargetName</i>	Specifies the iSCSI target name (for example, <code>iqn.sn9216.iscsi-hw1</code>).
-u <i>UserName</i>	Specifies the Challenge Handshake Authentication Protocol (CHAP) user name that can be used during login. This value is used only if the CHAP password is supplied. If the CHAP user name is not specified, the iSCSI initiator uses the local iSCSI Qualified Name (as specified by the initiator_name attribute of the iSCSI device) for the CHAP user name.

Exit Status

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

Security

The `mkiscsi` command is executable only by root.

Examples

1. To add one statically configured iSCSI target, enter the following command:

```
mkiscsi -l ics0 -g static -t iqn.sn1234.iscsi_hw1
```

2. To add all the entries from the file `/etc/iscsi/targetshw`, enter the following command:

```
mkiscsi -l ics0 -g static -f /etc/iscsi/targetshw
```

Location

`/usr/sbin/mkiscsi`

Files

Item	Description
<code>src/bos/usr/sbin/iscsia</code>	Contains the common source files from which the iSCSI commands are built.

mkitab Command

Purpose

Makes records in the **/etc/inittab** file.

Syntax

```
mkitab [ -i Identifier ] { [ Identifier ] : [ RunLevel ] : [ Action ] : [ Command ] }
```

Description

The **mkitab** command adds a record to the **/etc/inittab** file. The *Identifier:RunLevel:Action:Command* parameter string specifies the new entry to the **/etc/inittab** file. You can insert a record after a specific record using the **-i** *Identifier* flag. The command finds the field specified by the *Identifier* parameter and inserts the new record after the one identified by the **-i** *Identifier* flag.

Parameters

The *Identifier:RunLevel:Action:Command* parameter string specifies the record in the **/etc/inittab** file, as follows:

Item	Description
<i>Identifier</i>	A 14-character parameter that uniquely identifies an object. The <i>Identifier</i> must be unique. If the <i>Identifier</i> is not unique, the command is unsuccessful. The <i>Identifier</i> cannot be changed; if you try to change it, the command is unsuccessful.
<i>RunLevel</i>	A 20-character parameter defining the run levels in which the <i>Identifier</i> can be processed. Each process started by the init command can be assigned one or more run levels in which it can be started.
<i>Action</i>	A 20-character parameter that informs the init command how to process the <i>Command</i> parameter that you specify. The init command recognizes the following actions: respawn If the process identified in this record does not exist, start the process. If the process currently exists, do nothing and continue scanning the /etc/inittab file. wait When the init command enters the run level specified for this record, start the process and wait for it to stop. While the init command is in the same run level, all subsequent reads of the /etc/inittab file ignore this object. once When the init command enters the run level specified for this record, start the process, do not wait for it to stop and when it does stop do not restart the process. If the system enters a new run level while the process is running, the process is not restarted. boot Read this record only when the system boots and reads the /etc/inittab file. The init command starts the process. Do not wait for the process to stop and when it does stop, do not restart the process. The run level for this process should be the default, or it must match the run level specified by the init command at startup time.

Item	Description
bootwait	Read this record only when the system boots and reads the /etc/inittab file. The init command starts the process. Wait for it to stop, and when it does stop, do not restart the process.
powerfail	Start the process identified in this record only when the init command receives a SIGPWR power fail signal.
powerwait	Start the process identified in this record only when the init command receives a SIGPWR power fail signal, and wait until it stops before continuing to process the /etc/inittab file.
off	If the process identified in this record is currently running, send the warning signal SIGTERM and wait 20 seconds before sending the SIGKILL kill signal. If the process is nonexistent, ignore this line.
hold	When the process identified in this record is terminated, do not start a new one. The hold action can only be activated by the phold command.
ondemand	Functionally identical to respawn . If the process identified in this record does not exist, start the process. If the process currently exists, do nothing and continue scanning the /etc/inittab file. Specify this action to perform the respawn action when using a , b , or c run levels.
initdefault	A line with this action is processed only when the init command is originally invoked. The init command uses this line to determine which run level to originally enter. The command does this by taking the highest run level specified in the <i>RunLevel</i> parameter and using that as the command's initial state. If the <i>RunLevel</i> parameter is empty, its value is interpreted as 0123456789, and the init command enters a run level of 9 . If the init command does not find an initdefault line in the inittab file, it requests an initial run level from the operator at initial program load (IPL) time.
sysinit	Start the process identified in this record before the init command tries to access the console. For example, you might use this to initialize devices.

Command A 1024-character field specifying the shell command.



Attention: To avoid possible corruption of system files, the **stdin**, **stdout**, and **stderr** files must be specified in the *Command* parameter with redirection, or they must be explicitly opened by the program being run by the command line.

Flags

Item	Description
-i Identifier	Specifies which record in the /etc/inittab file the new record follows.

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 add a new record to the **/etc/inittab** file, telling the **init** command to handle a login on tty2, type:

```
mkitab "tty002:2:respawn:/usr/sbin/getty /dev/tty2"
```

2. To add a new record to the **/etc/inittab** file, telling the **init** command to execute the **/etc/rc.tcpip** file after the **/usr/sbin/srcmstr** file is started, type:

```
mkitab -i srcmstr "rctcpip:2:wait:/etc/rc.tcpip > /dev/console"
```

3. To add a new record to the **/etc/inittab** file, telling the **init** command to execute the **/etc/rc** file and send its output to the boot log, type:

```
mkitab ((rc:2:wait:/etc/rc 2>&1 | alog -tboot > /dev/console))
```

Files

Item	Description
/etc/inittab	Contains the mkitab command.

mkkeyserv Command

Purpose

Uncomments the entry in the **/etc/rc.nfs** file for the **keyserv** daemon and invokes the daemon by using the **startsrc** command.

Syntax

```
/usr/sbin/mkkeyserv [ -I | -B | -N ]
```

Description

The **mkkeyserv** command uncomments the entry in the **/etc/rc.nfs** file for the **keyserv** daemon. The **mkkeyserv** command starts the daemon by using the **startsrc** command.

You can use the System Management Interface Tool (SMIT) **smit mkkeyserv** fast path to run this command.

Flags

Item	Description
-I	Uncomments the entry in the /etc/rc.nfs file to start the keyserv daemon on the next system restart.
-B	Uncomments the entry in the /etc/rc.nfs file to start the keyserv daemon and uses the startsrc command to start the keyserv daemon. This flag is the default.

Item Description

- N** Uses the **startsrc** command to start the **keyerv** daemon. This flag does not change the **/etc/rc.nfs** file.

Examples

To modify the **/etc/rc.nfs** file to invoke the **keyerv** daemon on the next system restart, enter:

```
mkkeyerv -I
```

Files

Item	Description
/etc/rc.nfs	Contains the startup script for the NFS and NIS daemons.

mkkrb5clnt Command

Purpose

Configures a Kerberos client.

Syntax

To configure Kerberos against IBM Network Authentication Service only:

```
mkkrb5clnt -h | [ -c KDC -r Realm -s Server -U [ -a Admin ] -d Domain [ -A ] [ -i Database ] [ -K ] [ -T ] [ -t ticket_lifetime ] [ -n renew_lifetime ] [ -l {ldapserver | ldapserver:port} ]
```

To configure Kerberos against non-kadmind services:

```
mkkrb5clnt -h | -c KDC -r Realm -s Server -d Domain [ -i Database ] [ -K ] [ -t ticket_lifetime ] [ -n renew_lifetime ] -D [ -l {ldapserver | ldapserver:port} ] | -U
```

Description

This command configures the Kerberos client. The first part of the command reads realm name, KDC, VDB path, and domain name from the input and generates a **krb5.conf** file.

Item	Description
/etc/krb5/krb5.conf:	Values for realm name, Kerberos admin server, and domain name are set as specified on the command line. Also updates the paths for default_keytab_name , kdc , and kadmin log files.

If DCE is not configured, this command creates a link to **/etc/krb5/krb5.conf** from **/etc/krb5.conf**.

The command also allows you to configure root as admin user, configure integrated Kerberos authentication, and configure Kerberos as default authentication scheme.

For integrated login, the **-i** flag requires the name of the database being used. For LDAP, use the load module name that specifies LDAP. For local files, use the keyword files.

Item	Description
Standard Output	Consists of information messages when the -h flag is used.
Standard Error	Consists of error messages when the command cannot complete successfully.

Flags

Item	Description
-a <i>Admin</i>	Specifies the principal name of the Kerberos server admin.
-A	Specifies root to be added as a Kerberos administrative user.
-c <i>KDC</i>	Specifies the KDC server.
-d <i>Domain</i>	Specifies the complete domain name for the Kerberos client.
-D	Specifies Kerberos against non-kadmind services.
-h	Specifies that the command is only to display the valid command syntax.
-i <i>Database</i>	Configures integrated Kerberos authentication.
-K	Specifies Kerberos to be configured as the default authentication scheme.
-l <i>ldapservers / ldapservers:port</i>	<p>For servers, specifies the LDAP directory used to store the Network Authentication Service principal and policy information.</p> <p>For clients, specifies the LDAP directory server to use for Administration server and KDC discovery using LDAP. If the -l flag is used, then the KDC and server flags are optional. If the -l option is not used, the KDC and server flags must be specified. The port number can optionally be specified.</p> <p>For clients and servers, the port number can optionally be specified. If the port number is not specified, the client connects to the default LDAP server port 389 or 636 for SSL connections.</p> <p>Note: Only the client configuration is updated.</p>
-n <i>renew_lifetime</i>	Specifies the client-specific time to generate a renewable ticket if the server supports it. By default, the ticket is nonrenewable. The <i>renew_lifetime</i> parameter value is composed of four numeric values that are delimited by colons.
-r <i>Realm</i>	Specifies the full realm name for which the Kerberos client is to be configured.
-s <i>Server</i>	Specifies the fully qualified host name for Kerberos admin server.
-t <i>ticket_lifetime</i>	Specifies the client-specific ticket lifetime for received tickets if the server supports it. If you do not specify the flag, the server sets the ticket lifetime. The <i>ticket_lifetime</i> parameter value is composed of four numeric values that are delimited by colons.
-T	Specifies the flag to acquire server admin TGT based admin ticket.
-U	Undo the setup from the previous configuration command.

Exit Status

Failure of this command to execute successfully may result in incomplete client configuration.

Item	Description
0	Indicates the successful completion of the command.
1	Indicates that an error occurred.

Security

A user with the **aix.security.kerberos** authorization is authorized to use this command.

Examples

1. To display the command syntax, enter the following command:

```
mkkrb5clnt -h
```

2. To configure **testbox.austin.ibm.com** as a client to **sundial.austin.ibm.com** where KDC is also running on **sundial.austin.ibm.com**, enter the following command:

```
mkkrb5clnt -c sundial.austin.ibm.com -r UD3A.AUSTIN.IBM.COM \  
-s sundial.austin.ibm.com -d austin.ibm.com
```

3. To configure **testbox.austin.ibm.com** as the client, make root as the server admin, configure integrated login, configure Kerberos as default authentication scheme, enter the following command:

```
mkkrb5clnt -c sundial.austin.ibm.com -r UD3A.AUSTIN.IBM.COM \  
-s sundial.austin.ibm.com -d austin.ibm.com \  
-A -i files -K -T
```

4. To configure **testbox.austin.ibm.com** as the client against a non-AIX machine, enter the following command:

```
mkkrb5clnt -c non-aix.austin.ibm.com -r NON-AIX.AUSTIN.IBM.COM \  
-s non-aix.austin.ibm.com -d austin.ibm.com -D
```

5. To configure **testbox.austin.ibm.com** as the client against a non-AIX machine with the ticket lifetime of 1 day, 2 hours, 3 minutes, and 4 seconds, and the renew lifetime of 5 days, 6 hours, 7 minutes, and 8 seconds, enter the following command:

```
mkkrb5clnt -c non-aix.austin.ibm.com -r NON-AIX.AUSTIN.IBM.COM \  
-s non-aix.austin.ibm.com -d austin.ibm.com -D \  
-t 1:2:3:4 -n 5:6:7:8
```

Files

Item	Description
<code>/usr/krb5/sbin</code>	Contains the mkkrb5clnt command.

mkkrb5srv Command

Purpose

Configures a Kerberos server.

Syntax

```
mkkrb5srv -h | [ -r Realm -d Domain -a AdminName ] [ -l ldapserver | ldapserver:port ] [ -u ldap_DN ] [ -p ldap_DN_pw ] [ -f {keyring | keyring:entry_dn} ] [ -k keyring_pw ] [ -b bind_type ] [ -m masterkey_location ] [ -U ]
```

Description

The **mkkrb5srv** command configures the Kerberos server. This command creates the **kadm5.acl** file, the **kdc.conf** file, and the Kerberos database. It also adds the administrator to the database and updates the **/etc/inittab** file with Kerberos daemons. This command does the initial configuration once the variables are set. They can be modified by editing the following files:

Item	Description
/etc/krb5/krb5.conf:	Values for realm name, Kerberos admin server, and domain name are set as specified on the command line. Also updates the paths for default_keytab_name , kdc , and kadmin log files.
/var/krb5/krb5kdc/kdc.conf	This command sets the value for kdc_ports . Paths for database name, admin_keytab , acl_file , dict_file , key_stash_file . Values for kadmin_port , max_life , max_renewable_life , master_key_type , and supported_encetypes .
/var/krb5/krb5kdc/kadm5.acl	Sets up the acls for admin, root, and host principals.

If DCE is not configured, this command creates a link to **/etc/krb5/krb5.conf** from **/etc/krb5.conf**.

Item	Description
Standard Output	Consists of information messages when the -h flag is used.
Standard Error	Consists of error messages when the command cannot complete successfully.

Flags

Item	Description
-a <i>AdminName</i>	Specifies the Kerberos Principal name for the administrator.
-b <i>bind_type</i>	Specifies the LDAP bind type. Supported values are the following: <ul style="list-style-type: none"> • simple • cram-md5 • external These bind types can be specified in either upper case or lower case.
-d <i>Domain</i>	Specifies the domain name for the Kerberos realm.
-f <i>{keyring keyring:entry_dn}</i>	Specifies the LDAP keyring database file name if you are using SSL communication.
-h	Specifies that the command is only to display the valid command syntax.
-k <i>keyring_pw</i>	Specifies the password for the LDAP keyring database file. If not specified, SSL uses the password that is encrypted in the appropriate password stash file.
-l <i>ldapserverserver ldapserverserver:port</i>	For servers, specifies the LDAP directory used to store the Network Authentication Service principal and policy information. For clients, specifies the LDAP directory server to use for Administration server and KDC discovery using LDAP. If the -l flag is used, then the KDC and server flags are optional. If the -l option is not used, the KDC and server flags must be specified. The port number can optionally be specified. For clients and servers, the port number can optionally be specified. If the port number is not specified, the client connects to the default LDAP server port 389 or 636 for SSL connections.
	Note: Only the client configuration is updated.

Item	Description
-m <i>masterkey_location</i>	Specifies the fully qualified file name for storing the master key in the local file system when using LDAP to store data. Note: This flag is only for use with the LDAP directory.
-p <i>ldap_DN_pw</i>	Specifies the password for the entry being used for the <i>ldap_DN_pw</i> .
-r <i>Realm</i>	Specifies the realm for which the Kerberos server is to be configured.
-u <i>ldap_DN</i>	Specifies the LDAP entry to be used as the <i>ldap_DN</i> . Note: With external bind, the -u and -p flags are not required, and the values come from the certificate.
-U	Undo the setup from the previous configuration command.

Exit Status

Failure of this command to execute successfully results in incomplete server configuration.

Item	Description
0	Indicates the successful completion of the command.
1	Indicates that an error occurred.

Security

A user with the **aix.security.kerberos** authorization is authorized to use this command.

Examples

1. To display the command syntax, type:

```
mkkrb5srv -h
```

2. To configure sundial as a Kerberos server, type:

```
mkkrb5srv -r UD3A.AUSTIN.IBM.COM -d austin.ibm.com
```

Files

Item	Description
/usr/sbin/mkkrb5srv	Contains the mkkrb5srv command.

mklost+found Command

Purpose

Creates a lost and found directory for the **fsck** command.

Syntax

mklost+found

Description

The **mklost+found** command creates a lost and found directory in the current directory. A number of empty files are created within the lost and found directory and then removed so that there are empty slots for the **fsck** command. The **fsck** command reconnects any orphaned files and directories by placing them in the lost and found directory with an assigned i-node number. The **mklost+found** command is not normally needed, since the **fsck** command automatically creates the lost and found directory when a new file system is created.

Examples

To make a lost+found directory for the **fsck** command, enter:

```
mklost+found
```

Files

Item	Description
<code>/usr/sbin/mklost+found</code>	Contains the mklost+found command.

mklpcmd Command

Purpose

Defines a new least-privilege (LP) resource to the resource monitoring and control (RMC) subsystem and specifies user permissions.

Syntax

```
mklpcmd [-n host] [-l] [-c 0 | 1 | 2 | 3] [-R RunCmdName] [-s FilterScript] [-A FilterArg] [-h] [-TV] resource_name command_path [ ID perm ] ...
```

Description

The `mklpcmd` command defines a new LP resource to the resource monitoring and control (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). Specify the LP resource using the *resource_name* parameter. The *command_path* parameter specifies the command or script that could be run with LP access. Specify the complete path name of the command or the script. If *command_path* exists when a resource is created, the LP resource manager calculates the CheckSum and assigns the CheckSum attribute value. If *command_path* does not exist, the LP resource manager assigns 0 as the CheckSum attribute value.

Use the `-l` flag to lock the LP resource. The resource must be unlocked before it can be deleted. Use the `-c` flag to specify the control settings of the resource.

You can also use the `mklpcmd` command to specify permissions for users when you are creating a resource. To do this, you need to have administrator permission on the resources. Administrator permission gives you the ability to set and edit permissions. You can specify multiple user IDs and permissions with this command. See the Examples section for more information.

This command runs on any node. In a management domain or a peer domain, use the `-n` flag to define the LP resource on the node that is specified by *host*. Otherwise, this command runs on the local node.

Flags

-n *host*

Specifies the node in the domain on which the LP resource is to be defined. By default, the LP resource is defined 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 `mk1pcmd` command runs once for the first valid scope that the LP resource manager finds.

-l

Defines the new LP resource as locked so that it cannot be changed accidentally. The resource cannot be removed from the RMC subsystem until the Lock attribute is unset.

If you do not specify this flag, the new resource is not locked. This is the default.

-c 0 | 1 | 2 | 3

Sets the `ControlFlags` attribute, which is used to specify the control features for an LP command. If `ControlFlags` is not specified, it is set to 1 by default. Use this flag to specify one of these values:

0

Does not validate the `Checksum` value.

1

Does not validate the `Checksum` value. This is the default.

2

Validates the `Checksum` value.

3

Validates the `Checksum` value.

When an attempt is made to run the LP resource using the `run1pcmd` command, the value of the `ControlFlags` attribute determines which checks are performed before running the command represented by the resource.

In this release of RSCT, the `ControlFlags` attribute value specifies whether the `Checksum` value is to be validated.

In previous releases of RSCT, the `ControlFlags` attribute value also specified whether the presence of certain characters in the input arguments to `run1pcmd` were to be disallowed. Checking for these characters is no longer necessary.

To maintain compatibility with LP resources that were defined in previous releases of RSCT, the `ControlFlags` attribute values, with respect to validating the `Checksum` value, have remained the same. Consequently, values 0 and 1 indicate that the `Checksum` value is not to be validated, and values 2 and 3 indicate that the `Checksum` value is to be validated.

-R *RunCmdName*

Specifies the `RunCmdName` value for this resource, which will be used as a parameter of the `run1pcmd` command.

-s *script_path*

Specifies the fully-qualified path of the filter script.

-A *argument*

Specifies a string of arguments to be passed to the filter script.

-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_name***

Is the name or identifier of the LP resource that is to be defined to the RMC subsystem.

command_path

Is the complete, fully-qualified path name of the command or script.

ID perm ...

Specifies permissions for users when you are creating a resource. This parameter is optional.

ID

Specifies the user identity for the ACL entry. See the *User identities* section of the `lpac1` information for the valid forms of this parameter.

perm

Specifies the user permissions for the ACL entry. This parameter can consist of a combination of any of the following values:

r

Read permission (consists of the *q*, *l*, *e*, and *v* permissions)

w

Write permission (consists of the *d*, *c*, *s*, and *o* permissions)

a

Administrator permission

x

Execute permission

q

Query permission

l

Enumerate permission

e

Event permission

v

Validate permission

d

Define and undefine permission

c

Refresh permission

s

Set permission

o

Online, offline, and reset permission

0

No permission

See the *User permissions* section of the `lpac1` information for descriptions of these permissions.

Security

- To run the `mk1pcmd` command with one or more *ID:perm* parameters, you need:
 - read and write permission in the Class ACL of the `IBM.LPCmds` resource class.
 - read and administrator permission in the Resource Initial ACL.

As an alternative, the Resource Initial ACL can direct the use of the Resource Shared ACL if these permissions exist in the Resource Shared ACL.

- To run the `mk1pcmd` command with no `ID:perm` parameters, you need write permission in the Class ACL of the `IBM.LPCmds` 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 create an LP resource called LP1 that points to a command called `/tmp/user1/lpcmd1` on the local node, enter:

```
mk1pcmd LP1 /tmp/user1/lpcmd1
```

2. To create an LP resource called LP2 that points to a command called `/tmp/my_command1` on nodeB in the management domain, enter:

```
mk1pcmd -n nodeB LP2 /tmp/my_command1
```

3. To create an LP resource called lp3 with `ControlFlags` set to 3 (which means verify the `Checksum` value), enter:

```
mk1pcmd -c 3 LP3 /tmp/cmd_lp3
```

4. To create an LP resource called lp4 that points to `/tmp/testscript`, has a `RunCmdName` value of `test`, a `FilterScript` value of `/tmp/filterscr`, and filter arguments `node1` and `node2`, enter:

```
mk1pcmd -R test -f /tmp/filterscr -A "node1,node2" lp4 /tmp/testscript
```

5. To create an LP resource called lp5 that points to `/usr/bin/mkrsrc` and gives users `user1@LOCALHOST` and `user2@LOCALHOST` read, write, and execute permission, enter:

```
mk1pcmd lp5 /usr/bin/mkrsrc user1@LOCALHOST rwx user2@LOCALHOST rwx
```

Location

`/opt/rsct/bin/mk1pcmd`

Contains the `mk1pcmd` command

mklv Command

Purpose

Creates a logical volume.

Syntax

```
mklv [ -a position ] [ -b badblocks ] [ -c copies ] [ -C stripewidth ] [ -d schedule ] [ -R PreferredRead ] [ -e range ] [ -i ] [ -L label ] [ -m mapfile ] [ -o y | n ] [ -r relocate ] [ -s strict ] [ -t type ] [ -T O ] [ -u upperbound ] [ -v verify ] [ -w mirrorwriteconsistency ] [ -x maximum ] [ -y newlogicalvolume | -Y prefix ] [ -S stripsize ] [ -U userid ] [ -G groupid ] [ -P modes ] [ -p copyn=mirrorpool ] [ -O y | n ] >| [ -k y | n ] |< volume group number [ physicalvolume ... ]
```

Description

The `mklv` command creates a new logical volume within the *volume group*. For example, all file systems must be on separate logical volumes. The `mklv` command allocates the number of logical partitions to the

new logical volume. If you specify one or more physical volumes with the *physicalvolume* parameter, only those physical volumes are available for allocating physical partitions; otherwise, all the physical volumes within the volume group are available.

The default settings provide the most commonly used characteristics, but use flags to tailor the logical volume to the requirements of your system. After a logical volume is created, its characteristics can be changed with the **chlv** command.

The default allocation policy is to use a minimum number of physical volumes per logical volume copy, to place the physical partitions belonging to a copy as contiguously as possible, and then to place the physical partitions in the desired region specified by the **-a** flag. Also, by default, each copy of a logical partition is placed on a separate physical volume.

The **-m** flag specifies exact physical partitions to be used when creating the logical volume.

The **-U**, **-G**, and **-P** flags can be used to set the ownership, group, and permissions, respectively, of the logical volume device special files. Only root users can set these values. For scalable and big vg format volume groups that are exported, specify the **-R** flag with the `importvg` command to restore these values upon import.

You can specify logical volumes sizes in 512 Blocks/KB/MB/GB when using the **mklv** command. The logical volumes sizes must be integer values.

Physical partitions are numbered starting at the outermost edge with number one.

Note:

1. Changes made to the logical volume are not reflected in the file systems. To change file system characteristics use the **chfs** command.
2. Each logical volume has a control block. This logical volume control block is the first few hundred bytes within the logical volume. Care has to be taken when reading and writing directly to the logical volume to allow for the control block. Logical volume data begins on the second 512-byte block.
3. To use this command, you must either have root user authority or be a member of the **system** group.
4. When creating a striped logical volume using the **-S** flag, you must specify two or more physical volumes or use the **-C** or **-u** flag.
5. When creating a striped logical volume, the number of partitions must be an even multiple of the striping width. If not, the number of partitions will be rounded up to the next valid value.
6. The **mklv** command is not allowed on a snapshot volume group.
7. Mirror Write Consistency (MWC) and Bad Block Relocation (BBR) are not supported in a concurrent setup with multiple active nodes accessing a disk at the same time. These two options must be disabled in this type of concurrent setup.
8. Bad block relocation policy of a logical volume is not supported on a volume group that is created with 4 KB block physical volumes.

You can use the System Management Interface Tool (SMIT) **smit mklv** fast path to run this command.

File Systems on Striped Logical Volumes

If you want to create a file system on a striped logical volume, you should create the striped logical volume before you run the **crfs** command or **mkfs** command to create the file system. In order to maximize the use of disk space within the striping width, you should choose hard disks of the same size when creating the striped logical volume. The striping width is the number of hard disks that form the striped logical volume.

Flags

Item

-a *position*

Description

Sets the intra-physical volume allocation policy (the position of the logical partitions on the physical volume). The *position* variable can be one of the following:

m

Allocates logical partitions in the outer middle section of each physical volume. This is the default position.

c

Allocates logical partitions in the center section of each physical volume.

e

Allocates logical partitions in the outer edge section of each physical volume.

ie

Allocates logical partitions in the inner edge section of each physical volume.

im

Allocates logical partitions in the inner middle section of each physical volume.

-b *badblocks*

Sets the bad-block relocation policy. The *Relocation* variable can be one of the following:

y

Causes bad-block relocation to occur. This is the default.

n

Prevents bad-block relocation from occurring.

-c *copies*

Sets the number of physical partitions allocated for each logical partition. The *copies* variable can be set to a value from 1 to 3; the default is 1.

-C *stripewidth*

Sets the Stripe width of the logical volume. If the *Stripewidth* is not entered it is assumed to be the *upperbound* or the total number of disks specified on the command line.

Item**Description**

-d *schedule*

Sets the scheduling policy when more than one logical partition is written. The *schedule* variable can be one of the following:

p

Establishes a parallel scheduling policy. This is the default for scheduling policy.

ps

Parallel write with sequential read policy. All mirrors are written in parallel but always read from the first mirror if the first mirror is available.

pr

Parallel write round robin read. This policy is similar to the parallel policy except an attempt is made to spread the reads to the logical volume more evenly across all mirrors.

s

Establishes a sequential scheduling policy.

Note: The **-R** flag overwrites the read policy specified by the **-d** flag. If the preferred copy is not available, the read operations follow the scheduling policy.

-R *PreferredRead*

Sets read preference to the copy of the logical volume. If the **-R** flag is specified and if the preferred copy is available, the read operation occurs from the preferred copy. If the preferred copy is not available, the read operations follow the scheduling policy of the logical volume. The *PreferredRead* variable can be set to a value in the range 0 -3. The default value is 0.

-e *range*

Sets the inter-physical volume allocation policy (the number of physical volumes to extend across, using the volumes that provide the best allocation). The *Range* value is limited by the *upperbound* variable, (set with the **-u** flag) and can be one of the following:

x

Allocates across the maximum number of physical volumes.

m

Allocates logical partitions across the minimum number of physical volumes. This is the default range.

-G *groupid*

Specifies group ID for the logical volume special file.

-i

Reads the *physicalvolume* parameter from standard input. Use the **-i** flag only when *physicalvolume* is entered through standard input.

Item

>>|-k y | n

Description

Enables the data encryption option in the logical volume. The **-k** flag is available in IBM AIX 7.2 with Technology Level 5, or later. You can specify the following values for this flag:

y

The data encryption option of the logical volume is enabled. The primary key of the logical volume must be initialized to access the logical volume. Use the **hdcryptmgr authinit** command to initialize the primary key of the logical volume.

n

The data encryption option of the logical volume is not enabled. This is the default value.

Note:

- The data encryption option must be enabled at the volume group level before you can enable the data encryption option for a logical volume.
- The **-k** flag is not supported on active boot, dump, paging, and `aio_cache` logical volume type.

<<<

-L

Sets the logical volume label. The default label is **None**. The maximum size of the label file is 127 characters.

Note: If the logical volume is going to be used as a journaled file system (JFS), then the JFS will use this field to store the mount point of the file system on that logical volume for future reference.

-m *mapfile*

Specifies the exact physical partitions to allocate. Partitions are used in the order given by the file designated by the *mapfile* parameter. All physical partitions belonging to a copy are allocated before allocating for the next copy. The *mapfile* format is:

PVname : PPnum1 [-PPnum2]

where *pvname* is a physical volume name (for example, `hdisk0`). It is one record per physical partition or a range of consecutive physical partitions.

PVname

Name of the physical volume as specified by the system.

PPnum

Physical partition number.

Important: When you use map files, you must understand and adhere to all LV-allocation parameters such as strictness, upperbound, and stripe width. Using map files bypasses the checks done in the LVM-allocation routines. This is important for striped LVs, which are assumed to have a typical striped allocation pattern conforming to the stripe width.

Item	Description
-o <i>y/n</i>	Turns on/off serialization of overlapping I/Os. If serialization is turned on then overlapping I/Os are not allowed on a block range and only a single I/O in a block range is processed at any one time. Most applications like file systems and databases do serialization so serialization should be turned off. The default for new logical volumes is off.
-O <i>y/n</i>	Enables the infinite retry option of the logical volume. <p>n</p> <p>The infinite retry option of the logical volume is not enabled. The failing I/O of the logical volume is not retried. This is the default value.</p> <p>y</p> <p>The infinite retry option of the logical volume is enabled. The failed I/O request is retried until it is successful.</p> <p>Note: The infinite retry option is ignored for a logical volume (LV) when <i>active</i> mirror write consistency is set. The infinite retry option must be enabled at the volume group level to work for a logical volume when <i>active</i> mirror write consistency is set.</p> <p>Note: The infinite retry option is not supported in the Geographic Logical Volume Manager (GLVM) environment.</p>
-p <i>copyn=mirrorpool</i>	Enables mirror pools for the logical volume. A mirror pool is assigned to a copy using the <i>copyn=mirrorpool</i> parameter. Specify a mirror pool for each copy. To specify more than one <i>copyn=mirrorpool</i> pair, provide multiple -p <i>copyn=mirrorpool</i> flags. Mirror pool names can be up to 15 characters and follow the same rules that apply to volume group names and logical volume names.
-P <i>modes</i>	Specifies permissions (file modes) for the logical volume special file.
-r <i>relocate</i>	Sets the reorganization relocation flag. For striped logical volumes, the <i>relocate</i> parameter must be set to n (the default for striped logical volumes). The <i>relocate</i> parameter can be one of the following: <p>y</p> <p>Allows the logical volume to be relocated during reorganization. This is the default for relocation.</p> <p>n</p> <p>Prevents the logical volume from being relocated during reorganization.</p>

Item	Description
-s <i>strict</i>	<p>Determines the strict allocation policy. Copies of a logical partition can be allocated to share or not to share the same physical volume. The <i>strict</i> parameter is represented by one of the following:</p> <p>y Sets a strict allocation policy, so copies for a logical partition cannot share the same physical volume. This is the default for allocation policy.</p> <p>n Does not set a strict allocation policy, so copies for a logical partition can share the same physical volume.</p> <p>s Sets a super strict allocation policy, so that the partitions allocated for one mirror cannot share a physical volume with the partitions from another mirror.</p>
-S <i>stripSize</i>	<p>Specifies the number of bytes per strip (the strip size multiplied by the number of disks in an array equals the stripe size). Valid values include 4K, 8K, 16K, 32K, 64K, 128K, 256K, 512K, 1M, 2M, 4M, 8M, 16M, 32M, 64M, and 128M.</p> <p>Note: The -d, -e, and -s flags are not valid when creating a striped logical volume using the -S flag.</p>
-t <i>type</i>	<p>Sets the logical volume type. The standard types are jfs (journaled file systems), jfslog (journaled file system logs), jfs2 (enhanced journaled file system), jfs2log (enhanced journaled file system logs), and paging (paging spaces), but a user can define other logical volume types with this flag. You cannot create a striped logical volume of type boot. The default is jfs2. If a log is manually created for a jfs file system, the user must run the logform command to clean out the new jfslog before the log can be used. For example, to format the logical volume <code>logdev</code>, type:</p> <pre style="background-color: #f0f0f0; padding: 5px;">logform /dev/logdev</pre> <p>where <code>/dev/logdev</code> is the absolute path to the logical volume.</p>
-T 0	<p>The -T 0 option indicates that the logical volume control block does not occupy the first block of the logical volume. Therefore, the space is available for application data. Applications can identify this type of logical volume with the IOCFINFO <code>ioctl</code> operation. The logical volume has a device subtype of <code>DS_LVZ</code>.</p> <p>A logical volume created without this option has a device subtype of <code>DS_LV</code>.</p>
-U <i>userid</i>	<p>Specifies user ID for logical volume special file.</p>

Item	Description
-u <i>upperbound</i>	<p>Sets the maximum number of physical volumes for new allocation. The value of the <i>upperbound</i> variable should be between one and the total number of physical volumes. When using super strictness, the upper bound indicates the maximum number of physical volumes allowed for each mirror copy. When using striped logical volumes, the upper bound must be multiple of <i>stripewidth</i>. If <i>upperbound</i> is not specified it is assumed to be <i>stripewidth</i> for striped logical volumes.</p>
-v <i>verify</i>	<p>Sets the write-verify state for the logical volume. Causes (y) all writes to the logical volume to either be verified with a follow-up read, or prevents (n) the verification of all writes to the logical volume. The <i>verify</i> parameter is represented by one of the following:</p> <p>n Prevents the verification of all write operations to the logical volume. This is the default for the -v flag.</p> <p>y Causes the verification of all write operations to the logical volume.</p>
-w <i>mirrorwriteconsistency</i>	<p>y or a Turns on <i>active</i> mirror write consistency that ensures data consistency among mirrored copies of a logical volume during typical I/O processing. This value is the default mirror write consistency policy for small type of volume group.</p> <p>p Turns on <i>passive</i> mirror write consistency that ensures data consistency among mirrored copies during volume group synchronization after a system interruption.</p> <p>Note: This function is available only on big type and scalable type of volume groups. This value is the default mirror write consistency policy for big type and scalable type of volume groups.</p> <p>n No mirror write consistency. See the -f flag of the syncvg command.</p>
-x <i>maximum</i>	<p>Sets the maximum number of logical partitions that can be allocated to the logical volume. The default value is 512. The number represented by the <i>number</i> parameter must be equal to or less than the number represented by the <i>maximum</i> variable.</p>

Item

-y *newlogicalvolume*

Description

Specifies the logical volume name rather than having the name generated automatically. Logical volume names must be unique system wide and can range from 1 to 15 characters. If the *volume group* is varied on in concurrent mode, the new logical volume name should be unique across all the concurrent nodes where the *volume group* is varied on. The name cannot begin with a prefix already defined in the **PdDv** class in the Device Configuration Database for other devices.

The logical volume name created is sent to standard output. The logical volume name can only contain the following characters:

- "A" through "Z"
- "a" through "z"
- "0" through "9"
- "_" (the underscore)
- "-" (the minus sign)
- "." (the period)

All other characters are considered not valid.

-Y *prefix*

Specifies the *prefix* 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. The 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.

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 make a logical volume in volume group `vg02` with one logical partition and a total of two copies of the data, type:

```
mklv -c 2 vg02 1
```

2. To make a logical volume in volume group `vg03` with nine logical partitions and a total of three copies spread across a maximum of two physical volumes, and whose allocation policy is not strict, type:

```
mklv -c 3 -u 2 -s n vg03 9
```

3. To make a logical volume in `vg04` with five logical partitions allocated across the center sections of the physical volumes when possible, with no bad-block relocation, and whose type is paging, type:

```
mklv -a c -t paging -b n vg04 5
```

4. To make a logical volume in `vg03` with 15 logical partitions chosen from physical volumes `hdisk5`, `hdisk6`, and `hdisk9`, type:

```
mklv vg03 15 hdisk5 hdisk6 hdisk9
```

5. To make a striped logical volume in vg05 with a strip size of 64K across 3 physical volumes and 12 logical partitions, type:

```
mklv -u 3 -S 64K vg05 12
```

6. To make a striped logical volume in vg05 with a strip size of 8K across hdisk1, hdisk2, and hdisk3 and 12 logical partitions, type:

```
mklv -S 8K vg05 12 hdisk1 hdisk2 hdisk3
```

7. To request a logical volume with a minimum size of 10MB, type:

```
mklv VGNAME 10M #
```

The **mklv** command will determine the number of partitions needed to create a logical volume of at least that size.

You can use uppercase and lowercase letters as follows:

B/b	512 byte blocks
K/k	KB
M/m	MB
G/g	GB

8. To create three copies of logical volume in volume group testvg and to set the read preference to the second logical volume copy, enter the following command:

```
mklv -c 3 -R 2 -y testlv testvg 10
```

Files

Item	Description
/usr/sbin	Directory where the mklv command resides.
/tmp	Directory where the temporary files are stored while the command is running.
/dev	Directory where the character and block device entries for the logical volume are created.

mklvcopy Command

Purpose

Provides copies of data within the logical volume.

Syntax

```
mklvcopy [ -a position ] [ -e range ] [ -k ] [ -m mapfile ] [ -s strict ] [ -u upperbound ] [ -p copyn=mirrorpool ]  
logicalvolume copies [physicalvolume...]
```

Description

The **mklvcopy** command increases the number of copies in each logical partition in *logicalvolume*. This is accomplished by increasing the total number of physical partitions for each logical partition to the number represented by *Copies*. The *logicalvolume* parameter can be a logical volume name or logical volume ID. You can request that the physical partitions for the new copies be allocated on specific physical volumes

(within the volume group) with the *physicalvolume* parameter; otherwise, all the physical volumes within the volume group are available for allocation.

The logical volume modified with this command uses the *copies* parameter as its new **copy** characteristic. The data in the new copies are not synchronized until one of the following occurs: the **-k** option is used, the volume group is activated by the **varyonvg** command, or the volume group or logical volume is synchronized explicitly by the **syncvg** command. Individual logical partitions are always updated as they are written to.

The default allocation policy is to use minimum numbering of physical volumes per logical volume copy, to place the physical partitions belong to a copy as contiguously as possible, and then to place the physical partitions in the desired region specified by the **-a** flag. Also, by default, each copy of a logical partition is placed on a separate physical volume.

Notes:

- To use this command, you must either have **root** user authority or be a member of the **system** group.
- The **mklvcopy** command is not allowed on a snapshot volume group.
- When you create a copy of a logical volume with a superstrict allocation policy, the **mklvcopy** command first attempts to mimic the physical-partition mapping of the first mirror copy onto another set of disks in the volume group. This algorithm ignores the interphysical and intraphysical volume allocation policies, even when the policies are specified as arguments to the **mklvcopy** command. If it is not possible to mimic the first copy's physical partition mapping, the usual allocation algorithm, which utilizes the interphysical and intraphysical volume allocation policies, is used.

You can use the System Management Interface Tool (SMIT) **smit mklvcopy** fast path to run this command.

Flags

Note: The **-e** and **-s** flags are not valid with a striped logical volume.

Item	Description
-a position	Sets the intra-physical volume allocation policy (the position of the logical partitions on the physical volume). The <i>position</i> variable can be one of the following: m Allocates logical partitions in the outer middle section of each physical volume. This is the default position. c Allocates logical partitions in the center section of each physical volume. e Allocates logical partitions in the outer edge section of each physical volume. ie Allocated logical partitions in the inner edge section of each physical volume. im Allocates logical partitions in the inner middle section of each physical volume.

Item	Description
-e <i>range</i>	<p>Sets the inter-physical volume allocation policy (the number of physical volumes to extend across, using the volumes that provide the best allocation). The <i>range</i> value is limited by the <i>upperbound</i> variable (set with the -u flag), and can be one of the following:</p> <p>x Allocates across the maximum number of physical volumes.</p> <p>m Allocates logical partitions across the minimum number of physical volumes. This is the default for the -e flag.</p>
-k	Synchronizes data in the new partitions.
-m <i>mapfile</i>	<p>Specifies the exact physical partitions to allocate. Partitions are used in the order given by the file designated by the <i>mapfile</i> parameter. All physical partitions belonging to a copy are allocated before allocating for the next copy. The <i>mapfile</i> format is:</p> <p>PVname:PPnum1[-PPnum2] where <i>pvname</i> is a physical volume name (for example, <i>hdisk0</i>). It is one record per physical partition or a range of consecutive physical partitions.</p> <p>PVname Name of the physical volume as specified by the system.</p> <p>PPnum Physical partition number.</p> <p>Important: When you use map files, you must understand and adhere to all LV-allocation parameters such as strictness, upperbound, and stripe width. Using map files bypasses the checks done in the LVM-allocation routines. This is important for striped LVs, which are assumed to have a typical striped allocation pattern conforming to the stripe width.</p>
-p <i>copyn=mirrorpool</i>	Assigns mirror pools to the copies being created. A mirror pool is assigned to a copy using the <i>copyn=mirrorpool</i> parameter. Specify a mirror pool for each copy being created. To specify more than one <i>copyn=mirrorpool</i> pair, provide multiple -p <i>copyn=mirrorpool</i> flags.
-s <i>strict</i>	<p>Determines the strict allocation policy. Copies of a logical partition can be allocated to share or not to share the same physical volume. The <i>strict</i> variable is represented by one of the following:</p> <p>y Sets a strict allocation policy, so copies for a logical partition cannot share the same physical volume. flag.</p> <p>n Does not set a strict allocation policy, so copies for a logical partition can share the same physical volume.</p> <p>s Sets a super strict allocation policy, so that the partitions allocated for one mirror cannot share a physical volume with the partitions from another mirror. See Note 4 for other effects of the superstrict allocation policy on mklvcopy behavior.</p> <p>Note: When changing a nonsuper strict logical volume to a super strict logical volume, you must specify physical volumes or use the -u flag.</p>

Item	Description
-u <i>upperbound</i>	Sets the maximum number of physical volumes for new allocation. The value of the <i>upperbound</i> variable should be between one and the total maximum number of physical volumes per VG. When using super strictness, the upper bound indicates the maximum number of physical volumes allowed for each mirror copy. When using striped logical volumes, the upper bound must be multiple of <i>stripewidth</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.

Example

To add physical partitions to the logical partitions in the logical volume `lv01`, so that a total of three copies exists for each logical partition, enter:

```
mklvcopy lv01 3
```

The logical partitions in the logical volume represented by directory `lv01` have three copies.

Files

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

mkmaster Command

Purpose

Executes the **ypinit** command and starts the NIS daemons to configure a controller server.

Syntax

```
/usr/sbin/mkmaster [ -s HostName [ ,HostName ... ] ] [ -O | -o ] [ -E | -e ] [ -P | -p ] [ -U | -u ] [ -C | -c ] [ -I | -B | -N ]
```

Description

The **mkmaster** command invokes the **ypinit** command to build the NIS maps for the current domain, if the domain name of the system is currently set. After the **ypinit** command completes successfully, the **mkmaster** command uncomments the entries in the `/etc/rc.nfs` file for the **ypserv** command, **ypasswdd** command, **ypupdated** command, and **ypbind** command.

You can use the System Management Interface Tool (SMIT) **smit mkmaster** fast path to run this command.

Flags

Item	Description
-s <i>HostName</i> [, <i>HostName</i> ...]	Specifies the worker host names for this controller server. These worker hosts must be configured after the controller server has been configured. The mkmaster command automatically adds the current host to this list.
-O	Overwrites existing maps for this domain.
-o	Prevents the overwriting of existing maps for this domain. This flag is the default.
-E	Prevents further action if errors are encountered while building new maps. This is true for both the ypinit command and the mkmaster command. This flag is the default.
-e	Does not exit from the ypinit command and the mkmaster command if errors are encountered.
-P	Starts the yppasswdd daemon along with the ypserv daemon.
-p	Suppresses the start of the yppasswdd daemon. This flag is the default.
-U	Starts the ypupdated daemon along with the ypserv daemon.
-u	Suppresses the start of the ypupdated daemon. This flag is the default.
-C	Starts the ypbind daemon along with the ypserv daemon. This flag is the default.
-c	Suppresses the start of the ypbind daemon.
-I	Directs the mkmaster command to change the /etc/rc.nfs file to start the appropriate daemons on the next system restart. The execution of the ypinit command occurs when this command is invoked.
-B	Executes the ypinit command, uncomments the entries in the /etc/rc.nfs file, and starts the daemons. This flag is the system default.
-N	Executes the ypinit command and starts the appropriate daemons without changing the /etc/rc.nfs file.

Example

To execute the **ypinit** command, overwrite any existing maps for the current domain, and make **host1** and **host3** worker servers, enter:

```
mkmaster -s host1,host3 -O -p -u -B
```

This command will not start the **yppasswdd** daemon or the **ypupdated** daemon.

Files

Item	Description
/var/yp/domainname directory	Contains the NIS maps for the NIS domain.
/etc/rc.nfs	Contains the startup script for the NFS and NIS daemons.

mknamsv Command

Purpose

Configures TCP/IP-based name service on a host for a client.

Syntax

```
mknamsv { -a "Attribute=Value ..." | -A FileName }
```

Description

The **mknamsv** high-level command configures a TCP/IP instance to use a name server. It calls the **namerslv** low-level command to configure the **resolv.conf** file appropriately.

You can use the System Management Interface Tool (SMIT) **smit mknamerslv** fast path to run this command.

Flags

Item	Description
-A <i>FileName</i>	Specifies the name of the file containing named daemon initialization information.
-a "Attribute=Value..."	Specifies a list of attributes with corresponding values to be used for updating the named server initialization files in the database. Attributes available are: domain Domain name nameserver Internet address of name server in dotted decimal format

Examples

1. To configure the name server initialization files, enter the command in the following format:

```
mknamsv -a"domain=austin.century.com nameserver=192.9.200.1"
```

In this example the domain name and name server address are updated. The previous domain and name server are overwritten.

2. To configure name server initialization files according to information in another file, enter the command in the following format:

```
mknamsv -A namsv.file
```

In this example, the file that contains the configuration information is *namsv.file*. The "**attribute=value**" pairs must not be placed in one line. As an example, enter the "**attribute=value**" pairs to *namsv.file* in the following format:

```
domain=austin.century.com  
nameserver=192.9.200.1
```

Files

Item	Description
/etc/resolv.conf	Contains domain name server information for local resolver routines.

mknetid Command

Purpose

Generates data for the **netid.byname** map for use by the Network Information Services (NIS).

Syntax

To Create an NIS Map:

```
/usr/sbin/mknetid [ -q ] [ -p PasswordFile ] [ -g GroupFile ] [ -h HostsFile ] [ -m NetidFile ]
```

Description

The **mknetid** command is used to produce the data for the **netid.byname** NIS map. It will parse the files specified on the command line and build the corresponding netid keys and values. Users will get the following entries:

```
unix.<uid>@<domainname> <uid>:<gid1>,<gid2>,...
```

Hosts will get the following entries:

```
unix.<hostname>@<domainname> 0:<hostname>
```

The domainname that is used is the same that is configured on the system at the time **mknetid** is run. The generated data is sent to **stdout**. Each line contains one entry, with the key and the data separated by a space.

Flags

Item	Description
-q	Quiet mode - do not report any warnings about the data.
-p PasswordFile	Specifies which passwd file to be used for reading the list of users.
-g GroupFile	Specifies which groups file to be used for reading the list of group memberships.
-h HostsFile	Specifies which hosts file to be used for reading the list of hostnames.
-m NetidFile	Specifies a file from which to read any additional netid entries to be included.

Files

Item	Description
/var/yp/Makefile	mknetid is most commonly used when rebuilding the NIS databases using /var/yp/Makefile .
/etc/passwd	Where <i>PasswordFile</i> resides.
/etc/groups	Where <i>GroupFile</i> resides.
/etc/hosts	Where <i>HostsFile</i> resides.
/etc/netid	Where <i>NetidFile</i> resides.

mknfs Command

Purpose

Configures the system to run NFS.

Syntax

```
/usr/sbin/mknfs [ -I | -N | -B ]
```

Description

The **mknfs** command configures the system to run the Network File System (NFS) daemons. The **mknfs** command adds an entry to the **inittab** file so that the **/etc/rc.nfs** file is executed on system restart.

Flags

Item	Description
------	-------------

- | | |
|-----------|---|
| -B | Adds an entry to the inittab file to execute the /etc/rc.nfs file on system restart. The mknfs command also executes the /etc/rc.nfs file immediately to start the NFS daemons. This flag is the default. |
| -I | Adds an entry to the inittab file to execute the /etc/rc.nfs file on system restart. |
| -N | Starts the /etc/rc.nfs file to start the NFS daemons immediately. When started this way, the daemons run until the next system restart. |

Files

Item	Description
inittab	Controls the initialization process of the system.
/etc/rc.nfs	Contains the startup script for the NFS and NIS daemons.

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.

mknfsexp Command

Purpose

Exports a directory to NFS clients.

Syntax

```
/usr/sbin/mknfsexp -d Directory [ -e pathname ] [ -f Exports_File ] [ -t [ { rw | ro | remove } { rm | h HostName [ , HostName ... ] } ] [ -a UID ] [ -r HostName [ , HostName ... ] ] [ -c HostName [ , HostName ... ] ] [ -s | -n ] [ -I | -B | -N ] [ -P | -p ] [ -D {yes | no} ] [ -v ] [ -x | -X ] [ -S flavor [ , flavor ] ] [ -G rootpath@host[+host][:rootpath@host[+host]] ] [ -g rootpath@host[+host][:rootpath@host[+host]] ] [ -o Ordering ]
```

Description

The **mknfsexp** command takes the flags and parameters specified and constructs a line that is syntactically correct for the **/etc/exports** file. If this command is invoked with the **-B** flag, an entry will be added to the **/etc/exports** file and the **exportfs** command is invoked to export the directory specified.

Alternatively, the **-I** flag adds an entry to the **exports** file and does not export the directory, or the **-N** flag does not add an entry to the **exports** file but does export the directory.

Flags

Item	Description
-a <i>UID</i>	Uses the UID variable as the effective user ID only if a request comes from an unknown user. The default value of this option is -2. Note: Root users (UID 0) are always considered unknown by the NFS server, unless they are included in the root option. Setting the value of UID to -1 disables anonymous access.
-B	Adds an entry to the /etc/exports file and the exportfs command is executed to export the directory. This flag is the default.
-c <i>HostName</i> [, <i>HostName</i>] ...	Gives mount access to each of the clients listed. A client can either be a host or a netgroup. The default is to allow all hosts access.
-d <i>Directory</i>	Specifies the directory that is to be exported or changed.
-D {yes no}	Enables or disables file delegation for the specified export. This option overrides the system-wide delegation enablement for this export. The system-wide enablement is done through nfso .
-e <i>pathname</i>	Specifies an export name for the directory.
-f <i>Exports_File</i>	Specifies the full path name of the exports file to use if other than the /etc/exports file.
-g <i>rootpath@host</i> [<i>+host</i>] [<i>:rootpath@host</i>] [<i>+host</i>]	The specified directory will be marked with replica information. If the server becomes unreachable by an NFS client, the client can switch to one of the specified servers. This option is only accessible using NFS version 4 protocol, and version 4 access must be specified in the options. Because the directory is being exported for client access, specifying NFS version 2 or version 3 access will not cause an error, but the request will simply be ignored by the version 2 or version 3 server. This option cannot be specified with the -G flag. Only the host part of each specification is verified. The administrator must ensure that the specified <i>rootpaths</i> are valid and that the target servers contain appropriate data. If the directory being exported is not in the replica list, that directory will be added as the first replica location. The administrator should ensure that appropriate data exists at the replica locations. For a more complete description of replication, see the exportfs command. The -g option is available only on AIX 5.3 with 5300-03 or later. Note: A referral or replica export can only be made if replication is enabled on the server. Use chnfs -R on to enable replication.

Item	Description
-G <i>rootpath@host</i> [+host] [:rootpath@host [+host]]	<p>A namespace referral will be created at the specified path. The referral directs clients to the specified alternate locations where they can continue operations. A referral is a special object. If a nonreferral object exists at the specified path, the export is disallowed and an error message is printed. If nothing exists at the specified path, a referral object is created there that includes the path name directories leading to the object. A referral cannot be specified for the <code>nfsroot</code>. The name <code>localhost</code> cannot be used as a <i>hostname</i>. The <code>-G</code> option is allowed only for version 4 exports. If the export specification allows version 2 or version 3 access, an error message will be printed and the export will be disallowed. The administrator should ensure that appropriate data exists at the referral locations. For a more complete description of referrals, see the exportfs command. The <code>-G</code> option is available only on AIX 5L Version 5.3 with the 5300-03 Recommended Maintenance package or later.</p> <p>Note: A referral or replica export can only be made if replication is enabled on the server. Use <code>chnfs -R</code> on to enable replication.</p>
-h <i>HostName</i> [, <i>HostName</i>] ...	Specifies which hosts have read-write access to the directory. This option is valid only when the exported file is to be read-mostly.
-I	Adds an entry to the <code>/etc/exports</code> file so that the next time the exportfs command is run during system restart, the directory will be exported.
-n	Does not require the client to use the more secure protocol. This flag is the default.
-N	Does not add an entry to the <code>/etc/exports</code> file but the exportfs command is run with the correct parameters so that the directory is exported.
-o <i>Ordering</i>	<p>Defines how the alternate locations list is generated from the servers that you specified on the refer or replicas option. The option applies only to directories exported for access by NFS version 4 protocol. The <i>Ordering</i> parameter has three allowable values:</p> <p>full All of the servers are scattered to form the combinations of alternate locations.</p> <p>partial The first location of all the combinations is fixed to the first server specified on the refer or replicas option. The rest of the locations and the first location are scattered as if they are scattered using the <code>scatter=full</code> method.</p> <p>none No scatter is to be used. The value can also be used to disable scattering if it was enabled previously.</p>
-p	Specifies that the exported directory is not a public directory.
-P	Specifies that the exported directory is to be a public directory.
-r <i>HostName</i> [, <i>HostName</i>] ...	Gives root users on the specified hosts access to the directory. The default is for no hosts to be granted root access.
-s	Requires clients to use a more secure protocol when accessing the directory.

Item	Description
-S <i>flavor</i> [, <i>flavor</i>]	<p>May be used in conjunction with the -c, -t, or -r options to associate the option with one or more specific security methods. Most exportfs options can be clustered using the sec option. Any number of sec stanzas may be specified, but each security method can be specified only once.</p> <p>Allowable flavor values are:</p> <p>sys UNIX authentication.</p> <p>dh DES authentication.</p> <p>none Use the anonymous ID if it has a value other than -1. Otherwise, a weak auth error is returned.</p> <p>krb5 Kerberos. Authentication only.</p> <p>krb5i Kerberos. Authentication and integrity.</p> <p>krb5p Authentication, integrity, and privacy.</p>
-t <i>Type</i>	<p>Specifies whether the directory is read-write, read-only, or read-mostly. The possible values for the <i>Type</i> variable are:</p> <p>rw Exports the read-write directory. This is the system default.</p> <p>ro Exports the read-only directory.</p> <p>remove Removes the exported directory.</p> <p>rm Exports the read-mostly directory. If chosen, the -h flag must be used to specify the hosts that have read-write permission.</p>
-v <i>number</i> [, <i>number</i>] ...	<p>The directory specified by the -d option is made available to clients using the specified NFS versions. Valid values are 2, 3, or 4. You can export two entries for the same directory with different versions 2 (or 3) and 4.</p>
-x	<p>Accepts the replica location information specified with the -g option as-is. Does not insert the server's primary hostname into the list if it is not present. This flag is intended for use with servers with multiple network interfaces. If none of the server's host names are in the replica list, NFSv4 clients might treat the location information as faulty and discard it.</p>
-X	<p>Enables auto-insert of the primary hostname into the replica list. If the server's primary hostname is not specified in the replica list, the hostname will be added as the first replica location.</p>

Examples

1. To export a directory with read-only permission, enter:

```
mknfsexp -d /usr -t ro
```

In this example, the `mknfsexp` command exports the `/usr` directory with read-only permission.

2. To export a directory with read-mostly permission and a secure protocol to specific hosts, enter:

```
mknfsxp -d /home/guest -t rm -h bighost,littlehost -s
```

In this example the `mknfsxp` command exports the `/home/guest` directory with read-mostly permission, using more secure protocol.

3. To export a directory with read-write permission to a specific netgroup and specific hosts, and to make the export effective on the next system restart, enter:

```
mknfsxp -d /usr -t rw -c host1,host3,grp3 -I
```

In the above example, the `mknfsxp` command exports the `/usr` directory and gives read and write permission to `host1`, `host2`, and `grp3`. The `-I` flag makes this change effective on the next system restart.

4. To export a directory with read-only permission to an exports file other than `/etc/exports`, enter:

```
mknfsxp -d /usr -t ro -f /etc/exports.other
```

In the above example, the `mknfsxp` command exports the `/usr` directory with read-only permission to the `/etc/exports.other` file.

5. To export the `/common/documents` directory to allow access only to clients using NFS version 4 protocol, enter:

```
mknfsxp -d /common/documents -v 4
```

6. To export the `/common/documents` directory, allowing access to `client1` and `client2` for clients using `krb5` access, enter:

```
mknfsxp -d /common/documents -S krb5 -r client1,client2
```

7. To export the `/common/documents` directory with full scattering for the hosts named `s1` and `s2` specified as referrals, enter the following command:

```
mknfsxp -d /common/documents -v 4 -G /common/documents@s1:/common/
documents@s2 -o full
```

8. To export the `/common/documents` directory with partial scattering at hosts named `s1`, `s2` and `s3`, specified as replicas, enter the following command:

```
mknfsxp -d /common/documents -v 4 -g /common/documents@s1:/common/
documents@s2:/common/documents@s3 -o partial
```

9. To export the `/common/documents` directory with the export name `/exports1/cool/mike`, enter the following command:

```
mknfsxp -d /common/documents -e /exports1/cool/mike -S sys -v 4
```

Files

Item	Description
/etc/exports	Lists the directories that the server can export.

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.

mknfsmnt Command

Purpose

Mounts a directory from an NFS server.

Syntax

```
/usr/sbin/mknfsmnt -f PathName -d RemoteDirectory -h RemoteHost [ -t { rw | ro } ] [ -m MountTypeName ] [ -w { fg | bg } ] [ -X | -x ] [ -S | -H ] [ -Y | -y ] [ -Z | -z ] [ -e | -E ] [ -a | -A ] [ -j ] [ -J ] [ -q | -Q ] [ -g | -G ] [ -s | -n ] [ -I | -B | -N ] [ -r TimesToRetry ] [ -R NumRetrans ] [ -b ReadBufferSize ] [ -c WriteBufferSize ] [ -o TimeOut ] [ -P PortNumber ] [ -u AcRegMin ] [ -U AcRegMax ] [ -v AcDirMin ] [ -V AcDirMax ] [ -T AcTimeOut ] [ -p NumBioids ] [ -K { any | 2 | 3 | 4 } ] [ -k { any | tcp | udp } ] [ -l | -L ] [ -M security_methods ] [ -i { dio | cio [,cior] } ]
```

Description

The **mknfsmnt** command constructs an entry that is appended to the **/etc/filesystems** file, thus making a file system available for mounting. If the mount is to be permanent, this entry remains. If the mount is temporary, the flags are used directly for the **mount** command. If the mount is soft and the server does not respond, the system returns an error. If the mount is hard, the client continues trying until the server responds. The hard mount is the default mount.

Flags

Item	Description
-A	The /etc/filesystems entry for this file system specifies that it should be automatically mounted at system restart.
-a	The /etc/filesystems entry for this file system specifies that it should not be automatically mounted at system restart. This is the default flag.
-B	Adds an entry to the /etc/filesystems file and attempts to mount the file system. This is the default flag.
-b <i>ReadBufferSize</i>	Indicates the size of the read buffer in bytes specified by the <i>ReadBufferSize</i> variable.
-c <i>WriteBufferSize</i>	Indicates the size of the write buffer in bytes specified by the <i>WriteBufferSize</i> variable.
-d <i>RemoteDirectory</i>	Specifies the directory that is mounted on the path name specified.
-E	Allows keyboard interrupts on hard mounts.
-e	Prevents keyboard interrupts on hard mounts. This is the default flag.
-f <i>PathName</i>	Specifies the mount point for the remote directory.
-G	Directs any file or directory created on the file system to inherit the group ID of the parent directory.
-g	Does not direct new files or directories created on the file system to inherit the group ID of the parent directory. This is the default flag.
-H	Creates a hard mount, which causes the client to continue retrying until the server responds. This is the default flag.
-h <i>RemoteHost</i>	Specifies the NFS server that is exporting the directory.
-I	Causes an entry to be added to the /etc/filesystems file. The directory is not mounted.

Item	Description
-i	<p>Specifies I/O mode for the mount. The options are:</p> <p><i>cio</i> Specifies concurrent I/O mode. Specifies the file system to be mounted for concurrent readers and writers. I/O on files in this file system will behave as if they had been opened with O_CIO specified in the open() system call.</p> <p><i>dio</i> Specifies direct I/O mode. Specifies that I/O on the file system will behave as if all the files had been opened with O_DIRECT specified in the open() system call.</p> <p><i>cior</i> Specifies concurrent I/O with read-only mode.</p> <p>Note: For more information on the cio and dio options, see the mount command.</p>
-J	Indicates that acls are used on this mount.
-j	Indicates that acls are not used on this mount. This is the default flag.
-K	<p>Specifies the NFS version used for this NFS mount. The options are:</p> <p><i>any</i> Uses the mount command to determine the correct match. Refer to the mount command for a description of the current default behavior.</p> <p>2 Specifies NFS Version 2.</p> <p>3 Specifies NFS Version 3.</p> <p>4 Specifies NFS Version 4.</p>
-k	<p>Specifies the transport protocol used for the mount. The options are:</p> <p><i>any</i> Uses the mount command to select the protocol to use. TCP protocol is the preferred protocol.</p> <p><i>tcp</i> Specifies the TCP protocol.</p> <p><i>udp</i> Specifies the UDP protocol.</p>
L	Indicates that the lock requests are handled locally without connecting to the server.
l	Indicates that the lock requests are not handled locally. The server handles the lock requests.
-M <i>security_methods</i>	A list of security methods to use when attempting the mount. A comma separated list of the values sys , dh , k1b5 , k1b5i , k1b5p , which correspond to UNIX, DES, Kerberos 5, Kerberos 5 with integrity, and Kerberos 5 with privacy. Multiple values are allowed, but are meaningful only with NFS Version 4 mounts. If multiple methods are given for a Version 2 or 3 protocol mount, the first method is used. For a NFS Version 4 mount, the methods are tried in the listed order.

Item	Description
-m <i>MountTypeName</i>	Specifies the type of file system to mount. File system types are specified in the /etc/filesystems file with the type variables. When the mount -t MountTypeName command is issued, all the currently unmounted file systems with a type equal to the <i>MountTypeName</i> are mounted.
-N	Mounts the directory with the options specified but does not modify the /etc/filesystems file.
-n	Instructs the mount not to use a more secure protocol. This is the default flag.
-o <i>TimeOut</i>	Indicates the length of the NFS timeout in tenths of a second as specified by the <i>TimeOut</i> variable.
-P <i>PortNumber</i>	Indicates the Internet Protocol port number for the server.
-p <i>NumBiods</i>	Specifies the number of biod daemons that are allowed to work on a particular file system. The biod daemons handle client requests. The default number of daemons is 7 for NFS Version 2 and 32 for NFS Version 3 and NFS Version 4.
-Q	Requests that no posix pathconf information be exchanged and made available on an NFS Version 2 mount. Requires a mount Version 2 rpc.mountd at the NFS server.
-q	Specifies that no posix pathconf information is exchanged if mounted as an NFS Version 2 mount. This is the default flag.
-r <i>TimesToRetry</i>	Indicates the number of times to retry a mount. The default value is 1000.
-R <i>NumRetrans</i>	For a soft mount, this flag specifies the number of times that a request has to be transmitted if it is not acknowledged by the server. If the request is unacknowledged after <i>NumRetrans</i> transmissions, the client gives up the request. If this flag is not specified, the default value 3 is used.
-S	Creates a soft mount, which means the system returns an error if the server does not respond.
-s	Instructs the mount to use a more secure protocol.
-T <i>AcTimeOut</i>	Sets the minimum and maximum times allowed for regular files and directories to the number of seconds specified by the <i>Actimeo</i> variable. If this flag is specified, the other cached attribute times are overridden.
-t <i>Type</i>	Specifies that the directory is either read-write or read-only. rw Mounts the directory read-write. This type is the default for the system. ro Mounts the directory read-only.
-U <i>AcRegMax</i>	Holds cached attributes for no more than the number of seconds specified by the <i>AcRegMax</i> variable after file modification.
-u <i>AcRegMin</i>	Holds cached attributes for at least the number of seconds specified by the <i>AcRegMin</i> variable after file modification.
-V <i>AcDirMax</i>	Holds cached attributes for no more than the number of seconds specified by the <i>AcDirMax</i> variable after directory update.

Item	Description
-v <i>AcDirMin</i>	Holds cached attributes for at least the number of seconds specified by the <i>AcDirMin</i> variable after directory update.
-w <i>Location</i>	Indicates where the mount should be attempted. The <i>Location</i> variable can have one of the following values: fg Attempts the mount in the foreground. This is the default value. bg Attempts the mount in the background. If background is specified and the attempt to mount the directory fails, the mount will be retried in the background.
-x	Specifies that the server does not support long device numbers. Use this flag when mounting from an NFS server that does not correctly handle device numbers that are 32 bits long.
-X	Specifies that the server does support long device numbers. This is the default flag.
-y	Indicates that the execution of suid and sgid programs is not allowed in this file system.
-Y	Indicates that the execution of suid and sgid programs are allowed in this file system. This is the default flag.
-z	Indicates that device access through this mount is not allowed; that is, the device cannot be opened on this mount point.
-Z	Indicates that device access through this mount is allowed. This is the default flag.

Example

To add the mount of a remote directory, enter:

```
mknfsmnt -f /usr/share/man -d /usr/share/man -h host1
```

In this example, the `mknfsmnt` command mounts the remote directory `/usr/share/man` on the `/usr/share/man` directory that resides on `host1`.

Files

Item	Description
<u>/etc/filesystems</u>	Lists the remote file systems to be mounted during the system restart.

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.

mknfsproxy Command

Purpose

Creates a new NFS proxy-enabled Cachefs instance that is backed with an NFS client mount to a back-end NFS server.

Syntax

```
/usr/sbin/mknfsproxy [-L] -c local_cache_directory -d Cachefs_mount_point [-o param=n[, param=n]]  
-m [nfs_mount_options] remote_server:remote_directory [-e [export_option, [export_option]]]
```

Description

The local file system used by the created Cachefs instance must be a JFS2 file system. The required inputs include the remote server and directory (*remote_server:remote_directory*) that the Cachefs instance will access, the local directory (*local_cache_directory*) where information will be cached, and the directory where the Cachefs will be mounted.

After the cache is initialized, the Cachefs instance is mounted and ready to be NFS exported. Provide NFS export information so that the cached view will also be NFS exported using the specified options.

Flags

Item	Description
-c	Specifies the local JFS2 file system directory where Cachefs will store cached data and state. This is a required option.
-d	Specifies the directory where Cachefs will be mounted. This is a required option.
-e	Specifies the NFS server export options for the created Cachefs instance. If this is supplied, the created Cachefs instance will also be NFS exported using the supplied options. If this option is not supplied, the created Cachefs instance will be exported with the same NFS version specified by the -m option.
-L	Causes the Cachefs instance to acquire a single lock from its associated NFS back-end that covers the entire file when any byte range locks are requested. When the count of byte range locks drops to 0 (zero), the lock on the back-end NFS server is released.
-m	Specifies the NFS client mount, which might optionally include NFS client mount options as described in the mount man page. This is a required option, and the remote server and remote directory must be supplied.
-o	Specifies Cachefs configuration options in the form <i>param=n</i> . For descriptions of the Cachefs resource parameters, refer to the <i>cfsadmin</i> command.

Parameters

Item	Description
<i>Cachefs_mount_point</i>	Specifies where the proxy-enabled Cachefs instance is to be mounted.
<i>export_option</i>	Specifies which options of the <code>export</code> command are used for the Cachefs instance.
<i>local_cache_directory</i>	Specifies the local directory where information is cached.
<i>nfs_mount_options</i>	Specifies the NFS client options of the <code>mount</code> command.
<i>remote_directory</i>	Specifies the remote directory that the Cachefs instance accesses.
<i>remote_server</i>	Specifies the remote server that the Cachefs instance accesses.

Exit Status

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

Examples

1. To create a proxy-enabled Cachefs instance at `/edge` that accesses `/project1` at NFS server `foo`, enter:

```
mknfsproxy -c /cache/project1 -d /edge -m hard,vers=4,intr foo:/project1
```

In the preceding example, information accessed through `/edge` is cached at `/cache/project1`.

The following variation of the preceding example also exports the created Cachefs instance (`proj1_cached`) for NFS V4 access with authentication flavor of Kerberos 5 and an external name set to `/nfs4/projects/project1`:

```
mknfsproxy -c /cache/project1 -d /edge -m hard,vers=4,intr foo:/project1  
-e sec=krb5,vers=4,exname=/nfs4/projects/project1
```

Location

`/usr/sbin/mknfsproxy`

mknod Command

Purpose

Creates a special file.

Syntax

Only executed by root or system group member

```
mknod Name { b | c } Major Minor
```

Creates FIFO (first-in, first-out) files, which are also called pipes or pipelines

```
mknod Name { p }
```

Description

The **mknod** command makes a directory entry and corresponding i-node for a special file. The first parameter is the name of the entry device. Select a name that is descriptive of the device. The **mknod** command has two forms that have different flags.

The first form of the **mknod** command can be executed by root or a member of the system group. In the first form, the **b** or **c** flag is used. The **b** flag indicates that the special file is a block-oriented device (disk, diskette, or tape). The **c** flag indicates that the special file is a character-oriented device (other devices).

The last two parameters of the first form are numbers that specify the *Major* device and the *Minor* device. The *Major* device number helps the operating system find the device driver code. The *Minor* device number is the unit drive or line number that might be either decimal or octal. The major and minor numbers for a device are assigned by the device's configure method and are kept in the [CuDvDr](#) class in [ODM](#).

It is important that the major and minor numbers be defined in this object class to ensure consistency of device definitions through the system.

In the second form of the **mknod** command, the **p** flag is used to create FIFO pipelines.

Flags

Ite	Description
-----	-------------

m	
----------	--

b	Indicates that the special file is a block-oriented device (disk, diskette, or tape).
----------	---

c	Indicates that the special file is a character-oriented device (other devices).
----------	---

p	Creates FIFOs (named pipelines).
----------	----------------------------------

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 special file for a new diskette drive, enter the following command:

```
mknod /dev/fd2 b 1 2
```

This command creates the `/dev/fd2` special file that is a special block file with the major device number 1 and the minor device number 2.

2. To create the special file for a new character drive, enter the following command:

```
mknod /dev/fc1 c 1 2
```

This command creates the `/dev/ƒc1` special file that is a special character file with the major device number 1 and the minor device number 2.

3. To create a FIFO pipe file, enter the following command:

```
mknod fifo1 p
```

This command creates a FIFO pipe file that has the name `fifo1`.

Files

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

mknotify Command

Purpose

Adds a notify method definition to the `Notify` object class.

Syntax

```
mknotify -n NotifyName -m NotifyMethod
```

Description

The **mknotify** command adds a notify method definition to the **Notify** object class. When a notify method is defined for both a subsystem name and a group name, the subsystem name takes precedence. For example, if the subsystem notify method is executed by the System Resources Controller (SRC), the group notify method is not performed.

The SRC places the name of the unsuccessful subsystem as the first argument to the method and the name of the unsuccessful subsystem group as the second argument.

Flags

Item	Description
-m <i>NotifyMethod</i>	Specifies an absolute path to an executable program that starts when the subsystem stops abnormally.
-n <i>NotifyName</i>	Specifies the subsystem or group name to which the notify method belongs. The <i>NotifyName</i> variable must exist as either a valid subsystem name or a valid group name in the Subsystem object class. The mknotify command is unsuccessful if the <i>NotifyName</i> variable already exists 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 that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To add a notify method for the `srctest` subsystem, enter:

```
mknotify -n srctest -m /usr/lpp/srctest/failure
```

This adds a subsystem notify method for the `srctest` subsystem, with a notify method designated in the `/usr/lpp/srctest/failure` file.

2. To add a notify method for the `tcpip` group, enter:

```
mknotify -n tcpip -m /usr/lpp/tcpip/tcpfailure
```

This adds a group notify method for the `tcpip` group, with a notify method designated in the `/usr/lpp/tcpip/tcpfailure` file.

Files

Item	Description
<code>/etc/objrepos/SRCsubsys</code>	Specifies the SRC Subsystem Configuration object class.
<code>/etc/objrepos/SRCnotify</code>	Specifies the SRC Notify Method object class.

mkpasswd Command

Purpose

Organizes the basic user database for efficient searches.

Syntax

```
mkpasswd [ -v ] { -f | -d | -c } indexname
```

Description

The **mkpasswd** generates indexes over certain security files. These indexes are used by the **getpwnam**, **getpwuid**, **getuserattr**, and **putuserattr** library subroutines.

This approach significantly enhances performance for large user base systems. The following indexes, defined in `/usr/include/usersec.h`, are created:

Item	Description
<code>/etc/passwd.nm.idx:</code>	Index over <code>/etc/passwd</code> file using username as key.
<code>/etc/passwd.id.idx:</code>	Index over <code>/etc/passwd</code> file using userid number as key.
<code>/etc/security/passwd.idx:</code>	Index over <code>/etc/security/passwd</code> file.
<code>/etc/security/lastlog.idx:</code>	Index over <code>/etc/security/lastlog</code> file.

Notes:

1. Modifying the security files over which indexes are built by an editor disables the use of indexing mechanism.
2. Indexed read of a data file is automatically done if a corresponding index exists over the file and is not older than it (except for lastlog index) .
3. In order for indexed mechanism to be used at login, the **mkpasswd** command must have generated indexes.

4. The indexing mechanism replaces the previous hashing mechanism which used dbm files.

Flags

Item	Description
-v	Reports progress if index built.
-f	Forces building of all indexes.
-d	Deletes all indexes.
-c	Checks all indexes and rebuilds the ones that look suspicious.
<i>indexname</i>	Forces building of a particular index.

Security

Access Control: Only the root user and members of the security group should have execute (x) access to this command. The command should be setuid to the root user so the command has access to the user database. Members of the security group should have access to all the files listed in the [Files](#) section. This command should have the **trusted computing base** attribute.

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 Accessed:

Item	Description
Mode	File
r	/etc/passwd
rw	/etc/passwd.nm.idxnnnn.tmp and /etc/passwd.id.idxnnnn.tmp where <i>nnnn</i> is the process id.
r	/etc/security/passwd
rw	/etc/security/passwd.idx
rw	/etc/security/passwd.idxnnnn.tmp where <i>nnnn</i> is the process id
r	/etc/security/lastlog
rw	/etc/security/lastlog.idx
rw	/etc/security/lastlog.idxnnnn.tmp where <i>nnnn</i> is the process id

Examples

1. To create and enable indexed read of security files, enter:

```
mkpasswd -f
```

2. To create and enable indexed read of only the /etc/security/passwd file, enter:

```
mkpasswd /etc/security/passwd.idx
```

3. To check and rebuild outdated or bad indexes, enter:

Files

Item	Description
<code>/usr/sbin/mkpasswd</code>	Contains the mkpasswd command.
<code>/etc/passwd</code>	Contains basic user attributes.
<code>/etc/security/passwd</code>	Contains user password attributes
<code>/etc/security/lastlog</code>	Contains lastlog related attributes

mkpath Command

Purpose

Adds to the system another path to an MPIO capable device.

Syntax

```
mkpath [ -l Name ] [ -p Parent ] [ -w Connection ] [ -i PathID ]
```

```
mkpath [ -l Name ] [ -p Parent ] [ -w Connection ] [ -d ]
```

```
mkpath -h
```

Description

The **mkpath** command defines, and possibly configures, one or more paths to the target device (**-l** *Name*). The paths are identified by a combination of the **-l** *Name*, **-p** *Parent*, and **-w** *Connection* flags. Both the target device and parent must be previously defined in the system to define a path. They both must be "AVAILABLE" to configure a path.

If the **-d** flag is specified, the **mkpath** command only defines the new path definition to the system. If the **-d** flag is not specified, the **mkpath** command attempts to define the path, if it does not already exist, before it attempts to configure the path. Configuring a path requires the path to already be defined and both the device and the parent device to already be configured.

The **mkpath** command displays a status message upon completion. It is possible for some paths to configure and others to fail.

Note that any device that cannot be manually defined using the **mkdev** command will not be able to have paths manually defined to using the **mkpath** command. These limitations are due to the way that path information is stored for these devices. Fiber channel devices fall into this category.

The **mkpath** command provides status messages about the results of operation. Messages in one of the following formats will be generated:

path [available | defined]

This message is displayed when **mkpath** is run on a single path. If the path is successfully configured the message "path available" is displayed. If the path is not successfully configured and there is no explicit error code returned by the method, the message "path defined" is displayed.

paths available

This message is displayed if multiple paths were identified and all paths were successfully configured.

some paths available

This message is displayed if multiple paths were identified, but only some of them were successfully configured.

no paths processed

This message is generated if no paths were found matching the selection criteria.

Flags

Item	Description
-d	Defines a new path to a device by adding a path definition to the system. The new path will not automatically be configured when the -d flag is specified. Note that only one path may be defined at a time.
-h	Displays the command usage message.
-i PathID	Indicates the path ID associated with the path to be added and is used to uniquely identify a path. This flag cannot be used with the -d flag.
-l Name	Specifies the logical device name of the target device to which the path(s) are being added. The path(s) to be added are qualified by the -p and -w flags.
-p Parent	Indicates the logical device name of the parent device associated with the path(s) to be added. This flag is required if the -d flag is specified.
-w Connection	Indicates the connection information associated with the path to be added. This flag is required if the -d flag is 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	mkpath,Define,<define method arguments>
DEV_Change	mkpath,Configure,<configure method arguments>

Examples

1. To define and configure an already defined path between scsi0 and the hdisk1 device at SCSI ID 5 and LUN 0 (i.e., connection 5,0), enter:

```
mkpath -l hdisk1 -p scsi0 -w 5,0
```

The system displays a message similar to the following:

```
path available
```

2. To configure an already defined path from '**fscsi0**' to fiber channel disk '**hdisk1**', the command would be:

```
mkpath -l hdisk1 -p fscsi0
```

The message would look similar to:

```
path available
```

3. To only add to the Customized Paths object class a path definition between **scsi0** and the **hdisk1** disk device at SCSI ID 5 and LUN 0, enter:

```
mkpath -d -l hdisk1 -p scsi0 -w 5,0
```

The system displays a message similar to the following:

path defined

Files

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

mkprojldap Command

Purpose

Configures the LDAP client and server machines for handling advanced accounting subsystem data.

Syntax

```
mkprojldap -s -h hostname -D bindDN -w bindPWD -i -p projectInstallPoint -a adminInstallPoint
```

```
mkprojldap -u -h hostname -D bindDN -w bindPWD
```

```
mkprojldap -c -D bindDN -w bindPWD [ -p accountingProjectDN ] [ -a accountingAdminDN ] [ -x cron ]
```

```
mkprojldap { -l | -L [ -D bindDN -w bindPWD ] | -V } [ -p ] [ -a ]
```

Description

The `mkprojldap` command configures the LDAP server and client machines for handling the advanced accounting subsystem data. The LDAP server and client relationship must already be defined, and `mkprojldap` makes only incremental changes. The `mkprojldap` command can be used to configure the basic LDAP connection.

To add advanced accounting support to the LDAP server, the LDAP schema for advanced accounting must be uploaded to the server. The schema describes the format of advanced accounting data to the server, enabling the server to process accounting data without being enabled specifically for accounting. This is accomplished with the `-u` option. The LDAP server is not dependent on advanced accounting. This command needs to be run only once for each LDAP server. After this command is run, use the `-s` option to define the location on the LDAP server where advanced accounting data is to be stored. This command can be run one or more times to establish one or more accounting domains. An LDAP client can only access only one accounting domain at a time.

To configure an LDAP client so that it receives advanced accounting data, use the `-c` option to specify the location of the advanced accounting data sets on the LDAP server that are to be used by the LDAP client. The `mkprojldap` command is used to configure absolute paths, which are known as *distinguished names* (DNs), to projects and admin policies. The advanced accounting subsystem stores project definitions and admin policies on LDAP servers, so there are two advanced accounting DN's that can be configured. The `mkprojldap -c` command must be run on each client.

Flags

Item	Description
<code>-a <i>accountingAdminDN</i></code>	Specifies the accounting admin DN location on the LDAP server, when used with <code>-s</code> or <code>-c</code> options. When used with <code>-l</code> or <code>-L</code> options, this flag displays the accounting admin DN.
<code>-c</code>	Configures the LDAP client.
<code>-D <i>bindDN</i></code>	Specifies the Bind DN to be used during the server configuration.

Item	Description
-h <i>hostname</i>	Specifies the host name of the LDAP server during the server configuration.
-i	Provides the admin (-a) and project (-p) install points during the server configuration.
-L	Displays the potential accounting DNs that are visible from the server.
-l	Displays the accounting DNs in the <code>ldap.cfg</code> file.
-p <i>accountingProjectDN</i>	Specifies the accounting project DN location on the LDAP server when used with the -s or -c options. When used with -l or -L options, this flag displays the accounting project DN.
-r <i>con</i>	Specifies the frequency for refreshing the LDAP repositories (hourly, daily, or off).
-s	Configures the LDAP server.
-u	Uploads the advanced accounting schema to the LDAP server.
-V	Displays the current LDAP client configuration details in a colon separated format.
-w <i>bindPWD</i>	Used to provide the Bind password for the Bind DN specified with the -D option.

Note: When using the preceding flags with this command, use the following guidelines:

- During server and client configuration, both the -p and -a arguments can be specified at the same time, but neither is required. If neither is specified, the `mkprojldap` command tries to compute the missing accounting DNs by searching for the objects on the LDAP server. These objects are `ou=projects` and `ou=adminpolicy`. If an object is found, the corresponding accounting DN is computed and added to the `ldap.cfg` file.
- While listing the accounting DNs using the -l or -L options, both -p and -a can be used. If neither of them are provided, all accounting DNs in the `ldap.cfg` file are listed.
- The colon-separated data displayed by the -V option takes the following format:

```
ldap-server-hostname:bind DN:bind password:default-projectdn:default-admindn:cron
```

Exit Status

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

Examples

1. To upload the advanced accounting schema, type:

```
mkprojldap -u -h mozilla -D cn=root -w mozillapasswd
```

2. To configure the LDAP server, type:

```
mkprojldap -s -h ldap.svr.com -D cn=root -w passwd -i  
-p cn=aixdata,o=ibm -a cn=aixdata,o=ibm
```

This command creates two DNs in the following format:

```
ou=projects,ou=aacct,cn=aixdata,o=ibm and ou=adminpolicy,ou=aacct,cn=aixdata,o=ibm
```

3. To configure the LDAP client, type:

```
mkprojldap -c -D cn=testroot -w testpwd -p ou=projects,ou=aacct,ou=cluster1,cn=aixdata -a  
ou=adminpolicy,ou=aacct,ou=cluster1,cn=aixdata -r hourly
```

4. To display the currently configured accounting DNs, type:

```
mkprojldap -l
```

Files

Item	Description
/usr/sbin/mkprojldap	Contains the mkprojldap command.
/etc/security/ldap/ldap.cfg	Contains the LDAP configuration data.
/etc/security/ldap/sec.ldif	Contains the LDAP schema for advanced accounting.

mkproto Command

Purpose

Constructs a prototype file system.

Syntax

mkproto *Special Prototype*

Description

The **mkproto** command is used to construct a prototype for a new file system. It exists solely for Berkeley Software Distribution (BSD) compatibility.

The *Special* parameter can be a block device name, raw device name, or file system name. The *Prototype* parameter is the name of the prototype file that specifies the structure and contents of the file system to be created. The **mkproto** command calls the **mkfs** command with the *Prototype* and *Special* parameters.

Prototype Files

The **mkproto** and **mkfs** commands require an extended prototype file to create a Journaled File System (JFS). A *prototype file* is a formatted listing of the contents and structure of a file system. A prototype file describes the file system by a series of tokens separated by spaces and new lines. The main body of a prototype file defines the objects of the file system.

A JFS prototype file consists of the main body, which can be created by the **proto** command, preceded by five special tokens. These five tokens are defined as follows:

Prototype Files

Item	Description
1st token	Name of a file to be copied onto block 0 as the bootstrap program or the special token <noboot>.

Prototype Files (*continued*)

Item	Description
2nd token	Size of the file system. For a JFS, the size is expressed in units of 512-byte blocks. If the 2nd token is 0, the mkfs command creates the file system to fill the entire logical volume.
3rd token	Number of i-nodes on the file system. This token is not used by a JFS but must be provided to preserve the position.
4th token	Size of the file system fragment in bytes. If the 4th token is 0 (zero), the mkfs command uses the default fragment size. For JFS, the token must be either 0 (default value used), 512, 1024, 2048, or 4096. The default fragment size is 4096 for a JFS. An invalid fragment size causes the mkfs command to fail.
5th token	Number of bytes per i-node (nbpi). If this token is 0, the mkfs command uses the default nbpi. For a JFS, this token must be either 0 (default value used), 512, 1024, 2048, 4096, 8192, or 16384. The default number of bytes per i-node is 4096 for a JFS. An invalid nbpi causes the mkfs command to fail.

The remaining tokens define the contents and structure of the file system. These tokens are grouped into sets, with each set defining one object of the file system. The syntax of each set is as follows:

```
{ [Name] { - | { - | d | b | c | l | L | p } { - | u } { - | g } { - | t } Mode Owner Group { Major Minor | SourceFile | DirectoryListing } } | { $ }
```

where:

Tokens

Item	Description
<i>Name</i>	Specifies the name of the object as it is to appear in the new file system. The <i>Name</i> token is required for every object except for the root directory definition.

Item	Description
<code>{- d b c l L p}{- u}{- g}{- t}</code>	<p>Represents a string of 4 positional characters, where:</p> <p><code>{- d b c l L p}</code> Defines the object type. Valid types are:</p> <ul style="list-style-type: none"> - Regular file d Directory b Block special file c Character special file l Symbolic link L Hard link p Named pipe <p><code>{- u}</code> Toggles the set UID bit of the object, as follows:</p> <ul style="list-style-type: none"> u Set UID on execution - Do not set UID on execution <p><code>{- g}</code> Toggles the set group ID (GID) bit of the object, as follows:</p> <ul style="list-style-type: none"> g Set GID on execution - Do not set GID on execution <p><code>{- t}</code> Toggles the sticky bit of the object, as follows:</p> <ul style="list-style-type: none"> t Sticky bit on - Sticky bit off <p>This 4-character token is required for every object.</p>
<i>Mode</i>	<p>Represents a string of 3 octal characters defining the read, write, and execute permissions of the object. The <i>Mode</i> token is required of every object. See the <code>chmod</code> command for more information about permissions.</p>
<i>Owner</i>	<p>Specifies the UID of the owner of the object. The owner token is required for every object.</p>

Item	Description
<i>Group</i>	Specifies the GID of the owner of the object. The group token is required for every object.
<i>Major Minor</i>	Specifies the major and minor device numbers of the object if its type is a block or character special file. If the object is not a block or character special file, these tokens are omitted.
<i>SourceFile</i>	Applies only to regular file, hard link, and symbolic link objects. For regular files, this token is the path name to the file from which the object file is to be initialized. For both symbolic and hard links, this token is the source of the link. The source of the link is relative to the new file system for hard links.
<i>DirectoryListing</i>	Defines the contents of the object if it is a directory. The contents of the directory are defined using the token syntax described here. For example, a directory listing can include one or more regular files, one or more block files, and one or more directory listings. The mkfs command creates the directory entries . (dot) and .. (dot dot). Each directory listing is terminated with the special \$ token.
\$	Ends the current directory listing or indicates the end of the prototype file.

Example Prototype Specification

The following prototype specification describes a JFS that does not have a boot program in block 0 and occupies the entire device. The 3rd token is ignored. The 4th and 5th tokens define the fragment size as 1024 bytes and the number of bytes per i-node as 2048. The main body of this prototype defines the file system contents.

```
<noboot> 0 0 1024 2048
d--- 755 0 0
dir1 d--- 755 0 2
  block_dev b--- 644 0 0 880 881
  char_dev c--- 644 0 0 990 991
  named_pipe p--- 644 0 0
  regfile3 ---- 644 0 0 /tmp/proto.examp/dir1/regfile3
  regfile4 ---- 644 0 0 /tmp/proto.examp/dir1/regfile4
$
dir2 d--- 755 205 300
  regfile6 ---- 644 0 0 /tmp/proto.examp/dir2/regfile6
  symlnOutofFS l--- 644 0 0 /tmp/proto.examp/dir2/regfile6
  symlnNoExist l--- 644 0 0 /home/foobar
  symlnInFs l--- 644 0 0 /dir2/regfile6
  regfile5 ---- 644 0 0 /tmp/proto.examp/dir2/regfile5
  hardlink L--- 644 0 0 /dir2/regfile5
$
dir3 d--- 755 0 0
  setgid --g- 755 0 0 /tmp/proto.examp/dir3/setgid
  setuid -u-- 755 0 0 /tmp/proto.examp/dir3/setuid
  sticky ---t 755 0 0 /tmp/proto.examp/dir3/sticky
$
dir4 d--- 755 0 0
dir5 d--- 755 0 0
  dir6 d--- 755 0 0
  $
  dir7 d--- 755 0 0
  $
  $
  regfile7 ---- 644 0 0 /tmp/proto.examp/dir4/regfile7
$
```



```
regfile1 ---- 555 205 1 /tmp/proto.examp/regfile1
regfile2 ---- 744 0 0 /tmp/proto.examp/regfile2
$
$
```

Three entries for the dir2 object deserve further examination:

dir2 object

Item	Description
symlnOutofFS l-- 644 0 0 /tmp/ proto.examp/dir2/regfile6	This entry defines a symbolic link to a file outside the file system to be created. The command <code>ls -l</code> lists something similar to <code>symlnOutofFS -> /tmp/proto.examp/dir2/regfile6</code> .
symlnNoExist l-- 644 0 0 /home/foobar	This entry defines a symbolic link to a file outside the file system to be created to a file that does not exist. The command <code>ls -l</code> lists something similar to <code>symlnNoExist -> /home/foobar</code> .
symlnInFs l-- 644 0 0 /dir2/regfile6	This entry defines a symbolic link to a file within the file system to be created. The command <code>ls -l</code> lists something similar to <code>symlnInFS -> /dir/regfile6</code> .

Examples

To make a prototype JFS using the prototype file described in the "[Example Prototype File Specification](#)" :

1. Generate the main body of the prototype file using the **proto** command or a text editor. For the purposes of this example, call the file `/tmp/ProtoFile`.
2. Add the first 5 tokens as required for a JFS. In the example prototype file, the tokens are:

```
<noboot> 0 0 1024 2048
```

3. Create a logical volume to hold the file system, as follows:

```
mklv -y protolv -t jfs SomeVGname 5
```

This command creates a logical volume named `protolv` in the `SomeVGname` volume group. The size of the logical volume is 5 logical partitions.

4. Add an appropriate stanza to the `/etc/filesystem` file. A minimal example stanza is:

```
/protofs:
dev       = /dev/protolv
vfs       = jfs
log       = /dev/loglv00
mount     = false
```

5. Run the following **mkproto** command:

```
mkproto /dev/protolv /tmp/ProtoFile
```

This command creates a JFS on the `protolv` logical volume. The size of the JFS is 5 logical partitions, its fragment size is 1024 bytes, and its nbpi ratio is 2048. The structure and contents of the file system are as specified in the prototype file `/tmp/ProtoFile`.

Files

Files

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

mkprtdap Command

Purpose

Configures IBM Directory (LDAP) for Directory enabled System V print. It also configures client machines to use the Directory for System V print information.

Syntax

To configure the IBM Directory to store System V Print information:

```
mkprtdap -s -a AdminDN -p Adminpasswd -w ACLBindPasswd [-f] [-d nodeDN]
```

To configure clients to use the IBM Directory for System V Print information:

```
mkprtdap -c -h DirectoryServerHostname -w ACLBindPasswd [ -d PrintBindDN ] [-U]
```

To get usage information for the **mkprtdap** command:

```
mkprtdap ?
```

Description

The **mkprtdap** command configures the IBM Directory (LDAP) server, and one or more clients to use the Directory (LDAP) for System V Print information. This command must be run on the system being setup as the server and on all the client systems. Once the Directory (LDAP) server is configured for System V print, the directory enabled System V Print commands (**dslpadmin**, **dslpaccess**, **dslpsearch**, **dslpenable**, **dslpdisable**, **dslpaccept**, **lp**, **lpstat**, **cancel** and **dslpreject**) must be run to add , remove and manage System V print information (printers and print queues) on the Directory (LDAP) server. The **mkprtdap** command configures client machines to use the Directory (LDAP) server for System V print information.

The **mkprtdap** command requires the IBM Directory server software to be installed on the machine being configured as the server. The command also requires the IBM Directory client software to be installed on all client machines that will use the Directory (LDAP) server for System V print information.

Note: The client (**-c** flag) and the server (**-s** server) options cannot be run at the same time. When setting up a system as the server, the **mkprtdap** command should be run twice on that system. Once to set up the server, and again to set up the client.

During the server side configuration, using the **-s** flag, the **mkprtdap** command:

- Requires the IBM Directory Administrator's DN and password if the Directory has been configured. If the Directory Administrator's DN and password have not been set, **mkprtdap** will set them with the values passed to the command.
- Creates the AIX Information tree DN (cn=aixdata container object) on the Directory if one is not present. The print subtree will be created under the AIX Information subtree. If an existing AIX Information subtree exists on the Directory, the print subtree will be created under it. All System V print information will be stored under the print subtree. The directory enabled System V print commands have to be run to add printers and print queues under the print subtree created.
- The default suffix and AIX Information tree for the **mkprtdap** command is a top level container object cn=aixdata. The Print subtree (ou=print) will be created under the AIX Information tree.
- The print subtree is ACL protected with the value of the **ACLBindPasswd** parameter passed to the command. The same value must be used when configuring clients to use the Directory for System V

print information. Select a password value that is difficult for people or password cracking programs to guess.

- If the **-d** option is used and a valid existing node on the Directory is passed to the command, the AIX Information subtree is created under the given node. The print subtree is then created under the AIX Information subtree.
- Starts the IBM Directory server after all the above is done
- Adds the IBM Directory server process (slapd) to the **/etc/inittab** file to have the server start after a reboot.

During the client configuration, the **mkprtdap** command:

- Saves the IBM Directory (LDAP) server host name in the **/etc/ldapsvc/server.print** file.
- Saves the AIX Print Bind DN in the **/etc/ldapsvc/server.print** file.
- Saves the ACL Bind Password for the AIX Print Bind DN in the **/etc/ldapsvc/system.print** file. The value of the ACL Bind password must be the same as the one specified during the configuration of the Directory server.
- Undo a previous client configuration if the **-U** flag is specified. This option will replace the **/etc/ldapsvc/system.print** and **/etc/ldapsvc/server.print** files with the previous saved copies of the files (**/etc/ldapsvc/server.print.save** and **/etc/ldapsvc/system.print.save**).

Flags

Server

Item	Description
-a <i>AdminDN</i>	Specifies the Directory (LDAP) Administrator's DN.
-d <i>nodeDN</i>	This advanced option requires a valid existing node DN on the Directory under which the AIX Information tree and Print Subtree will be created.
-f	The force flag is required by the mkprtdap command to force the creation of the Print subtree (and AIX Information subtree if needed) when one or more AIX Information trees exist on the Directory.
-p <i>adminpasswd</i>	Specifies the Directory (LDAP) Administrator's password.
-s	Indicates the command is being run to configure the Directory for System V print.
-w <i>ACLBindPasswd</i>	Specifies the password to ACL protect the Print Subtree on the Directory. Select a password value that is difficult for people or password cracking programs to guess.

Client

Item	Description
-c	Indicates the command is being run to configure clients to use the Directory for System V Print information.
-d <i>PrintBindDN</i>	Specifies the Print Bind DN. The default Print Bind DN is ou=print,cn=aixdata . The Print Bind DN to use during Client configuration is displayed at the end of the server setup of the mkprtdap command.
-h <i>DirectoryServerHostname</i>	Hostname of the IBM Directory server setup to store System V Print information.
-U	Undo a previous configuration of a client.

Item	Description
-w <i>ACLBindPasswd</i>	The ACL Bind Password for the print subtree. The ACL Bind password is specified during the server setup of the mkprtldap command. The value of the ACL Bind Password must match the one used during the setup of the Directory server.

Usage

Item	Description
?	Displays usage information for the mkprtldap command.

Security

This command can be run by the root user only.

Examples

1. To configure System V print on a machine with a configured IBM Directory server -

The Administrator DN and password are required to configure System V print on the Directory. Assume the existing Administrator's DN and password are **cn=admin** and **passwd**.

```
mkprtldap -s -a cn=admin -p passwd -w pass123wd
```

2. The **mkprtldap** command provides the option to configure the IBM Directory to store the print information under a pre-existing node (e.g. **o=ibm,c=us**) on the Directory [Advanced Option]. This is only recommended when it is necessary to store the print information under the existing node on the Directory for specific reasons. The recommend option is to store the print subtree in the default location on the Directory by not specifying the **-d** option. The Administrator DN and password are required to configure System V print on the Directory Assume the existing Administrator's DN and password are **cn=admin** and **passwd**.

```
mkprtldap -a cn=admin -p passwd -w ac1123passwd -d o=ibm,c=us
```

Running the command will create an AIX Information tree (**cn=aixdata**) under the **o=ibm,c=us** object. The print subtree will be created under this new object (**cn=aixdata, o=ibm, c=us**).

3. To configure System V print on a machine with a configured IBM Directory server and an existing AIX Information tree. There might be situations where the Directory contains an existing AIX information tree with other subsystem specific information (e.g Security or NIS information). It might be required to store the print information in a separate location on the Directory under a different AIX Information tree. The command, by default, will not create a new AIX Information tree if one exists on the Directory. To force the command to create a new AIX Information tree to store the print information, use the **-f** flag with the command. Consider the case where the Security and NIS subsystem information is stored under the AIX Information tree at **cn=aixdata,o=ibm,c=us**. To create a new AIX Information tree for print information different from the existing one, run the command with the **-f** flag and specify the default location or another node. The Administrator DN and password are required to configure System V print on the Directory. Assume the existing Administrator's DN and password are **cn=admin** and **passwd**:

```
mkprtldap -a cn=admin -p passwd -w passwd123 -f
```

Running the command will create a new AIX Information tree (**cn=aixdata**) with the suffix (**cn=aixdata**) and the print information will be stored under this new AIX Information tree (**ou=print, cn=aixdata**). There will be two AIX Information trees on the Directory in this example **cn=aixdata,o=ibm,c=us** and **cn=aixdata**. The print information will be under the **cn=aixdata** object (suffix - **cn=aixdata**). For **mkprtldap**, it is recommend to use the default location to add the print information to the Directory.

4. To configure a client to use an IBM Directory setup for System V Print on host **server.ibm.com**, type:

```
mkprtlldap -c -h server.ibm.com -w passwd
```

Please ensure that the ACL Bind Password (**passwd**) is the same as the one specified during the setup of the Directory Server. Running the command without specifying a Print Bind DN value with the **-d** option will cause the command to use the default Print Bind DN **ou=print,cn=aixdata**. The Print Bind DN must match the one displayed at the end of running the **mkprtlldap** command to configure the server.

5. To change the information in the client side configuration files, run the **mkprtlldap** command with the new information

```
mkprtlldap -c -h server.ibm.co.uk -w aclpasswd -d ou=print,cn=aixdata,c=uk
```

Executing this command on a client that has already been configured will change the information in the **/etc/ldapsvc/server.print** and **/etc/ldapsvc/system.print** files to contain the new configuration information. The original contents of the **/etc/ldapsvc/server.print** and **/etc/ldapsvc/system.print** will be stored in the **/etc/ldapsvc/server.print.save** and **/etc/ldapsvc/system.print.save** files.

Files

Mod	File	Description
rw	/etc/slapd32.conf	(Server configuration) - Contains the IBM Directory (LDAP Version 5.2) configuration information.
rw	/home/ldapdb2/idsslapd-ldapdb2/etc/ibmslapd.conf	(Server configuration) - Contains the IBM Directory (LDAP Version 6.0 or later) configuration information.
rw	/etc/ldapsvc/server.print	(Client configuration) - Contains information about the Directory Server configured to store System V Print information. (Machine name, Location of Print subtree on the Directory and LDAP port)
rw	/etc/ldapsvc/system.print	(Client configuration) - Contains the ACL Bind Password for the Print subtree on the Directory.

mkprtsv Command

Purpose

Configures TCP/IP-based print service on a host.

Syntax

To Configure and Start Print Service for a Client Machine

```
mkprtsv -c [ -S ] [ -q QueueName -v DeviceName -b "Attribute=Value ..." -a "Attribute=Value ..." | -A FileName ]
```

To Configure and Start Print Service for a Server Machine

```
mkprtsv -s [ -S ] [ -q QueueName -v DeviceName -b "Attribute=Value ..." -a "Attribute=Value ..." | -A FileName ] [ -h "HostName ..." | -H FileName ]
```

Description

The **mkprtsv** high-level command configures a TCP/IP-based print service on a host. The print service configuration can be done for a host functioning as a client or for a host functioning as a server.

Use the command to configure and start the print service.

To configure print service for a client, the **mkprtsv** command calls the spooler **mkque** and **mkquedev** commands to change the **/etc/lpd/qconfig** file (or its object class equivalent) appropriately and set up a spooler queue on the client machine.

To configure print service for a server, the **mkprtsv** command does the following:

1. Calls the **ruser** command to set up remote users to print on the server.
2. Calls the **mkque** and **mkquedev** commands to change the server's **/etc/lpd/qconfig** file appropriately and set up the necessary device queues on the server machine.
3. Calls the **startsrc** command to activate the **lpd** and **qdaemon** server daemons. The **qdaemon** server daemon starts the **pio** printer backend.

Flags

Item	Description
-A <i>FileName</i>	Specifies name of file containing entries related to the qconfig file.
-a " <i>Attribute =Value...</i> "	Specifies a list of attributes and their corresponding values to be used for updating the spooler's qconfig file or object class. The -a flag is optional. Valid attribute types are listed below: acctfile (true/false) Identifies the file used to save print command accounting information. The default value of false suppresses accounting. If the named file does not exist, no accounting is done. argname Specifies the logical printer name. device Identifies the symbolic name that refers to the device stanza. discipline Defines the queue-serving algorithm. The default value of fcfs means first come, first served. A sjn value means shortest job next. pserver Specifies the remote print server. up (true/false) Defines the state of the queue. The default value of true indicates that it is running. A false value indicates that it is not.

Item	Description
-b " <i>Attribute =Value...</i> "	<p>Specifies a list of attributes and their corresponding values to be used for updating the spooler's qconfig file or object class. At least one attribute must be defined for the -b option. The backend attribute is required. Valid attribute types are listed below:</p> <p>access (true/false) Specifies the type of access the backend has to the file specified by the file attribute. The access attribute has a value of write if the backend has write access to the file, or a value of both if the backend has both read and write access. This field is ignored if the file field has a value of false.</p> <p>align (true/false) Specifies whether the backend sends a form-feed control before starting the job if the printer has been idle. The default value is false.</p> <p>backend Specifies the full path name of the backend, optionally followed by flags and parameters to be passed to it. The backend attribute is required.</p> <p>feed Specifies the number of separator pages to print when the device becomes idle, or takes a never value, which indicates that the backend is not to print separator pages.</p> <p>file Identifies the special file where the output of the backend is to be redirected. The default value of false indicates no redirection. In this case, the backend opens the output file.</p> <p>header (never/always/group) Specifies whether a header page prints before each job or group of jobs. The default value of never indicates no header page. To produce a header page before each job, specify an always value. To produce a header before each group of jobs for the same user, specify a group value.</p> <p>trailer (never/always/group) Specifies whether a trailer page prints after each job or group of jobs. The default value of never indicates no trailer page. To produce a trailer page after each job, specify an always value. To produce a trailer after each group of jobs for the same user, specify a group value.</p> <p>host Specifies the host name from which to print.</p> <p>s_statfilter Translates short queue-status information to a format recognized by this operating system.</p> <p>l_statfilter Translates long queue-status information to a format recognized by this operating system.</p>
-c	Performs print service configuration for a client machine. Use the -q flag with the -c option.
-H <i>FileName</i>	Specifies the name of a file containing a list of host names.
-h " <i>HostName...</i> "	Specifies a list of host names to be included in the list of remote users who can use the print server. The queuing system does not support multibyte host names.
-q <i>QueueName</i>	Specifies the name of a queue in the qconfig file.

Item	Description
-S	Starts print service after it is configured. If the -S flag is omitted, print service is configured but not started.
-s	Performs print service configuration for a server machine. Use the -h , -H , and -q flags with the -s flag.
-v DeviceName	Specifies the name of the device stanza in the qconfig file.

Examples

1. To configure and enable print service for a client, enter the command in the following format:

```
mkprtsv -c -S -a"argname=rp1 backend=piobe \  
pserver=print802"
```

In this example, `rp1` is the logical printer name, `piobe` is the printer backend, and `print802` is the remote print server.

2. To configure a print server using initialization information and allow remote printing, enter the command in the following format:

```
mkprtsv -s -H hnames -A qinfo
```

In this example, attribute information stored in the `qinfo` file initializes the spooler, and the list of host names stored in the `hnames` file is the list of remote hosts that have access rights to the print server.

Files

Item	Description
<u>/etc/lpd/qconfig</u>	Contains configuration information for the printer queuing system.

mkps Command

Purpose

Adds an additional paging space.

Syntax

To add a logical volume for additional paging space

```
mkps [ -t lv | [ps_helper psname] ] [ -a ] [ -n ] [ -k ] [ -c ChksumSize ] -s LogicalPartitions VolumeGroup  
[ PhysicalVolume ]
```

To add additional paging space on an NFS server

```
mkps [ -a ] [ -n ] -t nfs ServerHostName ServerFileName
```

Description

The **mkps** command adds an additional paging space. Before the paging space can be used it must be activated, using the **swapon** command. The *VolumeGroup* parameter specifies the volume group within which the logical volume for the paging space is to be made. The *PhysicalVolume* parameter specifies the physical volume of the *VolumeGroup* on which the logical volume is to be made.

Note: A paging space larger than 2 GB is possible when using NFS (Network File System) v4 rather than the default UDP (User Datagram Protocol) or the NFSv2 protocol. NFSv2 swapping can only handle swap file size up to 2 GB due to NFSv2 protocol limitation.

In the second form of the **mkps** command, the *ServerHostName* parameter specifies the NFS server where the *ServerFileName* resides. The *ServerFileName* specifies the file which will be used for the NFS paging of the system. The *ServerFileName* file must exist and be exported correctly to the client that will use the file for paging.

When adding a NFS paging space, the client attempts to contact the server using UDP and then TCP. The method that succeeds first is used to contact the server when accessing that paging space.

If the **-t** flag is specified, the argument will be assumed to be a third-party helper executable. If the helper executable is present in the */sbin/helpers/pagespace* path then it will be spawned passing all the arguments and with the **-m** flag to specify **mkps** command. An entry will be added into */etc/swapspace* path if the helper executable returns zero. In this case, if *psname* starts with */*, it is considered to be absolute path of device entry, or else */dev* is prepended to the *psname*. The helper executable must take care of creating the device, making it pageable and adding an entry into ODM. If the helper program doesn't exist in the */sbin/helpers/pagespace* directory the **mkps** command will display the usage error. The helper executable must exit with a 0 if successful and a non-zero if it fails.

You can use the System Management Interface Tool (SMIT) **smit mkps** fast path to run this command.

Flags

Item	Description
-a	Specifies that the paging space is configured at subsequent restarts.
-c	Specifies the size of the checksum to use for the paging space, in bits. Valid options are 0 (checksum disabled), 8, 16 and 32. If -c is not specified it will default to 0.
> > -k	<p>Enables the data encryption in the logical volume for the paging space. The -k flag is available in IBM AIX 7.3 with Technology Level 1, or later.</p> <p>The mkps command initializes the primary encryption key of the logical volume for the paging space. You cannot use the -k flag if data encryption is not enabled for the volume group in which the paging space is created. If you attempt to create an encrypted paging space on such a volume group, the mkps command fails with an error message.</p>
-n	Activates the paging space immediately.
-s LogicalPartitions	Specifies the size of the paging space and the logical volume to be made in logical partitions.
-t	<p>Specifies the type of paging space to be created. One of the following variables is required:</p> <p>lv Specifies that a paging space of type logical volume should be created on the system.</p> <p>nfs Specifies that a paging space of type NFS should be created on the system.</p> <p>ps_helper Name of the helper program for a third party device.</p> <p>psname Name of the device entry for paging space.</p>

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 paging space in volume group myvg that has four logical partitions and is activated immediately and at all subsequent system restarts, enter the following command:

```
mkps -a -n -s4 myvg
```

2. To create an NFS paging space on the NFS server swapserve where the host swapserve has the /export/swap/swapclient file exported, enter the following command:

```
mkps -t nfs swapserve /export/swap/swapclient
```

3. To create a paging space myps by using helper executable test, enter the following command:

```
mkps -t test /dev/myps -s1 myvg mydisk
```

4. **>** To create an encrypted paging space in myvg volume group that is encrypted and activated and has logical partitions such that the paging space is configured at all subsequent system restart operations, enter the following command:

```
mkps -a -k -n -s4 myvg
```

[|<](#)

Files

Item	Description
/etc/swapspace	Specifies the paging space devices and their attributes.

mkqos Command

Purpose

Configures the system to support QoS.

Syntax

```
/usr/sbin/mkqos [ -I | -N | -B ]
```

Description

The **mkqos** command configures the system to support Quality of Service (QoS).

Flags

Item	Description
-B	Adds an entry to the inittab file to execute the /etc/rc.qos file now and on the next system restart. This flag is the default.

Item	Description
-I	Adds an entry to the inittab file to execute the /etc/rc.qos file on the next system restart.
-N	Executes the /etc/rc.qos file to start the QoS daemons. When invoked in this way, the QoS daemons run until the next system restart.

Files

Item	Description
inittab	Controls the initialization process of the system.
/etc/rc.qos	Contains the startup script for the QoS daemons.

mkque Command

Purpose

Adds a printer queue to the system.

Syntax

mkque [**-D**] **-q** *Name* [**-a** '*Attribute = Value*' ...]

Description

The **mkque** command adds a printer queue to the system by adding the stanza described on the command line to the end of the **/etc/qconfig** file.

You can use the System Management Interface Tool (SMIT) **smit mkque** fast path to run this command.

To use the SMIT fast path to go directly to the **Add a Local Queue** dialog, enter:

```
smit mklque
```

To use the SMIT fast path to go directly to the **Add a Remote Queue** dialog, enter:

```
smit mkrque
```

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
<code>-a 'Attribute = Value'</code>	Specifies a line to be added to the queue stanza in the <code>/etc/qconfig</code> file. This flag must be the last flag when entering the <code>mkque</code> command on the command line. For a list of all valid attributes, see the <code>/etc/qconfig</code> file. Note: It is recommended that you do not use the <code>'device = '</code> attribute. This attribute is handled automatically by the <code>mkquedev</code> command. Also note that the queuing system does not support multibyte host names.
<code>-D</code>	Specifies that the queue defined by the <code>Name</code> variable queue is added to the top of the <code>/etc/qconfig</code> file and is therefore the default queue. If you do not specify this flag, the <code>Name</code> variable is added to the bottom of the <code>/etc/qconfig</code> file and is not the default queue.
<code>-q Name</code>	Specifies the name of the queue to be added. Note: The queue name must not exceed 20 characters.

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 add the print queue `lp0` specifying a host name of `leo` and a remote print queue named `lp013`, enter:

```
mkque -q lp0 -a 'host = leo' -a 'rq = lp013'
```

Files

Item	Description
<code>/usr/bin/mkque</code>	Contains the <code>mkque</code> command.
<code>/etc/qconfig</code>	Configuration file.

mkquedev Command

Purpose

Adds a printer queue device to the system.

Syntax

```
mkquedev -d Name -q Name -a 'Attribute = Value' ...
```

Description

The **mkqueuedev** command adds a printer queue device to the system by adding the stanza described on the command line to the **/etc/qconfig** file.

You can use the System Management Interface Tool (SMIT) **smit mkqueuedev** 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
-a <i>'Attribute = Value'</i>	Specifies the <i>'Attribute = Value'</i> attribute to be added to the device stanza in the /etc/qconfig file. This flag must be the last flag when entering the mkqueuedev command on the command line. For a list of valid attributes, see the /etc/qconfig file. Note: The 'backend = ' attribute must be included when entering this command on the command line.
-d <i>Name</i>	Specifies with the <i>Name</i> variable the name of the queue device to add. Note: The queue device name must not exceed 20 characters.
-q <i>Name</i>	Specifies with the <i>Name</i> variable the name of the queue (this name must already exist) to which the queue device is added. The mkqueuedev command automatically adds the 'device = ' attribute to the specified 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 that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

To add the postscript print queue device to the lp0 queue, specify the backend program to be the **piobe** command (**backend = /usr/lib/lpd/piobe**) and direct the backend program not to align the paper (**align = FALSE**), enter:

```
mkqueuedev -qlp0 -dpostscript -a 'backend = /usr/lib/lpd/piobe' \  
-a 'align = FALSE'
```

Files

Item	Description
/usr/bin/mkqueuedev	Contains the mkqueuedev command.
/etc/qconfig	Contains the configuration file.

mkramdisk Command

Purpose

Creates a RAM disk using a portion of RAM that is accessed through normal reads and writes.

Syntax

```
mkramdisk [ -u ] size[ M | G ]
```

Description

The **mkramdisk** command is shipped as part of **bos.rte.filesystems**, which allows the user to create a RAM disk. Upon successful execution of the **mkramdisk** command, a new RAM disk is created, a new entry added to **/dev**, the name of the new RAM disk is written to standard output, and the command exits with a value of 0. If the creation of the RAM disk fails, the command prints an internalized error message, and the command will exit with a nonzero value.

The size can be specified in terms of MB or GB. By default, it is in 512 byte blocks. A suffix of *M* will be used to specify size in megabytes and *G* to specify size in gigabytes.

The names of the RAM disks are in the form of **/dev/rramdiskx** where *x* is the logical RAM disk number (0 through 63).

The **mkramdisk** command also creates block special device entries (for example, **/dev/ramdisk5**) although use of the block device interface is discouraged because it adds overhead. The device special files in **/dev** are owned by root with a mode of 600. However, the mode, owner, and group ID can be changed using normal system commands.

Up to 64 RAM disks can be created.

Note: The size of a RAM disk cannot be changed after it is created.

The **mkramdisk** command is responsible for generating a major number, loading the ram disk kernel extension, configuring the kernel extension, creating a ram disk, and creating the device special files in **/dev**. Once the device special files are created, they can be used just like any other device special files through normal **open**, **read**, **write**, and **close** system calls.

RAM disks can be removed by using the **rmramdisk** command. RAM disks are also removed when the machine is rebooted.

By default, RAM disk pages are pinned. Use the **-u** flag to create RAM disk pages that are not pinned.

Flags

Item	Description
-u	Specifies that the ram disk that is created will not be pinned. By default, the ram disk will be pinned.

Parameters

Item	Description
<i>size</i>	Indicates the amount of RAM (in 512 byte increments) to use for the new RAM disk. For example, typing: <pre>mkramdisk 1</pre> creates a RAM disk that uses 512 bytes of RAM. To create a RAM disk that uses approximately 20 MB of RAM, type: <pre>mkramdisk 40000</pre>

Exit Status

The following exit values are returned:

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 create a new ram disk using a default 512-byte block size, and the size is 500 MBs (1048576 * 512), enter:

```
mkramdisk 1048576  
/dev/rramdisk0
```

The **/dev/rramdisk0** ramdisk is created.

2. To create a new ramdisk with a size of 500 Megabytes, enter:

```
mkramdisk 500M  
/dev/rramdisk0
```

The **/dev/rramdisk0** ramdisk is created. Note that the ramdisk has the same size as example 1 above.

3. To create a new ram disk with a 2-Gigabyte size, enter:

```
mkramdisk 2G  
/dev/rramdisk0
```

4. To set up a RAM disk that is approximately 20 MB in size and create a JFS file system on that RAM disk, enter the following commands:

```
mkramdisk 40000  
ls -l /dev | grep ram  
mkfs -V jfs /dev/ramdiskx  
mkdir /ramdisk0  
mount -V jfs -o nointegrity /dev/ramdiskx /ramdiskx
```

x is the logical RAM disk number.

To set up a RAM disk that is approximately 20 MB in size and create a JFS2 file system on that RAM disk, enter the following commands:

```
mkramdisk 40000
ls -l /dev | grep ram
/sbin/helpers/jfs2/mkfs -V jfs2 /dev/ramdiskx
mkdir /ramdiskx
mount -V jfs2 -o log=NULL /dev/ramdiskx /ramdiskx
```

x is the logical RAM disk number.

Note: For both JFS and JFS2, for using a file system on a RAM disk, the RAM disk must be pinned.

Files

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

mkresponse Command

Purpose

Creates a new response definition.

Syntax

To create a response with no actions:

```
mkresponse [-b] [-p node_name] [-h] [-TV] response
```

To create a response with one action:

```
mkresponse -n action [-d days_of_week[,days_of_week...]] [-t time_of_day[,time_of_day...]] [-s action_script [-r return_code] [-b] [-e a | A | b | e | r]] [-o] [-E env_var=value[,env_var=value...]] [-u] [-p node_name] [-h] [-TV] response
```

To copy a response:

```
mkresponse -c existing_response[:node_name] [-p node_name] [-h] [-TV] response
```

Description

The **mkresponse** command creates a new response definition with the name specified by the *response* parameter. One action can also be specified when the response is defined. Actions define commands to be run when the response is used with a condition and the condition occurs. The action defines days of the week when the action can be used, the time of day for those days of the week, the script or command to be run, what type of event causes the command to be run, the expected return code of the script or command, and whether to keep standard output. The days and times are paired so that different times can be specified for different days. A response with no actions only logs the events.

Use the **-b** flag to specify that the response, and all actions to be defined in this response, support event batching. For event batching, multiple events can be batched or grouped together and passed to a response. The actions of the response are directed to a file that contains the details for the batched events. A response that supports event batching can only be used for conditions that specify the events are to be batched. The **-b** flag cannot be specified with the **-e** flag.

In a cluster environment, use the **-p** flag to specify the node in the domain that is to contain the response definition. If you are using **mkresponse** on the management server and you want the response to be defined on the management server, do *not* specify the **-p** flag. If the **-p** flag is not specified, the response is defined on the local node.

Use the **chresponse** command to add actions to a response or to remove actions from a response. Use the **startcondresp** command to start monitoring. The **startcondresp** command links a response to a condition, if they are not already linked.

To lock a response so it cannot be modified or removed, use the **chresponse** command with the **-L** flag.

Flags

-b

Specifies that the response, and all actions to be defined in this response, support event batching. For event batching, multiple events can be batched or grouped together and passed to a response. The actions of the response are directed to a file that contains the details for the batched events. A response that supports event batching can only be used for conditions that specify the events are to be batched.

An event response can be created for batched event conditions without an action script.

The **-b** flag cannot be specified with the **-e** flag.

-c existing_response[:node_name]

Copies an existing response. Links with conditions are not copied. The existing response is defined on the node known as *node_name* in a cluster. 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. If any other flags are specified, update the new response as indicated by the flags.

-d days_of_week

Specifies the days of the week when the action being defined can be run. *days_of_week* and *time_of_day* together define the interval when the action can be run.

Enter the numbers of the days separated by a plus sign (+) or as a range of days separated by a hyphen (-). More than one *days_of_week* parameter can be specified, but the parameters must be separated by a comma (.). The number of *days_of_week* parameters specified must match the number of *time_of_day* parameters specified. The default is all days. If no value is specified but a comma is entered, the default value is used. The values for each day follow:

- 1** Sunday
- 2** Monday
- 3** Tuesday
- 4** Wednesday
- 5** Thursday
- 6** Friday
- 7** Saturday

-e a | A | b | e | r

Specifies the type of event that causes the action being defined to run:

- a** Specifies an event. This is the default.
- A** Specifies any type of event (event, error event, or rearm event).
- b** Specifies an event and a rearm event.

e
Specifies an error event.

r
Specifies a rearm event.

More than one event type can be specified, for example: **-e ae**. The **-e** flag cannot be specified with the **-b** flag.

-E env_var=value[,env_var=value...]

Specifies any environment variables to be set before running the action. If multiple *env_var=value* variables are specified, they must be separated by commas.

-n action

Specifies the name of the action being defined. Only one action can be defined when the response is created. Use the `chresponse` command to add more actions to the response.

-o

Directs all standard output from *action_script* to the audit log. The default is not to keep standard output. Standard error is always directed to the audit log.

-p node_name

Specifies the name of the node where the response is defined. This is used in a cluster environment and the node name is the name by which the node is known in the domain. The default *node_name* is the local node on which the command runs. *node_name* is a node within the scope determined by the `CT_MANAGEMENT_SCOPE` environment variable.

If you are using `mkresponse` on the management server and you want the response to be defined on the management server, do *not* specify the `-p` flag.

-r return_code

Specifies the expected return code for *action_script*. If the expected return code is specified, the actual return code of *action_script* is compared to the expected return code. A message is written to the audit log indicating whether they match. If the `-x` flag is not specified, the actual return code is written to the audit log, and no comparison is performed.

-s action_script

Specifies the fully-qualified path for the script or command to run for the action being defined. See the `logevent`, `notifyevent`, and `wallevent` commands for descriptions of the predefined response scripts provided with the application.

-t time_of_day

Specifies the time range when *action* can be run, consisting of the start time followed by the end time, separated by a hyphen. *days_of_week* and *time_of_day* together define the interval when the action can be run.

The time is in 24-hour format (HHMM) where the first two digits represent the hour and the last two digits represent the minutes. The start time must be less than the end time because the time is specified by day of the week. More than one *time_of_day* parameter can be specified, but the parameters must be separated by a comma (.). The number of *days_of_week* parameters specified must match the number of *time_of_day* parameters specified. The default value is 0000-2400. If no value is specified but a comma is entered, the default value is used.

-u

Specifies that the action is to be run when a monitored resource becomes undefined.

-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

The *response* name is a character string that identifies the response. If the name contains spaces, it must be enclosed in quotation marks. A name cannot consist of all spaces, be null, or contain embedded double quotation marks.

Security

The user needs write permission for the IBM.EventResponse resource class to run `mkresponse`. Permissions are specified in the access control list (ACL) file on the contacted system. See the *Administering RSCD* 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 define a response with the name "Log event in audit log", run this command:

```
mkresponse "Log event in audit log"
```

2. To define a response with the name "E-mail root anytime" that has an action named "E-mail root", to be used any time Saturday and Sunday and uses the command `/opt/rsct/bin/notifyevent root` for both events and rearm events, run this command:

```
mkresponse -n "E-mail root" -d 1+7 \  
-s "/opt/rsct/bin/notifyevent root" -e b \  
"E-mail root anytime"
```

3. To define a response with the name "E-mail root anytime" that has an action named "E-mail root", to be used anytime Saturday and Sunday but only 8 am to 5 pm Monday through Friday and that uses the command `/opt/rsct/bin/notifyevent root` for events, run this command:

```
mkresponse -n "E-mail root" \  
-d 1+7,2-6 -t 0000-2400,0800-1700 \  
-s "/opt/rsct/bin/notifyevent root" -e a \  
"E-mail root anytime"
```

4. To define a response with the name "E-mail root anytime" that has an action named "E-mail root" to be used any time Saturday and Sunday, that uses the command `/opt/rsct/bin/notifyevent root` for both events and rearm events, and that sets the environment variable LANG to en_US, run this command:

```
mkresponse -n "E-mail root" -d 1+7 \  
-s "/opt/rsct/bin/notifyevent root" -e b \  
-E LANG="en_US" "E-mail root anytime"
```

5. To define a response with the name "E-mail root first shift" that has an action named "E-mail root" to be used Monday through Friday from 8 am to 6 pm, that uses the command `/opt/rsct/bin/notifyevent root` for rearm events, and that saves standard output in the audit log, expecting return code 5, run this command:

```
mkresponse -n "E-mail root" -d 2-6 -t 0800-1800 \  
-s "/opt/rsct/bin/notifyevent root" -e r -o \  
-r 5 "E-mail root first shift"
```

6. To define a response with the name "Critical notifications" as a copy of "Warning notifications", enter:

```
mkresponse -c "Warning notifications" "Critical notifications"
```

7. To define a batching-capable response called "Batched Event Response" without an action script, enter:

```
mkresponse -b "Batched Event Response"
```

These examples apply to management domains:

1. To define a response on the management server with the name "E-mail root anytime" that has an action named "E-mail root", to be used any time Saturday and Sunday and uses the command **/opt/rsct/bin/notifyevent root** for both events and rearm events, run this command on the management server:

```
mkresponse -n "E-mail root" -d 1+7 \  
-s "/opt/rsct/bin/notifyevent root" -e b \  
"E-mail root anytime"
```

2. To define a response on the managed node nodeB with the name "E-mail root anytime" that has an action named "E-mail root", to be used any time Saturday and Sunday and uses the command **/opt/rsct/bin/notifyevent root** for both events and rearm events, run this command on the management server:

```
mkresponse -n "E-mail root" -d 1+7 \  
-s "/opt/rsct/bin/notifyevent root" -e b \  
-p nodeB "E-mail root anytime"
```

3. To define a response on the managed node nodeB with the name "nodeB Warning notifications" as a copy of "nodeA Warning notifications" on the managed node nodeA, run this command on the management server:

```
mkresponse -c "nodeA Warning notifications":nodeA \  
-p nodeB "nodeB Warning notifications"
```

These examples apply to peer domains:

1. To define a response on the current node with the name "E-mail root anytime" that has an action named "E-mail root", to be used any time Saturday and Sunday and uses the command **/opt/rsct/bin/notifyevent root** for both events and rearm events, run this command from any node in the domain:

```
mkresponse -n "E-mail root" -d 1+7 \  
-s "/opt/rsct/bin/notifyevent root" -e b \  
"E-mail root anytime"
```

2. To define a response on the node nodeB in the domain with the name "E-mail root anytime" that has an action named "E-mail root", to be used any time Saturday and Sunday, that uses the command **/opt/rsct/bin/notifyevent root** for both events and rearm events, and that sets two environment variables (PAGE ALL and TIMER SET), run this command from any node in the domain:

```
mkresponse -n "E-mail root" -d 1+7 \  
-s "/opt/rsct/bin/notifyevent root" -e b \  
-p nodeB -E 'ENV1="PAGE ALL", ENV2="TIMER SET"' \  
"E-mail root anytime"
```

3. To define a response on the node nodeB in the domain with the name "nodeB Warning notifications" as a copy of "nodeA Warning notifications" on the node nodeA in the domain, run this command from any node in the domain:

```
mkresponse -c "nodeA Warning notifications":nodeA \  
-p nodeB "nodeB Warning notifications"
```

Location

/opt/rsct/bin/mkresponse

mkrole Command

Purpose

Creates new roles.

Syntax

mkrole [-R *load_module*] *Attribute=Value* [*Attribute=Value ...*] *Name*

Description

The **mkrole** command creates a new role. The *Name* parameter must be a unique role name. You cannot use the **ALL** or **default** keywords as the role name.

You can use the System Management Interface Tool (SMIT) **smit mkrole** fast path to run this command.

If the system is configured to use multiple domains for the role database, the new role is created in the first domain specified by the **secorder** attribute of the roles stanza in the **/etc/nscontrol.conf** file. Use the **-R** flag to create a role in a specific domain.

Every role must have a unique role ID that is used for security decisions. If the **id** attribute is not specified when a role is created, the **mkrole** command automatically assigns a unique ID to the role.

When the system is operating in enhanced (RBAC) mode, roles created in the role database can be immediately assigned to users but are not used for security considerations until the database is sent to the kernel security tables using the **setkst** command.

Flags

Item	Description
-R <i>load_module</i>	Specifies the loadable module to use for role creation.

Parameters

Item	Description
<i>Attribute=Value</i>	Initializes a role attribute. Refer to the chrole command for the valid attributes and values.
<i>Names</i>	Specifies a unique role name string.

Restrictions on Creating Role Names

The *Name* parameter that you specify must be unique, and can be a maximum of 63 single-byte printable characters. To prevent inconsistencies, restrict role names to characters with the POSIX portable filename character set. You cannot use the keywords **ALL** or **default** as a role name. Additionally, do not use any of the following characters within a role-name string:

- : (colon)
- " (quotation mark)
- # (pound sign)
- , (comma)
- = (equal sign)
- \ (backslash)
- / (forward slash)
- ? (question mark)
- ' (single quotation mark)
- ` (back quotation mark)

Restriction: The *Name* parameter cannot contain any space, tab, or newline characters.

Security

The **mkrole** command is a privileged command. You must assume a role that has the following authorization to run the command successfully.

Item	Description
aix.security.role.create	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 that are associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Files Accessed:

Mode	File
rw	/etc/security/roles
r	/etc/security/user.roles

Auditing Events:

Event	Information
ROLE_Create	role

Examples

1. To create the ManageRoles role and have the command automatically generate a role ID, use the following command:

```
mkrole authorizations=aix.security.role ManageRoles
```

2. To create the ManageRoles role in LDAP, use the following command:

```
mkrole -R LDAP authorizations=aix.security.role manageRoles
```

Files

Item	Description
<u>/etc/security/roles</u>	Contains the attributes of roles.
<u>/etc/security/user.roles</u>	Contains the role attribute of users.

mkcrpdomain Command

Purpose

Creates a peer domain definition.

Syntax

To create a peer domain definition, by:

- Specifying node names on the command line:

```
mkcrpdomain [-t TS_port] [-g GS_port] [-Q quorum_type | quorum_type_name] [-c] [-m fanout]  
[-S mode] [-k cssk_type [-i refresh_interval]] [-6] [-C cluster_type -R repository_disk [-D  
shared_disk1[,shared_disk2...]] [-h] [-TV] peer_domain node_name1 [node_name2 ...]
```

- Using a list of node names in an input file:

```
mkcrpdomain -f | -F {file_name | "-" } [-t TS_port] [-g GS_port] [-Q {quorum_type |
quorum_type_name}] [-c] [-m fanout] [-S mode] [-k cssk_type [-r refresh_interval]] [-6] [-C
cluster_type -R repository_disk [-D shared_disk1[,shared_disk2...]]] [-h] [-TV] peer_domain
```

To create a peer domain definition with the policy information:

```
mkcrpdomain [-p Policy] ApplDomain nameA [ @host_nameA ] [ nameB [ @host_nameB ] ... ]
```

Description

The `mkcrpdomain` command creates a peer domain definition with the name specified by the `peer_domain` parameter. The nodes that are specified by `node_name` are defined to the new peer domain. A peer domain can be used to provide high-availability services when you configure application and system resources.

The `preprnode` command must have been run on each of the nodes to be defined to the peer domain. The `preprnode` command prepares the security environment for the peer domain operations. See the `preprnode` command for more information about peer domain definition requirements. Only those nodes that have the appropriate security setup are successfully defined to the peer domain.

The **mkcrpdomain** command fails if one or more of these situations occurs:

- The name of the peer domain is already in use.
- One or more nodes cannot be successfully defined to the peer domain.
- The UDP port numbers for group services and topology services are not available on all of the nodes to be defined to the peer domain.

Use the **-c** flag to enable **mkcrpdomain** to continue when there is an error on one of the nodes. The peer domain quorum rules can be modified by using the **-Q** flag. The quorum rules determine under what conditions operational changes, such as starting or stopping resources, and configuration changes, such as adding or removing a node, can be made. Start up quorum defines how many nodes are contacted to get configuration information to start the peer domain. In a typical environment, two quorum rule types are used: normal and quick. For the quick quorum type, only one node is contacted before the peer domain group is started. Operational and configuration quorum rules are the same. To see what quorum rule types are available on a node, run:

```
lsrsrc -c IBM.PeerDomain AvailableQuorumTypes
```

You can use the **-k** flag to set the cluster shared secret key (CSSK). The CSSK is used for message authentication in the peer domain. By default, the CSSK is disabled (that is, set to `CSSKTYPE_None`). To enable message authentication, use a CSSK value such as `CSSKTYPE_DES_MD5` with the **-k** flag. Enabling message authentication affects performance. The complexity of the encryption algorithm determines the effect.

Message authentication also requires that the time-of-day clocks (TODs) of the nodes in the peer domain are synchronized — according to the system time — to within 2 minutes of each other. When the nodes' TODs are synchronized across the peer domain, this function helps to defend against message replay attacks. If the nodes' TODs are not synchronized to within 2 minutes of each other, messages that are passed between a sending node and a receiving node with a time difference that is longer than 2 minutes are discarded.

When message authentication is enabled by using the **-k** flag, a key refresh interval can be specified by using the **-r** flag. By default, the key is refreshed daily.

To change the CSSK type for a peer domain, use the `chrsrc` command. For example:

```
chrsrc -c IBM.RSCTParameters CSSKType=cssk_type
```

To list the CSSK type that is used for an online peer domain, use the `lsrsrc` command. For example:

```
lsrsrc -c IBM.RSCTParameters CSSKType
```


To cause the CSSK to be refreshed, use the `runact` command. For example:

```
runact -c IBM.PeerDomain UpdateKey
```

For information about setting up and managing CSSK settings, see the *Administering RSCT* guide.

Use the **-6** flag to establish a peer domain in which the IPv6 addresses that are configured on the nodes' network interfaces are visible as resources in **IBM.NetworkInterface** class. These IPv6 addresses are not used for heartbeating or internal peer domain operations. If the **-6** flag is not specified, no IPv6 addresses are visible as resources in **IBM.NetworkInterface**.

The `mkixpdomain` command does not bring the peer domain online automatically. To bring the peer domain online, run the `startixpdomain` command. You can add nodes to the peer domain by using the `addixpnode` command. To remove nodes from the peer domain, use the `rmixpnode` command.

A node can be defined in more than one peer domain but it can be online in only one peer domain at a time.

Flags

Item	Description
-6	<p>Specifies that the IPv6Support persistent class attribute of the IBM.NetworkInterface class has a value of 1 rather than the default (0) in the peer domain that is to be created. For any IP interface on any node in a cluster that has one or more IPv6 addresses configured, only one of these IPv6 addresses are made visible as a resource in IBM.NetworkInterface. Therefore, if a network interface has IPv4 addresses and IPv6 addresses configured on it, two resources in IBM.NetworkInterface refers to the interface (through the Name attribute), one with the IP address value set to the primary IPv4 address, and one with the selected IPv6 address. If multiple IPv6 addresses are configured on an interface, preference is given to global addresses over link-local addresses for representation as a resource. In addition, IPv6 addresses are used for heartbeating or internal peer domain operations.</p> <p>Note: Even if IPv6Support is changed, the current registered applications do not receive the notification for any resource addition or deletion until the domain or the IBM.ConfigRM class is restarted.</p>
-c	<p>Continues to run the <code>mkixpdomain</code> command on the remaining nodes.</p> <p>By default, if the <code>mkixpdomain</code> command fails on any node, it fails on all nodes. The -c flag overrides this behavior, so that the <code>mkixpdomain</code> command runs on the other nodes, even if it fails on one node.</p>
-C <i>cluster_type</i>	<p>Specifies the cluster type. Valid values are as follows:</p> <p>0 Creates a peer domain. This value is the default.</p> <p>1 Creates a peer domain and the underlying Cluster-Aware AIX (CAA) cluster.</p> <p>If you specify the -C 1 flag, you must also specify a repository disk by using the -R flag. Also, you can optionally specify one or more shared disks by using the -D flag.</p>
-D <i>shared_disk1</i> [, <i>shared_disk2</i> ...]	<p>Specifies one or more shared disks for a CAA cluster. If you specify the -D flag, you must also specify the -C and -R flags.</p>

Item

-f | -F {file_name | " -"

Description

Specifies that node names are read from a file or from standard input. Use **-f** *node_file* or **-F** *node_file* to read the node names from a file.

Note: The command requires that the following conditions be met to display a valid output:

- Specify 1 node name per line. The command ignores any blank characters to the left of the node name.
- Use a number sign (#) to indicate that the remainder of the line (or the entire line if the # is in column 1) is a comment.
- Specify the actual host name of the node by using @ sign without any space between node name and its host name. An example of the syntax follows:

```
[nodeA@hostA]
```

By default, all of the nodes that are listed in *node_file*:

- are group services group leader candidates
- are used for quorum decisions
- have access to the peer domain's tiebreaker mechanism

You can customize node characteristics by using an at sign (@) control character followed by one or more of these special characters:

P | p

Specifies that the node is a group services group leader candidate.

Q | q

Specifies that the node is a quorum node.

B | b

Specifies that the node has access to the peer domain's tiebreaker mechanism. B or b can be specified for quorum nodes only.

!

Specifies that the node does not have a certain characteristic. For example, **!Q** indicates that the node is not a quorum node.

When customizing node characteristics, consider the following (where x is P, Q, or B):

- Use only one @ control character per line, followed immediately by one or more special characters, after the node name and before any comments.
- Do not specify !QB for a node, as it results in an error.
- If you use a node number, add it after the node name and before any comments. The node number can precede or follow the node characteristic specifications.
- If x is specified for one or more nodes and !x is not specified for any nodes, the nodes that do not have an x specified are assumed to have a value of !x.
- If !x is specified for one or more nodes and x is not specified for any nodes, the nodes that do not have an !x specified are assumed to have a value of x.
- If x and !x are specified for different nodes in the same node file, all of the nodes in the file must have a specification of x or !x.

See the *Administering RSCT* for more information.

Use **-f** " - " or **-F** " - " to read the node names from standard input.

-g *GS_port*

Specifies the group services port number. This UDP port is for daemon-to-daemon communication. Any unused port in the range 1024 - 65535 can be assigned. The command fails if the specified port is unavailable. The default is 12348.

-h

Writes the command's usage statement to standard output.

Item`-k cssk_type`**Description**

Specifies the cluster shared secret key (CSSK) to be used for message authentication in the peer domain. Use the CSSK that best suits your applications in terms of the degree of data protection, overhead, and performance. The longer the key and message digest, the stronger the encryption algorithm. The stronger the algorithm, the slower the performance. The valid key types are as follows:

CSSKTYPE_None

Indicates that message authentication is disabled. This is the default value.

Note: If the **-S** flag is specified with mode value `nist_sp800_131a`, the default CSSK type is `CSSKTYPE_AES256_SHA256`.

CSSKTYPE_DES_MD5

Indicates that a Data Encryption Standard (DES) key with the message digest function MD5 is used to generate a 16-byte signature. This CSSK is recommended if a high degree of data protection is not required and if you want good performance with less data overhead.

CSSKTYPE_3DES_MD5

Indicates that a triple DES key with an MD5 digest is used to generate a 16-byte signature. Compared to `CSSKTYPE_DES_MD5`, this CSSK provides added data protection with slower performance, but with the same data overhead.

CSSKTYPE_AES256_MD5

Indicates that an Advanced Encryption Standard (AES) 256-bit key with an MD5 digest is used to generate a 24-bit signature. This CSSK provides more data protection than `CSSKTYPE_3DES_MD5`, but with slower performance and more data overhead.

The following CSSK types are compliant with the National Institute of Standards and Technology (NIST) Special Publications SP800-131a. You must be running RSCT 3.2.0.0, or later, to configure these key types.

CSSKTYPE_AES128_SHA256

Indicates that an Advanced Encryption Standard (AES) 128-bit key that has an SHA-1 (Secure Hash Algorithm) 256-bit digest is used to generate a 16-byte signature.

CSSKTYPE_AES128_SHA512

Indicates that an AES 128-bit key that has an SHA-1 512-bit digest is used to generate a 16-byte signature.

CSSKTYPE_AES256_SHA256

Indicates that an AES 256-bit key that has an SHA-2 256-bit digest is used to generate a 32-byte signature.

CSSKTYPE_AES256_SHA512

Indicates that an AES 256-bit key that has an SHA-2 512-bit digest is used to generate a 32-byte signature.

Notes:

- You must be running RSCT 2.4.7.1 or later to use this flag.
- If the **-S** flag is specified with the mode value `nist_sp800_131a`, the CSSK type must be either `CSSKType_None` or a key type that is compliant with the mode. If the created domain is compliant with the mode value `nist_sp800_131a`, and the **-k** flag is not specified, the domain is configured to use CSSK type `CSSK_AES256_SHA256`.

`-m fanout`

Specifies the maximum number of threads to use in parallel operations 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. If this flag is not specified, the default value (128) is used.

Item	Description
-p <i>Policy</i>	<p>Reads the policy from the user input when the <code>mkxpdomain</code> command creates the domain. You can use this command to specify the policy information when you create the domain. The valid values for the <i>Policy</i> attribute are 0 and 1.</p> <p>If you do not specify the <code>-p</code> flag for the <code>mkxpdomain</code> command, the default value 0 is set in non-CAA clusters and 1 is set in CAA clusters.</p> <p>If the value of policy is set as 1, the Name field of the <code>IBM.PeerNode</code> class is maintained in sync with the host name of the <code>IBM.PeerNode</code> class.</p> <p>If the value of policy is set as 0, the Name field is not maintained in sync with the host name, irrespective of the domain.</p> <p>However, the <code>-p 0</code> flag cannot be specified for CAA domain as a limitation. The policy information can be changed by using a <code>chrsrc</code> class action after the cluster is created.</p>
-Q <i>quorum_type</i> <i>quorum_type_name</i>	<p>Specifies the quorum rules that are used for startup, operational, and configuration quorum. Startup quorum defines how many nodes are contacted to obtain configuration information before the peer domain is started. Operational quorum defines how many nodes must be online to start and stop resources and how tie breaking is used. Configuration quorum defines how many nodes must be online to change the peer domain (adding or removing a node, for example). To see what quorum rule types are available on a node, run:</p>
	<pre>lsrsrc -c IBM.PeerDomain AvailableQuorumTypes</pre>
	<p>The valid values are as follows:</p>
	<p>0 normal</p> <p>Specifies normal quorum rules. This value is the default. For startup quorum, at least half of the nodes are contacted for configuration information. For configuration quorum, more than half of the nodes must be online to make configuration changes. For operational quorum, the cluster or subcluster must have a majority of the nodes in the peer domain. If a tie exists between subclusters, the subcluster that holds the tiebreaker has operational quorum.</p>
	<p>1 quick</p> <p>Specifies quick quorum rules. For startup quorum, even if no other nodes can be contacted, the node still comes online. For configuration quorum, more than half of the nodes must be online to make configuration changes. For operational quorum, the cluster or subcluster must have a majority of the nodes in the peer domain. If a tie exists between subclusters, the subcluster that holds the tiebreaker has operational quorum.</p>
-r <i>refresh_interval</i>	<p>Specifies the CSSK refresh interval when message authentication is enabled in the peer domain. This is the interval at which the CSSK is refreshed. The format of <i>refresh_interval</i> is: <i>dd:hh:mm:ss</i>, where <i>dd</i> is the number of days between key refreshes, <i>hh</i> is the number of hours, <i>mm</i> is the number of minutes, and <i>ss</i> is the number of seconds. The <i>refresh_interval</i> value can be truncated on the right, so <code>-r 5</code> means refresh every 5 days and <code>-r 0:12</code> means refresh every 12 hours.</p> <p>The default refresh interval is 1 day. The minimum refresh interval is 30 seconds. The maximum refresh interval is 30 days.</p> <p>The <code>-r</code> flag can be specified when the <code>-k</code> flag is used.</p> <p>You must be running RSCT 2.4.7.1 or later to use this flag.</p>
-R <i>repository_disk</i>	<p>Specifies the repository disk for a CAA cluster. If you specify the -R flag, you must also specify the -C flag.</p>
-S <i>mode</i>	<p>Enforces a security compliance mode for RSCT in the peer domain. The <i>mode</i> parameter can have the following values:</p> <p>none</p> <p>The domain does not enforce a security compliance mode.</p> <p>nist_sp800_131a</p> <p>RSCT is configured to be compliant with the National Institute of Standards and Technology (NIST) Special Publications SP800-131a. This mode value requires that all nodes that are specified in the <code>mkxpdomain</code> command must already be migrated to <code>nist_sp800_131a</code> mode or the nodes must be configured to use public or private keys that are compliant with this specification.</p> <p>Note: You must be running RSCT 3.2.0.0, or later, to use the -S flag.</p>

Item	Description
-t <i>TS_port</i>	Specifies the topology services port number. This UDP port is used for daemon-to-daemon communication. Any unused port in the range 1024 - 65535 can be assigned. The command fails if the specified port is unavailable. The default is 12347.
-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 new peer domain to be created. You can use these ASCII characters only in the peer domain name: A to Z, a to z, 0 to 9, . (period), and _ (underscore). In addition, the peer domain name *cannot* be IW.

node_name1 [node_name2 ...]

Specifies the node (or nodes) to include in this peer domain definition. The node name is the IP address or the long or short version of the DNS host name. The node name must resolve to an IP address.

Security

The user of the `mkxpdomain` command requires **write** permission to the `IBM.PeerDomain` resource class on each node that is to be defined to the peer domain. This permission is set up by running the `preprnode` command on each node that is to be defined to the domain, specifying the name of the node on which the user runs `mkxpdomain`.

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` has meaning only if `CT_CONTACT` is set to an IP address; it does not rely on the domain name system (DNS) service.

Restrictions

Any node to be defined to the peer domain must be reachable from the node on which this command runs.

Implementation Specifics

This command is part of the Reliable Scalable Cluster Technology (RSCT) file set 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 define a peer domain that is called `App1Domain` that consists of a node that is called `nodeA`, run this command on `nodeA`:

```
mkripdomain App1Domain nodeA
```

2. To define a peer domain that is called `App1Domain` that consists of three nodes that are called `nodeA`, `nodeB`, and `nodeC`, run this command on `nodeA`, `nodeB`, or `nodeC`:

```
mkripdomain App1Domain nodeA nodeB nodeC
```

3. To define a peer domain that is called `App1Domain` that consists of 2 nodes that are called `nodeA` and `nodeB`, with a topology services port number of 1200 and a group services port number of 2400, run this command on `nodeA` or `nodeB`:

```
mkripdomain -t 1200 -g 2400 App1Domain nodeA nodeB
```

4. To define a peer domain that is called `App1Domain` that consists of 2 nodes that are called `nodeA` and `nodeB` by using message authentication key algorithm `CSSKTYPE_DES_MD5`, run this command on `nodeA` or `nodeB`:

```
mkripdomain -k CSSKTYPE_DES_MD5 App1Domain nodeA nodeB
```

5. To define a peer domain that is called **`App1Domain`** that consists of the nodes **`nodeA`**, **`nodeB`**, **`nodeC`**, **`nodeD`**, and **`nodeE`**, by using the `/pd/pdnodes.config` file, run the following command on any of the nodes:

```
mkripdomain -f /pd/pdnodes.config App1Domain
```

where the contents of `/pd/pdnodes.config` are as follows:

```
# peer domain nodes for mkripdomain
nodeA      # dev node
nodeB      # dev node
nodeC      # prod node
nodeD      # test node
nodeE      # test node
```

- To define a peer domain that is called **AppDomain** that consists of **nodeA**, **nodeB**, **nodeC**, **nodeD**, and **nodeE**, by using the **/pd/pdnodes.config** file, which specifies that **nodeA** has access to the peer domain's tiebreaker mechanism, **nodeB** and **nodeC** cannot be used in quorum decisions, and **nodeC** and **nodeD** cannot be the group services group leader, run the following command on any of the nodes:

```
mkrpdomain -f /pd/pdnodes.config AppDomain
```

where the contents of **/pd/pdnodes.config** are as follows:

```
# peer domain nodes for mkrpdomain
nodeA   @QB   # dev node
nodeB   @!Q   # dev node
nodeC   @!Q!P # prod node
nodeD   @!P   # test node
nodeE   @Q    # test node
```

- To define a peer domain that is called **AppDomain**, which consists of 2 nodes that are called **nodeA** and **nodeB**, with the policy **NamePolicy 1**, run the following command:

```
mkrpdomain -p 1 AppDomain nodeA nodeB
```

NamePolicy 1 means that any change in host name also updates the node name. In this case, the host name is not specified in the beginning. Hence, the node names (**nodeA** and **nodeB**) are set as host names for the respective nodes.

- To define a peer domain that is called **AppDomain**, which consists of 2 nodes that are called **nodeA** and **nodeB**, whose host names are **hostA** and **hostB**, run the following command:

```
mkrpdomain AppDomain nodeA@hostA nodeB@hostB
```

These host names are the actual host names that are used for communication.

Location

/opt/rsct/bin/mkrpdomain

Files

The **/etc/services** file is modified.

mkrset Command

Purpose

Makes an rset containing the specified CPUs and memory regions and places it in the system registry.

Syntax

```
mkrset -c CPUlist [ -m MEMlist ] rsetname
```

Description

The **mkrset** command creates and places into the system registry an rset or exclusive rset (xrset) with the specified set of CPUs and/or memory regions. The rset name must not exist in the registry. The owner and group IDs of the rset will be set to the owner and group IDs of the command issuer. The rset will have read/write owner permissions and read permission for group and other. When used to create an xrset, the **mkrset** command changes the state of the corresponding CPUs on the system to exclusive mode. Creating an xrset requires root privilege.

Flags

Item	Description
-c	List of CPUs to be in the rset. This can be one or more CPUs or CPU ranges.
-m	List of memory regions to be in the rset. This can be one or more memory regions or ranges.

Parameters

Item	Description
rsetname	The name of the rset to be placed in the system registry. The name consists of a <i>namespace</i> and an <i>rsname</i> separated by a "/" (slash). Both the namespace and <i>rsname</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 and CAP_PROPAGATE capability.

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 make an **rset** containing CPUs 0-7 named **test/cpus0to7**, type:

```
mkrset -c 0-7 test/cpus0to7
```

2. To make an **rset** containing CPUs 1, 3, 5, 6, 7, 10 named **test/lotsofcpus**, type:

```
mkrset -c 1 3 5-7 10 test/lotsofcpus
```

Files

Item	Description
/usr/bin/mkrset	Contains the mkset command.

mkrsrc Command

Purpose

Defines a new resource.

Syntax

To define a new resource, using data that is...

- entered on the command line:

```
mkrsrc [-a | -N { node_file | "-" }] [-v] [-h] [-TV] resource_class attr=value...
```

- predefined in an input file:

```
mkrsrc -f resource_data_input_file [-v] [-a | -N { node_file | "-" }] [-h] [-TV] resource_class
```

To display the names and datatypes of the command arguments:

mkrsrc -l [-h] resource_class

To see examples of the `mkrsrc` command for a resource class:

```
mkrsrc -e [-h] [-TV] resource_class
```

Description

The `mkrsrc` command requests that the RMC subsystem define a new resource instance for the class specified by the `resource_class` parameter. At least one persistent attribute name and its value must be specified either as a parameter or by a resource definition file using the `-f` flag.

Before you run `mkrsrc`, you should run the `lsrsrcdef` command to determine which attributes are designated as `reqd_for_define` (required) or `option_for_define` (optional). Only attributes that are designated as `reqd_for_define` or `option_for_define` can be defined using the `mkrsrc` command. The `lsrsrcdef` command also identifies the datatype for each attribute. The value specified for each attribute must match this datatype.

To verify that all of the attribute names that are specified on the command line or in `resource_data_input_file` are defined as persistent attributes and are designated as `reqd_for_define` or `option_for_define`, use the `-v` flag. When the `mkrsrc` command is run with the `-v` flag, the resource is not defined. Instead, the resource attributes are merely verified to be persistent and designated as `reqd_for_define` or `option_for_define`. Once you have run `mkrsrc -v` to verify that all of the attributes that are specified on the command line or in `resource_data_input_file` are valid, you can issue the `mkrsrc` command without the `-v` flag to define the new resource.

If you are running in an RSCT peer domain or on the management server in an RSCT management domain and the resource class management type is subdivided, you can create the same resource on multiple nodes in one of two ways. The first way is to use the `-N node_file` flag to indicate that the node names to create the resources on are in a file. Use `-N "-"` to read the node names from standard input. The second way is to specify multiple node names in the **NodeNameList** resource attribute. The **NodeNameList** attribute defines where the resource is created when a cluster is present. If the **NodeNameList** attribute is not used, the resource is created on the local node. To find out if a resource class management type is subdivided, enter `lsrsrcdef -c resource_class | grep properties`.

Flags

-e

Displays examples of `mkrsrc` command-line input for:

1. required attributes only
2. required and optional attributes

-f resource_data_input_file

Specifies the name of the file that contains resource attribute information.

-l

Lists the command arguments and datatypes. Some resource managers accept additional arguments that are passed to the define 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 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 the resource class management type of the resource that is to be defined is subdivided and **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.

-v

Verifies that all of the attribute names specified on the command line or in the input file are defined as persistent attributes and are designated as `reqd_for_define` or `option_for_define`. The `mkrsrc` command does *not* define any resources when you use this flag.

-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 of the resource to be defined.

attr=value...

Specifies the attributes of the resource being defined. When defining a new resource instance, there are specific required attributes for each resource that must be defined. These attributes can be specified as parameters on the command line or defined in an input file by using the `-f` flag.

attr

The name of a persistent attribute for this resource. This attribute must be designated as `reqd_for_define` or `option_for_define`. Use the `lsrsrccdef` command to check the designation.

value

The value for this persistent attribute. The data type for this value must match the defined data type for the value of this attribute. Use the `lsrsrccdef` command to verify the data type for each attribute.

Security

The user needs write permission for the *resource_class* specified in `mkrsrc` to run `mkrsrc`. Permissions are specified in the access control list (ACL) file on the contacted system. See *Administering RSCT* 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 **rsct.rmc** fileset for the AIX® operating system.

Standard Output

- All command output is written to 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 (if there are any available) are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

1. To create a new resource in the IBM.Host class, assuming you already know which persistent attributes are required when defining a resource of this class, enter:

```
mkrsic IBM.Host Name=c175n05
```

2. To create a new resource in the `IBM.Processor` class by first generating a template to aid in the defining of these resources, enter:

```
lsrsrcdef -i IBM.Processor > /tmp/IBM.Processor.rdef
```

Then, edit the file `/tmp/IBM.Processor.rdef` and enter values for all of the attributes, substituting the type for an appropriate value, or leaving it blank for the default value.

Finally, enter:

```
mkrsrc -f /tmp/IBM.Processor.rdef IBM.Processor
```

3. To create two new `IBM.Host` resources using the information defined in file `/tmp/IBM.Host.rdef`, enter:

```
mkrsrc -f /tmp/IBM.Host.rdef IBM.Host
```

where the file `/tmp/IBM.Host.rdef` looks like this:

```
PersistentResourceAttributes::
resource 1:
  Name          = c175n04

resource 2:
  Name          = c175n05
```

4. This example creates a new resource in the `IBM.Foo` class. In this class, `Name` and `NodeList` are required attributes. The `Binary`, `SD`, `StringArray`, and `SDArray` attributes are optional. This example shows how to enter the more difficult data types from the command line. The data types for the optional attributes (`Binary`, `SD`, `StringArray`, and `SDArray`) are self-explanatory. Enter:

```
mkrsrc IBM.Foo Name=c175n05 \
NodeList={1} \
Binary="0xaabbccddeeff00" \
SD='[testing123,1,{2,4,6}]' \
StringArray='{ "testing 1 2 3",testing123,"testing 1 2 3"}' \
SDArray='{["testing 1 2 3",1,{1,3,5}], [testing,2,{2,4,6}]}'
```

5. To create resources for the `IBM.Example` class on multiple nodes in a peer domain, run this command:

```
mkrsrc -N /u/joe/common_node_file IBM.Example Name=Example_bar1 \
Binary="0xaabbccddeeff00"
```

where the contents of `/u/joe/common_node_file` look like this:

```
# common node file
#
node1.ibm.com      main node
node2.ibm.com      main node
node4.ibm.com      backup node
node6.ibm.com      backup node
#
```

6. To create resources of the `IBM.Example` class on multiple managed nodes in a management domain, run this command on the management server:

```
mkrsrc IBM.Example Name=Example_bar1 Binary="0xaabbccddeeff00" \
NodeNameList='{ "mgnode1.ibm.com", "mgnode2.ibm.com" }'
```

where the contents of `/u/joe/common_node_file` look like this:

```
# common node file
#
node1.ibm.com      main node
node2.ibm.com      main node
node4.ibm.com      backup node
node6.ibm.com      backup node
#
```

Note: As discussed in the `rmccli` general information file, attribute values for certain data types (structured data, array of structured data, and arrays containing strings enclosed in double quotation marks) should be enclosed in single quotation marks.

Location

`/opt/rsct/bin/mkrsrc`

mkrtc Command

Purpose

Configures or unconfigures Power SC real-time compliance for the operating system instance.

Syntax

To configure Power SC real-time compliance:

```
mkrtc -e email1, email2... [ -a alertStyle ] [ -d debug ] [ -i infoLevel ] [ -s emailSubject ] [ -c minCheckTime ]
```

To unconfigure Power SC real-time compliance:

```
mkrtc -u
```

Description

The **mkrtc** command is used to configure or unconfigure Power SC real-time compliance. To configure the **-e** flag, the email addresses must be provided as arguments. All other flags are optional. The **mkrtc** command saves the options to the `/etc/security/rtc/rtcd.conf` file, adds the Power SC real-time compliance entry to the `/etc/inittab`, and starts the **rtcd** daemon.

On unconfiguration, the **mkrtc** command removes the entry from the `/etc/inittab` file and stops the **rtcd** daemon.

Flags

Flag	Description
-a <i>alertStyle</i>	Specifies the alert style. The following are valid values: <ul style="list-style-type: none">• Once: Alerts once for the same set of compliance violations. This is the default alert style.• Event: Alerts once for the same set of compliance violations, but keeps alerting for each file modification event.• Always: Alerts compliance violations and the file modification. It keeps alerting for the file modification.
-c <i>minCheckTime</i>	Specifies the minimum amount of time between the compliance verifications. This flag checks the Power SC for compliance regularly even without file modification triggers, so that the mkrtc command can detect compliance implications in the files that are created by the user. For example, this flag can detect the <code>.xhost</code> file creation in the home directory that can have compliance implication. The default minimum time is 30 minutes. If this value is set to 0, it indicates that the compliance check is never run unless the files are modified.
-d <i>debug</i>	Specifies the debug option to be turned on or off. The valid values are <code>On</code> or <code>Off</code> . The default value is <code>Off</code> .

Flag	Description
-e <i>email1, email2...</i>	Provides a comma-separated list of emails to which email alerts are to be sent.
-i <i>infoLevel</i>	Specifies the information level of file modification events.
-s <i>emailSubject</i>	Provides the subject line to be used for the email alert.
-u	Unconfigures the Power SC real-time compliance.

On configuring Power SC real-time compliance, the **mkrtc** command performs the following tasks:

1. Updates the `/etc/security/rtc/rtcd.conf` file with the options from the command line.
2. Updates the `/etc/inittab` file with `psrtc:2:wait: /usr/bin/startsrc -s rtcd`.
3. Starts the **rtcd** daemon.

On unconfiguration, the **mkrtc** command performs the following tasks:

1. Removes the Power SC real-time compliance entry from `/etc/inittab` file.
2. Stops the **rtcd** daemon.

Security

Only the root user and users with **aix.security.aixpert** authorization are authorized to run this command.

Exit Status

Value	Description
0	The command runs successfully.
>0	An error occurred. The printed error message lists further details about the type of failure.

Examples

1. To configure Power SC real-time compliance, type the following command:

```
# mkrtc -e test@abc.com,dummy@abc.com -a event
```

This command configures Power SC real-time compliance to send compliance violation alert and file modification events to `test@abc.com` and `dummy@abc.com`. The alert style is set to `event`.

2. To unconfigure Power SC real-time compliance, type the following command:

```
# mkrtc -u
```

Files

Mode	File
rw	<code>/etc/security/rtc/rtcd.conf</code>

mkseckrb5 Command

Purpose

Migrates existing operating system users to Kerberos.

Syntax

```
mkseckrb5 [ -h | [ -r ] [user_name... ] ]
```

Description

This command gets the list of user names and creates Kerberos users. If the **-r** flag is not specified, the command prompts for a new password for each user.

Item	Description
Standard Output	Consists of information messages when the -h flag is used.
Standard Error	Consists of error messages when the command cannot complete successfully.

Flags

Item	Description
-h	Specifies that the command is only to display the valid command syntax.
-r	Specifies that random passwords are to be used.

Exit Status

Failure of this command to execute successfully results in incomplete migration. The admin must check the Kerberos database for the users that were migrated before taking further action.

Item	Description
0	Indicates the successful completion of the command.
1	Indicates that an error occurred.

Security

Only the root user is authorized to use this command.

Examples

1. To display the command syntax, type:

```
mkseckrb5 -h
```

2. To migrate existing users to Kerberos users, type:

```
mkseckrb5
```

3. To migrate user trojan to Kerberos user with random passwd, type:

```
mkseckrb5 -r trojan
```

Files

Item	Description
/usr/sbin/mkseckrb5	Contains the mkseckrb5 command.

mksecdap Command

Purpose

Sets up an AIX system as an LDAP server or client for security authentication and data management.

Syntax

The syntax to set up a server is:

```
mksecdap -s -a adminDN -p adminpasswd -S schematype [ -d baseDN ] [ -n port ] [ -k SSLkeypath ] [ -w SSLkeypasswd ] [ -x proxyDN -X proxypasswd ] [ -u NONE ] [ -v LDAPVersion ] [ -U ] [ -j <ssl|tls|ssltls|none|sslonly> ]
```

The syntax to set up a client is:

```
mksecdap -c -h serverlist -a bindDN -p bindpwd [ -d baseDN ] [ -n serverport ] [ -k SSLkeypath ] [ -w SSLkeypasswd ] [ -t cachetimeout ] [ -C cachesize ] [ -P NumberOfThreads ] [ -T heartBeatInt ] [ -M searchMode ] [ -D defaultEntry ] [ -A authType ] [ -i databaseModule ] [ -u userlist ] [ -U ] [ -j <ssl|tls> ]
```

Description

The **mksecdap** command can be used to set up IBM Directory servers and clients for security authentication and data management.

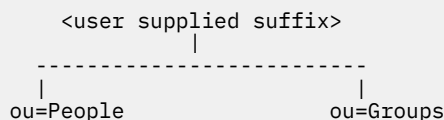
Notes:

1. The client (**-c** flag) and the server (**-s** flag) options cannot be used at the same time. When setting up a server, the **mksecdap** command might need to be run twice on that machine. Once to set up the server, and again to set up the system as a client.
2. The name and location of the LDAP server configuration file depends on the version of LDAP software installed. Refer to the LDAP software documentation of the installed release for more information.

Server Setup

Make sure that the LDAP server and the back-end IBM DB2® software are installed. You do not need to pre-configure IBM DB2 to run the **mksecdap** command for LDAP server setup. When you run the **mksecdap** command to set up the server, the command will:

1. Create a DB2 instance with **ldapdb2** as the default instance name.
2. If IBM Directory Server 6.0 or later is being configured then an LDAP server instance with the default name of **ldapdb2** is created. A prompt is displayed for the encryption seed to use to create the key stash files. The input encryption seed must be at least 12 characters.
3. Create a DB2 database with **ldapdb2** as the default database name. If a database already exists, **mksecdap** will bypass the above two steps. (This is the case when the LDAP server has been set up for other usage.) The **mksecdap** command will use the existing database to store the AIX user/group data.
4. Create the base DN (suffix) of the directory information tree (DIT). It is required that the base DN start with one of these attributes: **dc**, **o**, **ou**, **c**, **cn**. If no baseDN is supplied from the command line, the default suffix is set to **cn=aixdata** and the user/group data is placed under the **cn=aixdata** DN. Otherwise, the **mksecdap** command uses the user-supplied DN specified with the **-d** option. Users and groups will be exported to LDAP using the **sectoldif** command. The directory information tree (DIT) that will be created by default is shown below.



5. If **-u NONE** is not specified, then export the data from the security database files from the local host into the LDAP database. If **-u NONE** is specified, then **mksecldap** does not create the ou=People and ou=Group containers as it normally would, nor does it export users and groups. Depending on the **-S** option, the **mksecldap** command exports users/groups using one of the three LDAP schemas:
 - **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** object classes).
6. Set the LDAP server administrator DN and password.
7. Set the server to listen to a specified port if the **-n** option is used. The default port is 389. Also, TLS uses this port as default port (636 for SSL).
8. Updates the `/usr/lib/security/methods.cfg` file with the LDAP module configuration. If the **-i** option is entered from the command line, it also sets a LDAPA authentication-only module and a compound loadmodule (for example, LDAPAfiles when the **-i files** option is specified) with LDAPA serves for authentication and the databaseModule serves for identification.
9. Create the proxy entry if the **-x** and **-X** options are specified. Create an ACL for the base DN using the proxy entry. The default ACL can be found in `/etc/security/ldap/proxyuser.ldif.template`. The proxy entry can be used by client systems to bind to the server (see client setup section in this file).
10. Set the server to use SSL (secure socket layer) or TLS (transport layer security) if the **-k** option is specified for secure data transfer between this server and the clients. You must install the **GSKitv8** fileset and create an SSL or TLS key for this setup. You can install the **GSKitv8** fileset after you mount the AIX 7.3 expansion pack DVD.
11. Installs the `/usr/ccs/lib/libsecldapaudit.a` LDAP server plug-in. This plug-in supports AIX audit of the LDAP server.
12. Start/restart the LDAP server after all the above is done.
13. Add the LDAP server process (**slapd**) to `/etc/inittab` to have the LDAP server start after reboot.

Note: The **-U** option resets a previous setup for the server configuration file. It has no effect on the database. The first time the **mksecldap** command is run, it saves two copies of the server configuration file in the `/etc/security/ldap` directory. One is saved as the server configuration file name appended with **.save.orig** and the other is appended with **.save**. During each subsequent run of the **mksecldap** command, only the current server configuration is saved as a **.save** file. The undo option restores the server configuration file with the **.save** copy. In AIX 5.3 it is possible to invoke **mksecldap -s** in succession to create and populate multiple suffixes. If this has been performed then the `.save.orig` file will need to be manually restored in order to revert to the initial configuration file.

Client Setup

Make sure that the LDAP client fileset is installed and the LDAP server has been setup and is running. The **mksecldap** command performs the following steps during client setup:

1. Saves the LDAP server(s)' host name.
2. Saves the user base DN and group base DN of the server. If no **-d** option is supplied from command line, the **mksecldap** command searches the LDAP server for **aixaccount**, **aixaccessgroup**, **posixaccount**, **posixgroup**, and **aixauxaccount** objectclasses, and sets up the base DN's accordingly. If the server has multiple user or group bases, you must supply the **-d** option with a Relative Distinguished Name (RDN) so that the **mksecldap** command can setup the base DN's to the ones within that RDN.

If the **posixaccount** objectclass is found during client setup, **mksecldap** will also try to search for base DN's for the following entities from the server and save any that are found:

- hosts
- networks
- services

- netgroups
 - protocols
 - rpc
 - authorizations
 - roles
 - privcmds
 - privdevs
 - privfiles
 - usrkeystore
 - grpkeystore
 - efscookies
 - admkeystore
 - domains
 - domobjjs
3. Determines the schema type used by the LDAP server - **AIX** specific schema, **RFC 2307** schema, **RFC 2307** schema with full AIX support, or Microsoft Services for UNIX 3.0 schema. It sets the object classes and attribute maps in the `/etc/security/ldap/ldap.cfg` file accordingly. The **mksecldap** command does not recognize other schema types, so clients must be setup manually.
 4. Sets SSL or TLS for secure data transfer between this host and the LDAP server. This step requires that the client SSL or TLS key and the key password are created in advance, and the server must be setup to use SSL or TLS for the client SSL or TLS to work. SSL or TLS functionality requires the installation of the **GSKitv8** fileset. You can install the **GSKitv8** fileset after you mount the AIX 7.3 Expansion pack DVD.
 5. Encrypts the bind password.
 6. Saves the LDAP server bind DN and password. The DN/password pair must exist on the LDAP server. If the bind DN and password are not given, **mksecldap** uses anonymous bind. Some of the data might not be returned from the LDAP server with anonymous bind. Consult your LDAP administrator before you choose anonymous bind.
 7. Sets the optionally specified configuration values as defined in the client setup flags section.
 8. Optionally sets the list of users or all users to use LDAP by modifying their SYSTEM line in the `/etc/security/user` file. For more information on enabling LDAP login, see the following note.
 9. Starts the client daemon process (**secldapclntd**).
 10. Adds the client side daemon process to `/etc/inittab` to have this daemon start after a reboot.

Note: All client configuration data is saved to the `/etc/security/ldap/ldap.cfg` configuration file. The **-U** option resets a previous setup to the `/etc/security/ldap/ldap.cfg` file by replacing the file with the configuration stored in `/etc/security/ldap/ldap.cfg.save`. Setting the SYSTEM to LDAP for the default stanza of `/etc/security/user` only allows LDAP users to login to the system. Setting the SYSTEM to LDAP or compat allows both LDAP users and local users to login to the system.

Flags

For Server Setup

Item	Description
-a <i>AdminDN</i>	Specifies the LDAP server administrator DN.
-d <i>baseDN</i>	Specifies the suffix or base DN of the AIX subtree. The default is cn=aixdata .

Item	Description
-j <ssl tls ssltls none sslonly>]	Specifies the encryption connection type that is used during the communication with the LDAP clients. Valid values are SSL, TLS, SSLTLS, and SSLONLY. If the -k and -w flags are specified without the -j flag, the default connection type is SSL.
-k <i>SSLkeypath</i>	Specifies the full path to the SSL or TLS key database of the server.
-n <i>port</i>	Specifies the port number that the LDAP server listens to. Default is 389 for non-SSL and 636 for SSL.
-p <i>adminpasswd</i>	Specifies the clear text password for the administrator DN.
-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.
-s	Indicates that the command is being run to setup the server.
-w <i>SSLkeypasswd</i>	Specifies the password for the SSL or TLS key database.
-U	Specifies to undo the previous server setup to the LDAP configuration file. The database is not affected.
-u NONE	Specifies not to migrate users and groups from local system. The only valid value is NONE. Any other values are ignored. When this option is used, <code>mksecldap</code> does not create the <code>ou=People</code> and <code>ou=Group</code> containers as it normally would, nor does it export users and groups. No -S option is required with this option.
-v <i>LDAPVersion</i>	Denotes a specific version of the LDAP server fileset to configure. The value must be in the format <code>##</code> where <code>#</code> is a number. For example, <code>6.0</code> . If not specified, the mksecldap command configures the most recent version of the LDAP server fileset that is installed.
-X <i>proxypasswd</i>	Specifies the password for the proxy DN.
-x <i>proxyDN</i>	Specifies the DN of the proxy entry. This entry can be used by client systems to bind to this server.

For Client Setup

Item	Description
-a <i>bindDN</i>	Specifies the DN to bind to the LDAP server. The DN must exist on the LDAP server. If <code>authtype</code> is <code>unix_auth</code> , <code>bindDN</code> must have read access to the <code>userPassword</code> field on the LDAP server. Without the <code>-a</code> option, mksecldap configures anonymous bind. Note: Some of the data might not be retrieved from the LDAP server with anonymous bind. Consult your LDAP server administrator about using anonymous bind.

Item	Description
-A <i>authType</i>	<p>Specifies the authentication mechanism used to authenticate users. Valid values are unix_auth and ldap_auth. The default is unix_auth. The values are defined as follows:</p> <ul style="list-style-type: none"> • unix_auth - Retrieve user password from LDAP and perform authentication locally. • ldap_auth - Bind to LDAP server, sending password in clear text, for authentication. <p>Note: When using ldap_auth type authentication, the use of SSL or TLS is strongly recommended since during authentication passwords will be sent in clear text to the LDAP server.</p>
-i <i>databaseModule</i>	<p>Specifies the configuration of LDAP as the authentication-only module (LDAPA) of a compound loadmodule. The <i>databaseModule</i> option specifies the database module of the compound loadmodule.</p>
-j <i><ssl tls></i>	<p>Specifies the encryption connection type that is used during the communication with the LDAP server. Valid values are SSL and TLS. If the -k and -w flags are specified without the -j flag, the default connection type is SSL.</p>
-c	<p>Indicates the command is being run to setup the client.</p>
-C <i>Cachsize</i>	<p>Specifies the maximum number of user entries that can be used in the client-side daemon cache. Valid value is in the range 100-65536 for user cache. The default is 1000. The valid range for the group cache is 10-65536. The default value is 100. If you set the user cache entry in the start-secldapcmd command, by using the -C option, the group cache is set to 10% of the user cache.</p>
-D <i>defaultEntryLocation</i>	<p>Specifies the location of the default entry. Valid values are ldap and local. The default is ldap. The values are defined as follows:</p> <ul style="list-style-type: none"> • ldap - Use the default entry in LDAP for all attribute default values. • local - Use the default stanza from local /etc/security/user file for all attribute default values.
-d <i>baseDN</i>	<p>Specifies the base DN for the mksecldap command to search for the user base DN and group base DN. If not specified from the command line, the entire database is searched.</p>
-h <i>serverlist</i>	<p>Specifies a comma separated list of hostnames (server and backup servers).</p>
-k <i>SSLkeypath</i>	<p>Specifies the full path to the client SSL or TLS key database.</p>

Item	Description
-M <i>searchMode</i>	<p>Specifies the set of user and group attributes to be retrieved. Valid values are ALL and OS. The default is ALL. The values are defined as follows:</p> <ul style="list-style-type: none"> • ALL - Retrieve all attributes of an entry. • OS - Retrieve only the operating system required attributes of an entry. Non-OS attributes like telephone number, binary images etc. will not be returned. <p>Note: Use OS only when entries have many non-OS required attributes or attributes with large value, e.g. binary data, to reduce sorting effort by the LDAP server.</p>
-n <i>serverport</i>	Specifies the port number that the LDAP server is listening to.
-p <i>bindpasswd</i>	Specifies the clear text password for the bindDN used to bind to the LDAP server.
-P <i>NumberOfThreads</i>	Specifies the number of threads that the client side daemon uses. Valid values are 1-256. The default value is 10.
-t <i>cachetimeout</i>	<p>Specifies the maximum time length that a cache entry expires. Valid values are 60-3600 seconds. The default is 300 seconds. Set this value to 0 to disable caching.</p> <p>Note: > The <i>cachetimeout</i> attribute is used to set up values for the <i>usercachetimeout</i> and <i>groupcachetimeout</i> attributes. Starting with AIX 7.3, the <i>cachetimeout</i> attribute has been deprecated. You can use the <i>usercachetimeout</i> and <i>groupcachetimeout</i> attributes instead. <</p>
-T <i>heartBeatInt</i>	Specifies the time interval of heartbeat between this client and the LDAP server. Valid values are 60-3600 seconds. Default is 300.
-u <i>userlist</i>	<p>Specifies the comma separated list of user names to enable for LDAP authentication. These users will have their registry and SYSTEM attributes set to use LDAP. Specify ALL to enable all users on the client.</p> <p>Note: Alternatively, the SYSTEM attribute in the default stanza of <code>/etc/security/user</code> can be set to LDAP, allowing only LDAP users to log in. Setting the SYSTEM attribute to LDAP or compat allows both LDAP users and local users to log in to the system.</p>
-w <i>SSLkeyfilepath</i>	Specifies the password for the client SSL or TLS key database.
-U	Specifies to undo the previous client setup to the LDAP client configuration file.

Security

A user with the **aix.security.ldap** authorization is authorized to use this command.

Examples

1. To setup an LDAP server of RFC2307AIX specific schema for users and groups, enter:

```
mksecldap -s -a cn=admin -p adminpwd -S rfc2307aix
```

This sets up an LDAP server with LDAP server administrator DN being **cn=admin**, password being **adminpwd**. User and group data is exported from local files to the default **cn=aixdata** suffix using RFC2307AIX schema.

2. To setup an LDAP server with a baseDN other than the default and with SSL secure communication , enter:

```
mksecldap -s -a cn=admin -p adminpwd -d o=mycompany,c=us -S rfc2307 -k /usr/ldap/
serverkey.kdb
-w keypwd
```

This sets up an LDAP server with LDAP server administrator DN being **cn=admin**, password being **adminpwd**. User and group data is exported from local files to the **o=mycompany,c=us** suffix using RFC2307 schema. The LDAP server uses SSL communications by using the key stored at `/usr/ldap/serverkey.kdb`. The password to the key, **keypwd**, must also be supplied.

3. To setup an LDAP server of RFC2307AIX schema type and create a proxy account, enter:

```
mksecldap -s -a cn=admin -p adminpwd -d c=us -S rfc2307aix -x cn=proxy,c=us -X proxypwd
```

This sets up an LDAP server with LDAP server administrator DN being **cn=admin**, password being **adminpwd**. User and group data is exported from local files to the **c=us** suffix using RFC2307AIX schema. A proxy identity is setup with DN being **cn=proxy,c=us** and password **proxypwd**. The ACL specified in `/etc/security/ldap/proxy.ldif.template` will also have been applied on the server for the **cn=proxy,c=us** DN.

4. To undo a previous server setup:

```
mksecldap -s -U
```

This undoes the previous setup to the server configuration file. Note, for safety reasons, this does not remove any database entries or database created by a previous setup. One has to remove the database entries/database manually if they are not needed anymore.

5. To setup a client to use the **server1.ibm.com** and **server2.ibm.com** LDAP servers, enter:

```
mksecldap -c -a cn=admin -p adminpwd -h server1.ibm.com,server2.ibm.com
```

The LDAP server administrator DN and password is supplied for this client to authenticate to the server. The **mksecldap** command contacts the LDAP server for schema type used, and sets up the client accordingly. Without the **-d** option from the command line, the entire server DIT is searched for the user base DN and the group base DN.

6. To setup the client to talk to the **server3.ibm.com** LDAP server using SSL, enter:

```
mksecldap -c -a cn=admin -p adminpwd -h server3.ibm.com -d o=mycompany,c=us
-k /usr/ldap/clientkey.kdb -w keypwd -u user1,user2
```

This sets up an LDAP client similar to case 3, but with SSL communication. The **mksecldap** command searches the **o=mycompany,c=us** RDN for user base DN and group base DN. Account user1 and user2 are configured to authenticate through LDAP.

Note: The **-u ALL** option enables all LDAP users to login to this client.

7. To setup a client to talk to **server4.ibm.com** and use **ldap_auth** authentication with a proxy bind, enter:

```
mksecldap -c -a cn=proxy,c=us -p proxypwd -h server4.ibm.com -A ldap_auth
```

This sets up an LDAP client to bind to the LDAP server with the **cn=proxy,c=us** DN. Because the administrator DN is not used, the access granted to the client is dependent on the ACL setup on the LDAP server for the **cn=proxy,c=us** DN. The client is also setup to use **ldap_auth**-type authentication which sends passwords in clear text to the LDAP server for comparison.

Note: When using **ldap_auth**-type authentication, the use of SSL or TLS is strongly recommended because during authentication passwords will be sent in clear text to the LDAP server.

8. To undo a previous client setup, enter:

```
mksecldap -c -U
```

This undoes the previous setup to the `/etc/security/ldap/ldap.cfg` file. This does not remove the **SYSTEM=LDAP** and **registry=LDAP** entries from the `/etc/security/user` file.

9. To setup a client using LDAP as authentication-only module, and using files for user identification, enter:

```
mksecldap -c -a cn=admin -p adminpwd -h server1.ibm.com -i files -A ldap_auth
```

This sets up an LDAPfiles compound loadmodule, where the module LDAPA is used for user authentication and files is used for user identification. Authentication is set to `ldap_auth`.

Files Accessed

Item	Description
Mode	File
r	<code>/etc/passwd</code>
r	<code>/etc/group</code>
r	<code>/etc/security/passwd</code>
r	<code>/etc/security/limits</code>
r	<code>/etc/security/user</code> (on the server)
rw	<code>/etc/security/user</code> (on the clients)
r	<code>/etc/security/environ</code>
r	<code>/etc/security/user.roles</code>
r	<code>/etc/security/lastlog</code>
r	<code>/etc/security/smitacl.user</code>
r	<code>/etc/security/mac_user</code>
r	<code>/etc/security/group</code>
r	<code>/etc/security/smitacl.group</code>
r	<code>/etc/security/roles</code>
rw	<code>/etc/security/login.cfg</code> (on the server)
rw	<code>/etc/slapd32.conf</code> (on the server)
rw	<code>/etc/security/ldap/ldap.cfg</code> (on the client)

mksecpki Command

Purpose

mksecpki configures AIX PKI server components. The components of AIX PKI are Certificate Authority, Registration Authority, and Audit subsystems.

Syntax

```
mksecpki {-u username -f reference_file [-p CA_port] [-H ldap_host] [-D dn -w password] [-i certificate_issuer_dn] | -U username}
```

Description

The **mksecpki** command configures AIX PKI server components. **mksecpki** must be run after configuring an LDAP server to publish certificates. The values for the options **-H**, **-D**, **-w**, and **-i** must be the same values as the ones specified during the LDAP configuration. Otherwise, the CA will not be able to publish certificates to LDAP.

The **-u** option specifies the AIX username which will host AIX PKI. The username must follow AIX username rules. Do not use **-u** and **-U** together. The invoker of the command will be asked to provide a password for the username. **mksecpki** will create a database instance with the same name.

The **-f** option specifies the file containing the reference number and passphrase. The client certificate requests will use these exact same values while communication with the CA. The reference number and passphrase are each specified on a separate line. The following is the contents of an example **iafile**:

```
11122233
tempwd1234
```

The **-p** option specifies the port that Certificate Authority accepts the certificate requests. If no port number is given, 1077 will be assumed.

The **-H** option specifies the hostname of the LDAP server where the certificates are published to. Prior to invoking the **mksecpki** command, an LDAP server must be setup to publish certificates. Otherwise, the certificates will not be published to LDAP, however, certificate will be returned to the requestor when certificate management commands are used. If the **-H** option is not given the localhost will be used as the hostname.

The **-D** option is used to specify the directory administrators distinguished name. This must be the same one that is specified during the configuration of the LDAP server.

The **-w** option specifies the password corresponding to the administrator DN. It is an error not to specify both the admin DN and password.

The **-i** option specifies the distinguish name of the Certificate Authority issuing the certificates. This must be the same value as the one given when setting an LDAP server for publishing certificates.

The **-U** option specifies the username that hosts the AIX PKI that will be unconfigured. The command will confirm the unconfiguration before starting its operation. This option removes the username from the system. The invokers of this command will be asked if they want to remove the home directory of the username. When this command runs without errors, it displays a message indicating the successful completion. The invoker of this command is recommended to wait for this message.

Flags

Item	Description
-u <i>username</i>	Specifies the name of the username that is going to be created that will host AIX PKI server components.
-f <i>reference_file</i>	Specifies the file which contains the reference number and passphrase that is used when making a certificate creation request.
-p <i>CA_port</i>	Specifies the Certificate Authority Communication Port.
-H <i>ldap_host</i>	Specifies the LDAP host where the certificates are going to be published.
-D <i>adminDN</i>	Specifies directory administrator distinguished name (DN). Note: The -D option requires that the -w password option also be specified.
-w <i>password</i>	Specifies directory administrator password.
-i <i>certificate_issuer_dn</i>	Specifies the distinguished name of the Certificate Authority issuing certificates.

Item	Description
-U <i>username</i>	Specifies the username which hosts the AIX PKI that will be unconfigured.

Security

This command should grant execute (x) access only to the root user and members of the security group.

Examples

To configure AIX PKI server side using **pkitest.ibm.com** as the LDAP host name for publish certificates and using **o=aix,c=us** as the issuer name, enter:

```
$ mksecpki -u pkiuser -f iafile -p 829 -H pkitest.ibm.com -D cn=admin
-w password -i o=aix,c=us
```

where **iafile** contains the reference number and passphrase.

To unconfigure the server, enter:

```
$ mksecpki -U pkiuser
```

Files

/usr/lib/security/pki/ca.cfg

mksensor Command

Purpose

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

Syntax

To define a sensor:

```
mksensor [ -n host1[, host2...] | -N { node_file | "-" } ] [ -i seconds ] [ -c n ] [ -e 0 | 1 | 2 ] [ -u user-ID ]
[-h] [ -v | -V ] sensor_name ["sensor_command["]
```

To define a microsensor:

```
mksensor -m [ -n host1[, host2...] | -N { node_file | "-" } ] [ -i seconds ] [-h] [ -v | -V ]
microsensor_name microsensor_module [ ["microsensor_arguments["]
```

Description

The `mksensor` command defines a sensor resource to the resource monitoring and control (RMC) subsystem. A *sensor* is an RMC resource with attributes that you can monitor. You can use the event-response resource manager (ERRM) commands to set up monitoring of the sensor attributes. The response actions defined will run when a monitored sensor event occurs. This enables administrators to extend RMC monitoring capabilities without having to write a resource manager.

For sensors, the *sensor_command* parameter specifies the command or script that the sensor resource manager will run to set (and then later, update) the sensor attribute values. After the sensor attributes have been monitored, the sensor resource manager sets the attribute values. Then, at defined intervals, the sensor resource manager updates these values.

For microsensors, the *microsensor_module* parameter specifies the path name to the loadable module that the microsensor resource manager will call to set (and then later, update) the microsensor attribute values. After the microsensor attributes have been monitored, the microsensor resource manager sets

the attribute values. Then, at defined intervals, the microsensor resource manager updates these values. Use the **-m** flag to create a microsensor.

Alternatively, you can use **chsensor** or **refsensor** to update the sensor or microsensor attribute values. The **lssensor** command displays values for sensor or microsensor attributes that you can set using a sensor command or a microsensor module, if the attributes are monitored. If the attributes are not monitored, **lssensor** does not display their values. To remove a sensor or a microsensor, use the **rmsensor** command.

The **mksensor** command runs on any node. To define a sensor or a microsensor on one or more nodes in a management domain or a peer 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*.

A sensor consists of the following attributes, which can be set using a sensor command: :

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.

A sensor command sets attribute values by sending the values to standard output in a format that the sensor resource manager can parse. The format is *attr=value*. For example, if the sensor command sets the **Int32** attribute to 57, it writes **Int32=57** to standard output. To set more than one attribute value, the sensor command can write multiple *attr=value* pairs to standard output. The *attr=value* pairs can be on one or more lines. If the sensor command output is not in *attr=value* form, it is assumed to be a string and the value is placed in the **String** attribute.

The sensor command runs using the user ID that creates the sensor resource. Once a sensor resource is monitored, the sensor command is run at intervals specified by the **-i** flag, which is expressed in seconds. The default interval is 60 seconds. Specify a value of 0 to indicate that the sensor command is not to run at intervals. In this case, the **refsensor** command is typically used to update the sensor values.

Use the **-e** flag to control how the exit values from *sensor_command* are interpreted. Depending on this setting, when the exit value of the *sensor_command* is considered to be an error, the sensor attributes are not set and information is written to the audit log.

A microsensor consists of the following attributes, which can be set using a microsensor load module:

Float32

The type float32 attribute for this microsensor resource

Float32Array

The type float32 array attribute for this microsensor resource

Float64

The type `float64` attribute for this microsensor resource

Float64Array

The type `float64` array attribute for this microsensor resource

Int32

The type `int32` attribute for this microsensor resource

Int32Array

The type `int32` array attribute for this microsensor resource

Int64

The type `int64` attribute for this microsensor resource

Int64Array

The type `int64` array attribute for this microsensor resource

Quantum

The type `quantum` attribute for this microsensor resource.

String

The type `string` attribute for this microsensor resource.

StringArray

The type `string` array attribute for this microsensor resource.

UInt32

The type `uint32` attribute for this microsensor resource.

UInt32Array

The type `uint32` array attribute for this microsensor resource.

UInt64

The type `uint64` attribute for this microsensor resource.

UInt64Array

The type `uint64` array attribute for this microsensor resource.

The microsensor resource manager will make calls to the microsensor load module to set the values of the microsensor attributes. See the *Administering RSCT* for information about how to use microsensors.

Flags**-m**

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

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

Specifies one or more nodes on which the sensor should be defined. By default, the sensor is defined 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 to the left of a node name are ignored
- Any characters to the right of a node name are ignored

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

-i *seconds*

Specifies the interval at which a sensor command is run to update a sensor's attribute values or a microsensor module is run to update a microsensor's attribute values. *seconds*, which is an integer value, must be greater than or equal to 10. The default interval is 60 seconds.

The sensor command is run at the specified interval only when the sensor resource is monitored. The microsensor module is run at the specified interval only when the microsensor resource is monitored. If the interval is set to 0, the sensor command or microsensor module will not run automatically.

a

Using this flag is independent of using the `refreshsensor` command to refresh a sensor.

-c n

Specifies whether special handling is required for this sensor. *n* can be one of these values:

0

Indicates that no special handling is required. This is the default.

The sensor command will run at the interval that is defined for *sensor_name*. The sensor command will *not* run when monitoring begins or when the `lssensor` command is run.

1

Indicates that the sensor command will run when monitoring begins. The sensor command will also run at the interval that is defined for *sensor_name*. The sensor command will *not* run when the `lssensor` command is run.

Specifying this value is not recommended, unless you expect the sensor command to run quickly. If the sensor command does not run quickly, it could block other requests to the sensor resource manager. These requests will not be processed until the sensor command finishes running.

2

Indicates that output from the command in the `SavedData` field is not saved permanently to `SavedData` persistent resource attributes. If this value is not specified, the sensor resource manager updates data in the registry's resource table whenever the command's standard output contains the line: `SavedData="any-string"`.

3

Indicates a combination of values 1 and 2.

4

Indicates that the sensor resource manager will run the sensor command after monitoring has stopped.

5

Indicates a combination of values 1 and 4.

6

Indicates a combination of values 2 and 4.

7

Indicates a combination of values 1, 2, and 4.

-e 0 | 1 | 2

Specifies how the sensor resource manager interprets the exit values of *sensor_command*, as follows:

0

No exit value from *sensor_command* is an error.

1

An exit value other than 0 from *sensor_command* is an error.

2

An exit value of 0 from *sensor_command* is an error.

The default value is 1. The sensor attributes are not updated when the exit value is interpreted as an error. For an error, information is written to the audit log.

-u user-ID

Specifies the name of a user whose privileges will be used to run the sensor command. The user should already be defined on the system. The default value for *user-ID* is the user name that is associated with the current effective user ID.

-h

Writes the command's usage statement to standard output.

-v | -V

Writes the command's verbose messages to standard output.

Parameters

[*microsensor_argument*]

Specifies a string that will be passed to the microsensor module callback function. The microsensor resource manager will break the string into an array of strings based on blank characters in the microsensor argument. The microsensor argument cannot be changed once the microsensor is defined.

If the microsensor argument contains any blank characters or any special characters that can be interpreted by the shell, it must be enclosed in double quotation marks. When the microsensor argument is enclosed in double quotation marks, you must include a backslash escape character (\) before an "inner" double quotation mark. You must also include a \ before a dollar sign (\$).

microsensor_module

Specifies the path name to the loadable microsensor module. A signature for the module is stored by the microsensor resource manager and is verified when the module is used. The microsensor module cannot be changed once the microsensor is defined.

microsensor_name

Specifies the name of the microsensor that is to be defined.

[*sensor_command*]

Specifies a command or script that the sensor resource manager will use to set the attribute values of the sensor. You should not call any of the sensor resource manager commands (`chsensor`, `lssensor`, `mksensor`, `refsensor`, or `rmsensor`) as part of this parameter.

If *sensor_command* contains any blank characters, or any special characters that can be interpreted by the shell, it must be enclosed in double quotation marks.

When *sensor_command* is enclosed in double quotation marks, you must include a backslash escape character (\) before an "inner" double quotation mark. You must also include a \ before a dollar sign (\$). See Example 2 for more information.

sensor_name

Specifies the name of the sensor that is to be defined.

Security

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

To create 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 *Administering RSCT* 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.

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.

Restrictions

You should not call any of the sensor resource manager commands (`chsensor`, `lssensor`, `mksensor`, `refsensor`, or `rmsensor`) as part of the *sensor_command* parameter, as this could cause a deadlock.

Implementation Specifics

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

Examples

1. To create a new sensor called `Sensor1` that runs the script `/usr/bin/updateSensor1`, which will update the sensor attributes every 30 seconds (once monitored), enter:

```
mksensor -i 30 Sensor1 "/usr/bin/updateSensor1"
```

The contents of `/usr/bin/updateSensor1` may be like:

```
#!/usr/bin/perl
my $int32 = some_fn_that_generates_i32_value;
my $string = some_fn_that_generates_string_value;
print "Int32=$int32 String=$string";
exit 0;
```

A sample condition could be:

```
mkcondition -r IBM.Sensor -s "Name==Sensor1" -e "Int32 > 100" Sensor1Int32
```

Using the response "E-mail root anytime", a start monitoring command may be:

```
startcondresp Sensor1Int32 "E-mail root anytime"
```

2. To create a sensor called `Sensor1` with a `sensor_command` value of

```
df -m /var | sed '1d' | sed 's/%%/g' | /bin/awk '{ print "Int32=\"$4}'
```

enter:

```
mksensor Sensor1 "df -m /var | sed '1d' | sed 's/%%/g' | /bin/awk \  
'{ print \"Int32=\\\"$4}'"
```

When `sensor_command` is enclosed in double quotation marks, you must include a backslash escape character (`\`) before an "inner" double quotation mark. You must also include a `\` before a dollar sign (`$`). So in this example, the sensor command substring `"Int32=\"$4` becomes `\\"Int32=\\\"$4` when it is part of `mksensor` command.

3. To create a sensor called `Sensor3` that runs the `/usr/bin/checkhealth` script on the nodes that are listed in the `/u/joe/common_nodes` file, enter:

```
mksensor -N /u/joe/common_nodes Sensor3 "/usr/bin/checkhealth"
```

where `/u/joe/common_nodes` contains:

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

4. To create a microsensor called `IBM.msensordq` that uses the shared module `/usr/lib/msensors/msensordq` and requires the parameters `db=abc`, `confirm=yes`, `retry=yes`, and `mirror=no`, enter:

```
mksensor -m IBM.msensordq /usr/lib/msensors/msensordq \  
"db=abc confirm=yes retry=yes mirror=no"
```

Location

`/opt/rsct/bin/mksensor`

mkserver Command

Purpose

Adds a subserver definition to the subserver object class.

Syntax

```
mkserver -c CodePoint -s Subsystem -t Type
```

Description

The **mkserver** command adds a subserver definition to the **Subserver** object class.

Flags

Item	Description
-c <i>CodePoint</i>	Specifies the <i>CodePoint</i> integer that identifies the subserver. This is the value by which the subsystem knows the subserver. The mkserver command is unsuccessful if this <i>CodePoint</i> value already exists for this subsystem. The limit for <i>CodePoint</i> storage is the same as a short integer (1 through 32,768).
-s <i>Subsystem</i>	Specifies the name that uniquely identifies the subsystem to which the subserver belongs. The mkserver command is unsuccessful if the <i>Subsystem</i> name is not known in the subsystem object class, or if the <i>Subsystem</i> name is that of a known subsystem in the subsystem object class but uses signals as its communication method.
-t <i>Type</i>	Specifies the name that uniquely identifies the subserver. The mkserver command is unsuccessful if the <i>Type</i> name is already known in the Subserver Type 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.

Auditing Events: If the auditing subsystem has been properly configured and is enabled, the **mkserver** command will generate the following audit record (event) every time the command is executed:

Event	Information
SRC_Addserver	Lists in an audit log subsystems that have been added and the entire Object Data Management record.

Examples

To add a subserver definition, enter:

```
mkserver -s srctest -t tester -c 1234
```

This adds a subserver definition to the **Subserver Type** object class, with an owning subsystem of *srctest* and a subserver code point of 1234.

Files

Item	Description
/etc/objrepos/SRCsubsys	Specifies the SRC Subsystem Configuration object class.
/etc/objrepos/SRCsubsvr	Specifies the SRC Subserver Configuration object class.

mkslave Command

Purpose

Invokes the **ypinit** command to retrieve maps from an NIS controller server and starts the **ypserv** daemon to configure a worker server.

Syntax

```
/usr/sbin/mkslave [ -C | -c ] [ -O | -o ] [ -I | -B | -N ] Master
```

Description

The **mkslave** command invokes the **ypinit** command to retrieve maps from the controller server you specify on the command line. The **ypserv** daemon starts after the **ypinit** command has completed successfully. Use the *Master* parameter to specify the host name of the controller server. The controller server specified should already be configured and running.

You can use the System Management Interface Tool (SMIT) **smit mkslave** fast path to run this command.

Flags

Item Description

m

- C** Invokes the **ypinit** command with the **-n** flag. The **mkslave** command continues on errors. This flag is the default.
- c** Stops execution when errors occur.
- O** Overwrites any maps that exist in the domain.
- o** Prevents the overwriting of maps that exist in the domain. This flag is the default.
- I** Invokes the **ypinit** command immediately but does not start the **ypserv** daemon until the next system reboot.
- N** Invokes the **ypinit** command and starts the **ypserv** daemon.
- B** Invokes the **ypinit** command, starts the **ypserv** daemon and configures the **ypserv** to start at system reboot. This flag is the default.

Examples

To invoke the **ypinit** command so that the controller server `host2` will be contacted for maps, enter:

```
mkslave -O host42
```

This command will overwrite the current maps.

Files

Item

`/var/yp/DomainName` directory

Description

Contains the NIS maps for the NIS domain.

> **lksmbcmnt** Command

Purpose

Adds a Server Message Block (SMB) client mount point to the `/etc/filesystems` file and performs the mount operation.

Syntax

```
lksmbcmnt -f MountPoint -d RemoteShare -h ServerName -w wrkgrp  
-c user [-p password] [-m MountTypeName]  
[-A | -a] [-I | -B | -N]  
[-t {rw|ro}] [-u uid] [-g gid] [-x fmode]
```

```
[-V {2.1|3.0.2|auto}] [-S {enabled|required}]  
[-D {desired|required|disabled}] [-E {desired|required|disabled}] [-s spn]
```

p

Description

The **mk smbcmnt** command constructs an SMB client file system entry that is appended to the `/etc/filesystems` file. The **mk smbcmnt** command then mounts the SMB client file system. The **mk smbcmnt** command options are parsed and passed to the **crfs** command that adds the SMB entry to the `/etc/filesystems` file.

The default value for all the mount options (**-t**, **-u**, **-g**, **-x**, **-w**, **-V**, **-S**, **-D**, **-E**) is same as the **mount** command.

Flags

- a**
Specifies that the new SMB entry for the SMB client mount point in the `/etc/filesystems` file must not be automatically mounted after the system restart. This is the default setting.
- A**
Specifies that the new SMB entry for the SMB client mount point in the `/etc/filesystems` file must be automatically mounted after the system restart.
- B**
Adds an entry to the `/etc/filesystems` file and attempts to mount the SMB client file system. This is the default setting.
- c user**
Specifies the username that must be used to access the SMB share.
- d RemoteShare**
Specifies the share name on the SMB server that must be mounted.
- D {desired|required|disabled}**
Specifies whether the SMB client file system needs secure dialect negotiation capability. SMB dialect 3.0.2 implements secure dialect negotiation capability to protect the SMB client against security-downgrade attacks. The valid values are `desired`, `required`, and `disabled`.
- E {desired|required|disabled}**
Specifies whether the SMB client file system requires data encryption. The valid values are `desired`, `required`, and `disabled`.
- f MountPoint**
Specifies the path name in which the SMB share must be mounted.
- g gid**
Specifies the group ID that is assigned to files in the SMB client mount point. The default value is 0.
- h ServerName**
Specifies the name of the remote host or the SMB server. This argument can be specified as a hostname, an IP address, or as a fully qualified domain name.
- I**
Adds the SMB client file system entry to the `/etc/filesystems` file but the directory is not mounted.
- m MountTypeName**
Defines the mount type of the entry that is added to the `/etc/filesystems` file. The mount type determines whether specific file systems can be mounted by using the **-t** flag of the **mount** command. By default, the value of the mount type is not added to the `/etc/filesystems` file.
- N**
Mounts the directory with the specified options but does not modify the `/etc/filesystems` file.

-p password

Specifies the password that is used to grant access to the specific user on the specific SMB server. The specific credentials (server name, username, and password) are added to the `smbcred` file. The password is encrypted. If the **-p** option is not specified, and the credentials do not exist in the `smbcred` file, the command-line prompts you to specify the password, and the credentials are added to the `smbcred` file. If the SMB server or user credentials exist in the `smbcred` file, this option is ignored, and the existing credentials are used for the mount operation.

-s spn

Specifies the service principal name (SPN) that must be used in the SMB client mount points. The format of the **spn** parameter is `cifs/<smbServerHostName>`, where `smbServerHostName` is the fully qualified domain name (FQDN) of the SMB server or the name that the Kerberos resolves as the SMB server. By default, SPN is constructed automatically by the SMB client file system as `cifs/<smbServerHostName>`.

-t {rw|ro}

Specifies whether the SMB client file system must be mounted as read-only.

-u uid

Specifies the user ID that is assigned to the files in the SMB client mount points.

-x fmode

Specifies the owner, group, and other permission bits that are assigned to the files in the SMB client mount points.

-w wrkgrp

Specifies the workgroup to which the SMB server belongs.

-V {2.1|3.0.2|auto}

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.

-S {enabled|required}

Specifies whether the SMB client file system needs digital signature for communication with the SMB server. The valid values are `enabled` and `required`.

Exit status**0**

The command completed successfully.

>0

An error occurred.

Examples

- To add an SMB client file system entry over the `/mnt` mount point to `share1` SMB share on `server1`, and then authenticate as `user1`, enter the following command:

```
mksmbcmnt -f /mnt -d share1 -h server1 -c user1
```

- To add an SMB client file system entry to the `/etc/filesystems` file and mount an SMB client filesystem with default options, run the following command:

```
mksmbcmnt -f /mnt -d /share -h host.xyz.com -c braz2lpa2usr1 -w SMB_WRKGRP
```

This command prompts you to enter a password if the password is not already added in the `/etc/smbcred` file. If the mount operation fails, this command displays the output of the **mount** command. The changes in the `/etc/filesystems` file are retained.

- To add an SMB client file system entry to `/etc/filesystems` file and mount an SMB client file system with the digital signatures set to `required`, run the following command:

```
mksmbcmnt -f /mnt -d /share -h host.xyz.com -c braz2lpa2usr1 -w SMB_WRKGRP -S required
```

Location

/usr/sbin/mksmbcmnt

Files

/etc/smbcred

Stores the credentials of the SMB client file system.

/etc/filesystems

Stores the SMB client file system entry that contains the mount points.

⏪

mksmbcred Command

Purpose

Adds the credentials of the Server Message Block (SMB) client file system to the `/etc/smbcred` file. These credentials can be used for future mount operation of SMB client *shares*.

Syntax

```
mksmbcred -s server_name -u user_name [-p password]
```

Description

When you run the **mksmbcred** command, you must specify a server name and a username. If you do not specify a password as an argument with the **-p** flag, the **mksmbcred** command prompts for a password as a hidden character input. The password is encrypted, and the credentials are stored in the `/etc/smbcred` file as a `server_name|username|password` set. You can create multiple sets of credentials for the same server with different usernames. You can also create multiple sets of credentials with the same username for different servers.

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 defined to access the specific remote host.

-p *password*

Specifies the password for a specific user of a specific remote host.

Exit status

0

The command completed successfully.

>0

An error occurred.

Example

To add credentials for `user1` to mount the SMB client file system on the `xxx.in.ibm.com` server, enter the following command:

```
mksmbcred -s xxx.in.ibm.com -u user1
```

Location

/usr/sbin/mksmbcred

Files

/etc/smbcred

Stores the credentials of the SMB client file system.

mkssys Command

Purpose

Adds a subsystem definition to the subsystem object class.

Syntax

```
mkssys { -p Path -s Subsystem -u UserID } [ -a Arguments ] [ -e StandardError ] [ -i StandardInput ] [ -o StandardOutput ] [ -t Synonym ] [ -O | -R ] [ -d | -D ] [ -q | -Q ] [ -K ] [ -I MessageQueue -m MessageMType ] [ -f StopForce -n StopNormal -S ] [ -E Nice ] [ -G Group ] [ -w Wait ]
```

Description

The **mkssys** command adds a new subsystem definition to the subsystem object class. If no flags are chosen after the **-p**, **-s**, and **-u** flags have been specified, the defaults are **-e /dev/console**, **-i /dev/console**, **-o /dev/console**, **-O**, **-d**, **-Q**, **-K**, **-E 20**, and **-w 20**.

Note: Any auditing performed by the System Resource Controller (SRC) when actions are taken for the subsystem is logged against the login ID of the user who created the subsystem by using the **mkssys** command. For example, if you are logged in with root user authority, the subsystem is added with root user authority as the audit account.

Flags

Item	Description
-a Arguments	Specifies any arguments that must be passed to the command, started as the subsystem. These <i>Arguments</i> variables are passed by the SRC to the subsystem according to the same rules used by the shell. For example, quoted strings are passed as a single argument, and blanks outside a quoted string delimit arguments. Single and double quotes can be used.
-d	Specifies that inactive subsystems are displayed when the lssrc -a command (status all) request is made. By default, if the -D and -d flags are not present, the -d flag is used.
-D	Specifies that inactive subsystems are not displayed when status-all or status-group requests are made.
-e StandardError	Specifies where the subsystem <i>StandardError</i> data is placed. If the -e flag is not specified, the /dev/console file is used for standard error.
-E Nice	Changes the execution priority of the subsystem. Valid values are 0 through 39 (ordinary <i>Nice</i> variables are mapped to all positive numbers). If the -E flag is not present, the subsystem priority defaults to 20. Values between 0 and 19 are reserved for users with root authority.

Item	Description
-f <i>StopForce</i>	Specifies the signal sent to the subsystem when a forced stop of the subsystem is requested. Use only when the subsystem uses signals. The mkssys command is unsuccessful if the <i>StopForce</i> parameter is not a valid signal.
-G <i>Group</i>	Specifies that the subsystem belongs to the <i>Group</i> specified, and that the subsystem responds to all group actions on the <i>Group</i> .
-i <i>StandardInput</i>	Specifies where the subsystem standard input is routed. This field is ignored when the subsystem uses sockets communication. If the -i flag is not specified, by default the /dev/console file is used for standard input.
-I <i>MessageQueue</i>	Specifies that the subsystem uses message queues as the communication method. The <i>MessageQueue</i> variable specifies the message queue key for creating the message queue for the subsystem. Use the ftok subroutine with the subsystem path name as input to generate a unique key.
-K	Specifies that the subsystem uses sockets as its communication method. If a communication method is not specified, sockets communication is used by default.
-m <i>MessageMType</i>	Specifies the message type key the subsystem expects on packets sent to the subsystem by the SRC. Use only when the subsystem uses message queues communication.
-n <i>StopNormal</i>	Specifies the signal sent to the subsystem when a normal stop of the subsystem is requested. Use only when the subsystem uses signals communication. The mkssys command is unsuccessful if the <i>StopNormal</i> variable is not a valid signal.
-o <i>StandardOutput</i>	Specifies where the subsystem standard output is placed. If the -o flag is not specified, by default the /dev/console file is used for standard out.
-O	Specifies that the subsystem is not restarted if it stops abnormally. The default is no restart.
-p <i>Path</i>	Specifies the absolute path to the subsystem executable program.
-q	Specifies that the subsystem can have multiple instances running at the same time.
-Q	Specifies that multiple instances of the subsystem are not allowed to run at the same time and the subsystem is not to share the same interprocess communication (IPC) queue. If the -q flag is not specified, the -Q flag is the default.
-R	Specifies that the subsystem is restarted if the subsystem stops abnormally.
-s <i>Subsystem</i>	Specifies a name that uniquely identifies the subsystem. The mkssys command is unsuccessful if the subsystem name is already known in the subsystem object class.
-S	Specifies that the subsystem uses the signals communication method. You cannot define subservers for a subsystem name when your communication method is signals.
-t <i>Synonym</i>	Specifies an alternate name for the subsystem. The mkssys command is unsuccessful if the synonym name is already known in the subsystem object class.
-u <i>UserID</i>	Specifies the user ID for the subsystem. The <i>UserID</i> that creates the subsystem is used for security auditing of that subsystem.

Item	Description
-w <i>Wait</i>	Specifies the time, in seconds, allowed to elapse between a stop cancel (SIGTERM) signal and a subsequent SIGKILL signal. Also used as the time limit for restart actions. If the subsystem stops abnormally more than twice in the time limit specified by the <i>Wait</i> value, the subsystem is not automatically restarted. By default, if the -w flag is not present, the wait time default is 20 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.

Auditing Events: If the auditing subsystem has been properly configured and is enabled, the **mkssys** command will generate the following audit record (event) every time the command is executed:

Event	Information
SRC_Addssys	Lists in an audit log the name of the subsystem being added to the Object Data Manager (ODM) database and the entire ODM record.

Examples

1. To add a subsystem that uses sockets as its communication type, type the following:

```
mkssys -s srctest -p /usr/lpp/srctest/srctest -u 0 -K
```

This adds a subsystem definition to the subsystem object class, with a communication type of sockets, a user ID of 0 (root), and a subsystem name of srctest.

2. To add a subsystem that uses message queues as its communication type, type the following:

```
mkssys -s srctest -p /usr/lpp/srctest/srctest -u 0 -I 123456 \ > -m 789
```

This adds a subsystem definition to the subsystem object class, with a communication type of message queues, a message queue key of 123456, and a subsystem message type of 789.

3. To add a subsystem that uses signals as its communication type, type:

```
mkssys -s srctest -p /usr/lpp/srctest/srctest -u 0 -S -n 30 \ > -f 31
```

This adds a subsystem definition to the subsystem object class, with a communication type of signals, a stop normal signal of 30, a stop force signal of 31.

4. To add a subsystem that uses sockets as its communication type and is always passed an argument, type:

```
mkssys -s srctest -p /usr/lpp/srctest/srctest -u 0 -a "-x"
```

This adds a subsystem definition to the subsystem object class with a communication type of sockets and a command argument of "-x".

Files

Item	Description
/etc/objrepos/SRCsubsys	Specifies the SRC Subsystem Configuration object class.
/dev/SRC	Specifies the AF_UNIX domain in the socket.h file.

Item	Description
<code>/dev/.SRC-unix</code>	Specifies the location for temporary file sockets.

mkstr Command

Purpose

Creates an error message file.

Syntax

mkstr [-] *MessageFile Prefix File ...*

Description

The **mkstr** command creates a file of error messages that can be removed from a single C source file or from multiple source files. Its use can reduce the size of programs that contain many error diagnostics and reduce system overhead in running such programs, because error messages are then not constantly swapped in and out of the source files.

The **mkstr** command processes each file specified by the *File* parameter, placing a massaged version of the file in a file having the name specified by the *Prefix* parameter followed by the original name.

To process the error messages in the source to the file specified by the *MessageFile* parameter, the **mkstr** command keys on the string ``error(` in the input stream. The string, starting at the `""` (two double quotation marks), is placed in the message file and followed by a null character and a new-line character. The null character terminates the message so it can be easily used when retrieved. The new-line character makes it possible to see the contents of the error message file by using the **cat** command.

The massaged copy of the input file then contains an **lseek** pointer into the file, which can be used to retrieve the message to its appropriate source file, as shown in the following example:

```
char efilename[] = "/usr/lib/pistrings";
int   efil = -1;

error(a1, a2, a3, a4)
{
    char buf[256];
    if (efil < 0) {
        efil = open(efilename, 0);
        if (efil < 0) {
oops:
            perror(efilename);
            exit(1);
        }
    }
    if (lseek(efil, (long) a1, 0) < 0 ||
        read(efil, buf, 256) <= 0)
        goto oops;
    printf(buf, a2, a3, a4);
}
```

Flags

Item Description

- The optional - (minus sign) causes the error messages to be placed at the end of the *MessageFile* for recompiling part of a large **mkstr** program.

Examples

1. To put the error messages from the current directory C source files into the file `pistrings` and to put processed copies of the source for these files into file names prefixed by `xx`, enter:

```
mkstr pistrings xx *.c
```

2. To append the error messages from an additional source file into the file `pistrings`, enter:

```
mkstr - pistrings xx newfile.c
```

Files

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

mksysb Command

Purpose

Creates an installable image of the root volume group either in a file or onto a bootable tape.

Syntax

```
mksysb [ -a ] [ -A ] [ -b number ] [ -e ] [ -F filename ] [ -i ] [ -m ] [ -p ] [ -P ] [ -t argument ] [ -v ] [ -V ] [ -x file ] [ -X ] [ -Z ] [ -G ] [ -N ] [ -M ] [ -T ] [ -C ] device | file
```

Description

The **mksysb** command creates a backup of the operating system (that is, the root volume group). You can use this backup to reinstall a system to its original state if it is corrupted. If you create the backup on tape or user defined file system (UDFS) capable media, the backup is bootable and includes the installation programs that are needed to install from the backup.

Note: If the system has a multibos environment where both instances are mounted, you can restore the backup only by using the **alt_disk_mksysb** command.

You can also use a **mksysb** image to restore another system.

The file system image is in backup-file format. The tape format includes a boot image, a bosinstall image, and an empty table of contents followed by the system backup (root volume group) image. The root volume group image is in backup-file format, starting with the data files and then any optional map files.

One of the data files that the **mksysb** command uses is the `/bosinst.data` file. If the `/bosinst.data` file does not exist, the `/var/adm/ras/bosinst.data` file is copied to `/` (root). The **mksysb** command always updates the `target_disk_data` stanzas in the `bosinst.data` file to match the disks currently in the root volume group of the system where the **mksysb** command is running.

If you are using a customized `/bosinst.data` file and do not want the `target_disk_data` stanzas that are updated, you must create `/save_bosinst.data_file`. The **mksysb** command does not update `/bosinst.data` if the `/save_bosinst.data_file` exists.

Notes:

1. When the **mksysb** command is running, ensure that system activity is minimal.
2. The image that the **mksysb** command creates does not include data on raw devices or in user-defined paging spaces.
3. If you are using a system with a remote-mounted `/usr` file system, you cannot reinstall your system from a backup image.

4. The **mksysb** command might not restore all device configurations for special features, such as `/dev/netbios` and some device drivers that are not shipped with the product.
5. The **mksysb** command uses the **backup** command to create an archive image. The **mksysb** command also saves the extended attributes (EA) format for any Enhanced Journaled File System (JFS2) that are being backed up. It uses the `/usr/bin/mkszfile` shell script to save this information.
6. If you remove the `/dev/ipldevice` before running the **mksysb** command, the 0301-150 bosboot error occurs. This message, in most cases, can be ignored. Confirm the success of the **mksysb** command by the return code.
7. If you are creating a tape backup and have encrypted file systems, you must use the **-Z** flag. You cannot reinstall your system from a tape backup image that contains encrypted file systems.

To create a backup of the operating system to a CD, refer to the [“mkcd Command” on page 523](#). To create a backup of the operating system to a DVD, refer to the [“mkdvd Command” on page 568](#).

Flags

Item	Description
-a	Does not back up extended attributes or Network File System version 4 (NFS4) access control lists (ACLs).
-A	Backs up Data Management API (DMAPI) file system files.
-b number	Specifies the number of 512-byte blocks to write in a single output operation. When the backup command writes to tape devices, the default is 100 for backups by name. The write size is the number of blocks that are multiplied by the block size. The default write size for the backup command that writes to tape devices is 51200 (100 * 512) for backups by name. The write size must be an even multiple of the tape's physical block size.
-C	Specifies whether the <code>/usr/lpp/bos.alt_disk_install/boot_images/bosboot.disk.chrp</code> boot image can be replaced with a new boot image when you create the mksysb image. This flag should be used if the interim fixes that affect the kernel are installed on your system and if you plan to use the alt_disk_mksysb command to install the mksysb image. The bos.alt_disk_install.boot_images fileset must be installed on your system if you want to use the -C flag. You must specify the -i flag to build a new <code>image.data</code> file, when using the -C flag. The new <code>bosboot.disk.chrp</code> image is replaced by the original image at the end of the mksysb image creation. The -C flag is not available when you create the mksysb image when defining a NIM resource. A multibos Base Operating System (BOS) standby instance is not affected by the -C flag. Note: If required, the size of the <code>/usr</code> file system can be increased to include the new boot image by using the -X flag with the -C flag.

Item	Description
-e	<p>Excludes files that are listed in the <code>/etc/exclude.rootvg</code> file from being backed up. The rules for exclusion follow the pattern matching rules of the grep command.</p> <p>If you want to exclude certain files from the backup, create the <code>/etc/exclude.rootvg</code> 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 that are listed in the <code>/etc/exclude.rootvg</code> file, select the Exclude Files field and press the Tab key once to change the default value to yes.</p> <p>For example:</p> <ol style="list-style-type: none"> To exclude all the contents of the directory that is called scratch, edit the exclude file to read as follows: <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;"><code>/scratch/</code></div> To exclude the contents of the directory that is called <code>/tmp</code>, and avoid excluding any other directories that contain <code>/tmp</code> in the path name, edit the exclude file to read as follows: <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;"><code>^./tmp/</code></div> <p>This excludes all contents of the /tmp directory, but the /tmp mount point for the file system is retained. It does not remove other directories or its contents such as /var, /adm, /sw, and /tmp file system.</p> <p>All files are backed up relative to <code>.</code> (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 the <code>^</code> (caret character) as the first character in the search string, followed by <code>.</code> (dot character), followed by the file name or directory to be excluded.</p> <p>If the file name or directory that is being excluded is a substring of another file name or directory, use the <code>^.</code> (caret character followed by dot character) to indicate that the search must begin at the beginning of the line and use the <code>\$</code> (dollar sign character) to indicate that the search must end at the end of the line.</p>
-F filename	Specifies a previously created mksysb image from which a backup tape is created. An attempt is made to make the backup tape bootable. Additionally, this flag must be used with a tape device.
-G	Excludes WPAR file systems from the system backup. This flag is not valid with -N flag.

Item	Description
-i	<p>Calls the mkszfile command, which generates the <code>/image.data</code> file. The <code>/image.data</code> file contains details about volume groups, logical volumes, file systems, paging space, and physical volumes. This information is included in the backup for future use by the installation process.</p> <p>You must use the -i flag. Otherwise, an older <code>/image.data</code> file might be saved that does not contain adequate space requirements to restore the system backup.</p> <p>Note: Before you run the mkszfile command, ensure that enough space is available in the <code>/tmp</code> file to store a boot image. This space is needed during both backup and installation. To determine the amount of space that is needed in the <code>/tmp</code> file, enter the following command:</p> <pre style="background-color: #f0f0f0; padding: 5px;">bosboot -q -a -d device</pre> <p>If you are using UDFS capable device named <code>/dev/usbms0</code>, you must specify <code>/dev/cd0</code> as the device name because the <code>/dev/usbms0</code> device is not supported by the bosboot command. If you use the -X flag with the mksysb command, you do not need to run the bosboot command to determine the amount of space needed in the <code>/tmp</code> file.</p>
-m	<p>Calls the mkszfile command, with the -m flag to generate map files.</p> <p>Note: The use of the -m flag causes the functions of the -i flag to be executed also.</p>
-M	<p>Creates a backup file that is intended for use with the multibos command. The -M flag backs up the <code>/</code>, <code>/usr</code>, <code>/var</code>, and <code>/opt</code> file systems. Do not use the backup to reinstall a system. You must install the <code>bos.alt_disk_install.boot_images</code> fileset at the same level as the system.</p>
-N	<p>Includes file systems that belong to a workload partition (WPAR) in the defined state in the system backup.</p> <p>Note: To be included in the backup, all file systems that belong to a WPAR in the defined state must be in the rootvg volume group.</p>
-p	<p>Disables software packing of the files as they are backed up. Some tape drives use their own packing or compression algorithms.</p>
-P	<p>Excludes files that are listed line by line in the <code>/etc/exclude_packing.rootvg</code>, <code>/etc/exclude_packing.vgname</code>, or <code>/etc/exclude_packing.WPARname</code> file from being packed.</p> <p>For example, to exclude the <code>/etc/filesystems</code> and <code>/usr/bin/zcat</code> file from being packed during the mksysb backup, edit the <code>/etc/exclude_packing.type</code> to add on consecutive lines <code>/etc/filesystems</code> and <code>/usr/bin/zcat</code>.</p> <p>In this case, the file <code>/etc/exclude_packing.type</code> must look like:</p> <pre style="background-color: #f0f0f0; padding: 5px;">/etc/filesystems /usr/bin/zcat</pre> <p>The -P and -p flags are mutually exclusive.</p>
-t argument	<p>Specifies the path to the directory or file system that is used to create a boot image from the mksysb file that is specified by the -F flag. If the -t flag is not used with the -F flag, the boot image is created in the <code>/tmp</code> file by default. Approximately 100 MB of free space is required. After the boot image is created, this space is freed.</p>

Item	Description
-T	<p>Creates backup by using snapshots. This command applies only to JFS2 file systems.</p> <p>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 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 thus, 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.</p> <p>When you specify the -T flag, you must also specify the -i flag. If you do not specify the -i flag, an older <code>/image.data</code> file might not have adequate space requirements. Therefore, a failure might occur when you save data to the snapshot.</p>
-v	Verbose mode. Lists files as they are backed up.
-V	Verifies a tape backup. This flag causes the mksysb command to verify the file header of each file on the backup tape and report any read errors as they occur.
-x <i>file</i>	Excludes the file systems that are listed in the file from the system backup. File system mount points must be listed one per line.
-X	Specifies to automatically expand the /tmp file system if necessary. The /tmp file system might 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 -Z flag is required if you have encrypted file systems and are creating a backup on a tape.



Attention: Use the **-x** flag with caution when you exclude the file systems from the backup of the operating system. The resulting backup might be unusable for system restoration.

Parameters

Item	Description
<i>Device</i> <i>File</i>	Specifies the name of the device or file.

Exit Status

This command returns the following exit values:

Item	Description
0	The command completed successfully.
1	An error in the mksysb command use occurred.
2	An error in the savevg command use occurred. The savevg is a link to mksysb .
3	An error occurred before any file systems were mounted.
4	Exit because of trap.
5	Exit because there was no space.
6	Exit because a volume group name was not valid.

Examples

1. To generate a system backup and create an `/image.data` file (generated by the **mkszfile** command) to a tape device named `/dev/rmt0`, enter the following command:

```
mksysb -i /dev/rmt0
```

2. To generate a system backup and create an `/image.data` file with map files (generated by the **mkszfile** command) to a tape device named `/dev/rmt1`, enter the following command:

```
mksysb -m /dev/rmt1
```

3. To generate a system backup with a new `/image.data` file, but exclude the files in directory `/home/user1/tmp`, create the file `/etc/exclude.rootvg` containing the line `/home/user1/tmp/`, and enter the following command:

```
mksysb -i -e /dev/rmt1
```

This command backs up the `/home/user1/tmp` directory but not the files it contains.

4. To generate a system backup file named `/mksysb_images/node1` and a new `/image.data` file for that image, enter the following command:

```
mksysb -i /mksysb_images/node1
```

Note: This file is not bootable and can be installed only by using Network Installation Management (NIM).

5. After running the **mkszfile** command independently, to generate a system backup on the tape device `/dev/rmt0`, and then to verify the readability of file headers, enter the following command:

```
mksysb /dev/rmt0 -V
```

6. To generate a system backup file named `/mksysb_images/mksysb1` to be used with the **multibos** command, and to create an `/image.data` file for that image, enter the following command:

```
mksysb -iM /mksysb_images/mksysb1
```

7. To generate a system backup and create an `/image.data` file (generated by the **mkszfile** command) to a UDFS capable device named `/dev/usbms0`, enter the following command:

```
mksysb -i /dev/usbms0
```

Note: For information backing up a volume group, see the **listvgbackup** command. To restore individual files from a volume group backup, see the **restorevgfiles** command.

Files

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

> **mksysb_iso** Command

Purpose

Creates a system backup image (**mksysb**) from the root volume group (**rootvg**) a bootable ISO 9660 format media from a previously created system backup image (**mksysb**).

Syntax

```
mksysb_iso -O <iso_image>
[-m <mksysb_image> | -M <mksysb_target>]
[-C <iso_fs_dir>] [-V <fs_volume_group>]
[-p <pkg_source_dir>]
[-i <image.data_file>] [-u <bosinst.data_file>]
[-e] [-P] [-l <package_list>] [-b <bundle_file>]
[-x <filename>] [-T]
[-z <custom_file>] [-D] [-n] [-c] [-a] [-A] [-Z] [-G]
```

Description

The **mksysb_iso** command creates a system backup image (mksysb) from the root volume group (rootvg) or from a previously created mksysb image to a bootable ISO 9660 image.

If you do not specify any file systems or directories as command parameters, the **mksysb_iso** command creates the necessary file systems. Except for the file system that is created for the ISO image, the **mksysb_iso** command removes other file systems that are created while the command is executed. The **mksysb_iso** command also checks the specified file systems for adequate space and write access.

If you specify a previously created mksysb image with the **-m** flag, the version of the mksysb image must match the version of the system where this command is executed. For instance, if you are running this command on an AIX 7.3 system, the version of the specified mksysb image must also be AIX 7.3.

Flags

-a

Does not back up extended attributes or Network File System version 4 (NFSv4) access control lists (ACLs). You cannot use this flag with the **-m** flag.

-A

Backs up data management application programming interface (DMAPI) file systems. You cannot use this flag with the **-m** flag.

-b *bundle_file*

Specifies the full path name of the file that contains a list of filesets that must be installed after the mksysb image is restored. This file is copied to the `./usr/sys/inst.data/user_bundles/bundle_file` location in the ISO file system (CDRFS) and the file that contains a list of filesets is copied to random access memory (RAM) if the CD-ROM drive is unmounted. The file is listed as `BUNDLES=/. ./usr/sys/inst.data/user_bundles/bundle_file` in the `bosinst.data` file.

-c

Does not compress files when the files are backed up. You cannot use this flag with the **-m** flag.

-C *iso_fs_dir*

Specifies the file system that is used to create the ISO file system structure. The specified file system must have enough space to store the system backup image and other necessary operating system files.

If you do not specify the **-C** flag and the `/mksysb_iso/iso_fs` directory exists, the **mksysb_iso** command uses that directory. If you do not specify the **-C** flag and the `/mksysb_iso/iso_fs` directory does not exist, the **mksysb_iso** command creates the `/mksysb_iso/iso_fs` directory and removes the directory after the command execution completes. The command creates the file system in the volume group that is indicated with the **-V** flag or the command creates a root volume group if the **-V** flag is not used.

-D

Displays the debug information. By default, the debug information is not displayed.

-e

Excludes the files and directories from the backup image that is listed in the `/etc/exclude.volume_group` file. You cannot use this flag with the **-m** flag.

-i image.data_file

Specifies the `image.data` file. This data file takes precedence over the `image.data` file in the `mksysb` image. If you do not specify the `-i` flag, the `mksysb_iso` command restores the `image.data` file from the specified `mksysb` image, or generates a new `image.data` file while creating the `mksysb` image.

-l package_list

Specifies the file that contains a list of additional packages that you want to copy to the `./usr/lpp/inst.images` directory of the ISO file system. The images are copied from the location that is specified with the `-p` flag. If you use the `-l` flag, you must also use the `-p` flag.

-m mksysb_image

Specifies a previously created `mksysb` image. If you do not specify the `-m` flag, the `mksysb_iso` command calls the current `mksysb` image. For more information about where the `mksysb` image is located, see the `-M mksysb_target` flag.

-M mksysb_target

Specifies the directory or file system where the `mksysb` image is stored if a previously created backup image is not specified by using the `-m` flag. The directory must exist and must have enough space to store the temporary backup image that is created by the `mksysb_iso` command completes execution. The temporary backup image is removed after the `mksysb_iso` command. If the `-M` flag is not used and an `mksysb` image is not specified, the `mksysb_iso` command verifies that the `/mksysb_iso/mksysb_image` directory exists. If the directory does not exist, the `mksysb_iso` command creates a separate file system, `/mksysb_iso/mksysb_image`, where the `mksysb` image is temporarily stored. The command creates the file system in the volume group that is specified by using the `-V` flag or in the root volume group if the `-V` flag is not used.

-n

Backs up user volume group information and administration data files. This flag backs up files such as `/tmp/vgdata/vgname/vgname.data` and map files, if these files exist. This flag does not back up user data files. You can use this flag to back up user volume group without restoring user data files. You cannot use this flag on root volume groups. You cannot use this flag with the `-m` flag.

-O iso_image

Specifies the file name or the full path name of the ISO image that must be created. This flag is mandatory. If only a file name is specified, the default `/mksysb_iso/iso_images` directory is used to create the ISO image file. If the directory does not exist, the `mksysb_iso` command creates a separate file system and mounts the ISO image file onto that directory. The `mksysb_iso` command creates the file system in the volume group that is specified by using the `-V` flag or in the root volume group if the `-V` flag is not used.

-p pkg_source_dir

Specifies the name of the directory or device that contains the device and kernel package images. If a device is specified, the device must be a optical device (for example, `/dev/cd0`).

-P

Creates mapping information from logical partition to physical partition (PP) during the `mksysb` creation. You cannot use this flag with the `-m` flag.

-T

Backs up the root volume group by using snapshots. This flag applies only to enhanced journaled file system (JFS2).

When you specify the `-T` flag, external JFS2 snapshots are created. Snapshots are a point-in-time image of a JFS2 file system. Therefore, snapshots 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 the 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 thus, snapshots are not used for backing up the file system. The use of the `-T` flag does not affect any JFS file systems that are present in the volume group. These file systems are backed up by using the previous methods of backup. You cannot use this flag with the `-m` flag.

-u bosinst.data_file

Specifies the `bosinst.data` file. This data file takes precedence over the `bosinst.data` file in the `mksysb` image. If you do not specify the `-u` flag, the `mksysb_iso` command restores the `bosinst.data` file from the specified `mksysb` image, or generates a new `bosinst.data` file when the `mksysb` image is created.

-V fs_volume_group

Specifies the volume group that must be used when the `mksysb_iso` command creates the file systems. If the `-V` flag is not specified and a file system does not exist (because a file system was not specified by using any other flags), the root volume group (`rootvg`) is used as the default volume group to create file systems. If the `mksysb_iso` command creates file systems in the backup volume group, the file systems are not included in the system backup image (`mksysb`). Except for the file system created for the ISO image (`-O` flag), other file systems that are created by the `mksysb_iso` command are removed after the `mksysb_iso` command completes execution.

-x file

Excludes the file systems that are listed in the specified file from the system backup image (`mksysb`). File system mount points must be listed one per line. You cannot use this flag with the `-m` flag.

-z custom_file

Specifies the full path name of the file that must be copied to the root directory of the ISO file system (CDRFS). The file can be a customization script that is specified in the `bosinst.data` file, such as `CUSTOMIZATION_FILE=filename`.

For example, if the `my_script` file is in the `/tmp` directory on the system in which the `mksysb_iso` command is running, enter `mksysb_iso -z /tmp/my_script` and specify `CUSTOMIZATION_FILE=my_script` in the `bosinst.data` file. For more information about the `bosinst.data`, see `-u` flag.

-Z

Indicates that the encrypted file system (EFS) information for all the files, directories, and file systems must not be backed up.

Examples

1. To generate a system backup ISO image, `backup1.iso`, in the `/mksysb_iso/iso_images` directory, enter the following command:

```
mksysb_iso -O backup1.iso
```

2. To generate a system backup image, `backup2.iso`, in the `/backup_isos` directory, enter the following command:

```
mksysb_iso -O /backup_isos/backup2.iso
```

3. To create an `mksysb` image in the `/mnt/temp_mksysb` directory, and create a bootable ISO image, `/backup_isos/backup3.iso`, enter the following command:

```
mksysb_iso -O /backup_isos/backup3.iso -M /mnt/temp_mksysb
```

Files

`/usr/bin/mksysb_iso`

Contains the `mksysb_iso` command.



mkszfile Command

Purpose

Saves the system state for reinstallation on the current system or another system.

Syntax

```
mkszfile [ -X] [ -d directory ] [ -m] [ -G] [ -N] [-x file ]
```

Description



Attention: The **mkszfile** command overwrites an existing **/image.data** file with new information.

The **mkszfile** command saves the system state for reinstallation on the current system or on another system. The information saved includes the following:

- System installation information
- Logical volume information for the root volume group
- File system information.

The saved information allows the **bosinstall** routine to recreate the logical volume information as it existed before the backup.

The **mkszfile** command creates the **/image.data** file. The contents of this file are defined by the system in which the image was created. The user can edit the **/image.data** file before calling the **mksysb** command. The **mksysb** command, in turn, only backs up the file systems specified in the **/image.data** file, which reflects the requirements of the **rootvg** file system.

All the saved information is obtained using list commands. The commands are listed in the **/image.data** file as comments for the user's reference when editing this file.

Files on tape cannot be changed. However, in order to override the data files on the tape, the user can create a diskette with the desired files.

The **mkszfile** command checks to be sure there is at least 8MB of free space available in the **/tmp** file system for the boot image.

Note:

1. Before running the **mkszfile** command, ensure that enough space is available in the **/tmp** file to store a boot image. This space is needed during both backup and installation. To determine the amount of space needed in the **/tmp** file, issue one of the following commands: **bosboot -qad rmt** or **bosboot -qad ipldevice**.
2. If you remove the **/dev/ipldevice** prior to executing the **mkszfile** command, the 0301-150 bosboot error occurs. This message, in most cases, can be ignored. Confirm the success of the **mkszfile** command by the return code.

Flags

Item	Description
-m	<p>Creates map files that specify the mapping of the logical-to-physical partitions for each logical volume in the volume group. This mapping can be used to allocate the same logical-to-physical mapping when the image is restored. The map file locations are stored in the MAPFILE field in the /image.data file for each logical volume. Sufficient space would exist in the /tmp file system for map creation because the installation routines place the maps in the /tmp file system before issuing the mklv command.</p> <p>For example, for the hd7 logical volume, the location of the map file is /tmp/vgdata/rootvg/hd7.map. The MAPFILE field in the /image.data file for the hd7 logical volume is under the entry MAPFILE=/tmp/vgdata/rootvg/hd7.map.</p> <p>The map files in the backup image are copied after the /bosinst.data and /image.data files.</p>
-N	<p>Includes file systems that belong to a workload partition (WPAR) in the defined state in the /image.data file.</p> <p>Note: To be included in the /image.data file, all file systems that belong to a WPAR in the defined state need to be in the rootvg volume group.</p>
-X	Expands /tmp if needed.
-d	Write the image.data file to the specified directory instead of / .
-G	Excludes the WPAR file systems from the /image.data file. This flag is not valid with -N flag.
-x file	Excludes the file systems that are listed in the file from the image.data file. File system mount points must be listed one per line.

Note: Use care when excluding file systems as a resulting backup can be unusable for system restoration.

Files

Item	Description
/usr/bin/mkszfile	Contains the mkszfile command.

mktcpip Command

Purpose

Sets the required values for starting TCP/IP on a host.

Syntax

```
mktcpip { -S Interface | -h HostName -a Address -i Interface [ -s ] [ -m SubnetMask ] [ -r RingSpeed ] [ -t CableType ] [ -g DefaultGateway ] [ -n NameServerAddress [ -d Domain ] ] [ [ -c Subchannel ] -D Destination ] }
```

Description

The **mktcpip** command sets the required minimal values required for using TCP/IP on a host machine. These values are written to the configuration database.

Note: The **mktcpip** command currently supports IPv4 only.

The basic functions of the **mktcpip** command include:

- Setting the host name in both the configuration database and the running machine.
- Setting the IP address of the interface in the configuration database.
- Making entries in the **/etc/hosts** file for the host name and IP address.
- Setting the domain name and IP address of the nameserver, if applicable.
- Setting the subnetwork mask, if applicable.
- Adding a static route to both the configuration database and the running machine, if applicable.
- Starting the specified TCP/IP daemons.

You can use the System Management Interface Tool (SMIT) **smit mktcpip** fast path to run this command.

Flags

Item	Description
-a <i>Address</i>	Sets the Internet address of the host. Specify the address in dotted decimal notation. Each network interface on the host should have a unique Internet address. The following is the standard format for setting the Internet address: <pre>127.10.31.2</pre>
-c <i>Subchannel</i>	Specifies the subchannel address for a System/370 channel adapter.
-D <i>Destination</i>	Sets the destination address for a static route. Specify the address in dotted decimal notation. The following is the standard format for setting the destination address for a static route: <pre>192.9.52.1</pre>
-d <i>Domain</i>	Specifies the subdomain name that is used by the host machine. The subdomain name must be specified in the following format: <pre>subdomain.subdomain.rootdomain</pre>
-g <i>DefaultGateway</i>	Adds the default gateway address to the routing table. Specify the address in dotted decimal notation. The following is the standard format for setting the default gateway address: <pre>192.9.52.0</pre>
-h <i>HostName</i>	Sets the name of the host. If using a domain naming system, the domain and any subdomains must be specified. The following is the standard format for setting the host name: <pre>hostname</pre> The following is the standard format for setting the host name in a domain naming system: <pre>hostname.subdomain.subdomain.rootdomain</pre>
-i <i>Interface</i>	Specifies a particular network interface, for example: <pre>tr0</pre>

Item	Description
-m <i>SubnetMask</i>	Specifies the mask the gateway should use in determining the appropriate subnetwork for routing. The subnet mask is a set of 4 bytes, as in the Internet address. The subnet mask consists of high bits (1s) corresponding to the bit positions of the network and subnetwork address, and low bits (0s) corresponding to the bit positions of the host address.
-n <i>NameServerAddress</i>	Specifies the Internet address of the name server the host uses for name resolution, if applicable. The address should be entered in dotted decimal notation, as follows: <pre>127.1.0.1</pre>
-r <i>RingSpeed</i>	Specifies the ring speed for a token-ring adapter. Valid values for the <i>RingSpeed</i> variable are either 4- or 16-Mbps.
-S <i>Interface</i>	Retrieves information for System Management Interface Tool (SMIT) display.
-s	Starts the TCP/IP daemons.
-t <i>CableType</i>	Specifies cable size for Standard Ethernet or IEEE 802.3 Ethernet networks. Valid values for the <i>CableType</i> variable are dix for thick cable, bnc for thin cable, or N/A for Not Applicable. The -t <i>CableType</i> flag should be used only for Standard Ethernet (en) and IEEE 802.3 Ethernet (et) interfaces.

Examples

To set the required values for starting TCP/IP enter:

```
mktcpip -h fred.austin.century.com -a 192.9.200.4 -i en0 \  
-n 192.9.200.1 -d austin.century.com -s
```

Note: Use the **mktcpip** command only to minimally configure TCP/IP for the first time. For further configuration changes, use the **smitty configtcp** fastpath.

Files

Item	Description
/usr/bin/mktcpip	Contains the mktcpip command.
/etc/resolv.conf	Contains the default system configuration database.
/etc/hosts	Contains the host name and IP address entries.

mkts Command

Purpose

Makes a thin server.

Syntax

```
mkts -i ipaddress -m subnetmask -g gateway [-s speed] [-d duplex] -c cosi [-p size] [-H | -h] [-t] [-l] [-v] [-D] thinserver
```

Description

The `mkts` command creates a thin server so that it can use the common image created with the `mkcosi` command. When a thin server is created, several directories are also created for the thin server to mount and use, including `/root`, `/dump`, `/home`, `/tmp`, `/shared_home`, and `/paging`. If you specify the `-l` flag when creating a thin server, the resulting thin server is a diskless client. That is, all resources are created on the server that calls the `mkts` command, except for the `/root` directory, which is created on the server storing the common image. However, if you do not specify the `-l` flag, the thin server is a dataless client. In this case, only the `/root` directory is created on the server storing the common image; all other directories are created locally on the thin server. If necessary, the 512 MB default size used for the paging can be changed by specifying a size value with the `-p` flag.

Flags

Item	Description
<code>-c <i>cosi</i></code>	Specifies the common image for the thin server to obtain its operating system, which is required for the thin server to start up and run.
<code>-d <i>duplex</i></code>	Specifies the duplex setting (optional). Use this setting to configure the client's network interface. This value can be full or half.
<code>-D</code>	Creates an iSCSI dump device in the Common Operating System Image (COSI) of the thin server. The name of the dump device is <code>dump_cosi_name</code> . In order to allow the <code>sysdumpdev</code> command to correctly set this new iSCSI dump device on the thin server, the thin server must boot in iSCSI mode from the COSI of the thin server.
<code>-g <i>gateway</i></code>	Specifies the thin server gateway.
<code>-h</code>	Defines or uses home resource. The home resource is a network installation management (NIM) home resource. It is a directory that is created on a NIM master or any NIM resource server. The directory is exported to the thin server to be mounted and used. It is basically the thin server's /home directory.
<code>-H</code>	Defines or uses <code>shared_home</code> resource. The <code>shared_home</code> resource is a network installation management (NIM) resource. It is a directory that is shared among all thin servers. The directory is exported and mounted on the clients from the NIM master.
<code>-i <i>ipaddress</i></code>	Specifies a thin server IP address or host name.
<code>-l</code>	Specifies whether local resources are used when configuring the thin server. If you specify this flag, all resources are created remotely from the thin server. If you do not specify this flag, only the <code>/root</code> resource is created remotely from the thin server, and all other resources are created locally on the thin server.
<code>-m <i>subnetmask</i></code>	Specifies the thin server subnet mask.

Item	Description
-p <i>size</i>	Specifies the size (in megabytes) of the paging space for the thin server. The minimum size is 64 MB of paging space. The default size is 512 MB of the paging space. If you specify the size less than 64 MB, 512 MB is used.
-s <i>speed</i>	Specifies speed setting (optional). This is the communication speed to use when configuring the client's network interface. This value can be 10, 100, or 1000.
-t	Defines or uses the TMP resource.
-v	Enables verbose debug output when the <code>mkts</code> 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 `mkts` command.

Examples

1. To define a thin server called `lobo` and have it use a common image called `cosi1` for its operating system with an IP address of `9.3.6.234`, a subnet mask of `255.255.255.0`, and a gateway of `9.3.6.1`, enter:

```
mkts -i 9.3.6.234 -m 255.255.255.0 -g 9.3.6.1 -c cosi1 lobo
```

Location

`/usr/sbin/mkts`

Files

Item	Description
<code>/etc/niminfo</code>	Contains variables used by NIM.

mktun Command

Purpose

Activates tunnel(s).

Syntax

```
mktun [-v 4|6] [-t tid_list] [-i] [-I]
```

Description

Use the **mktn** command to activate tunnel(s). For IBM tunnels, this command initiates the security protocol exchanges between the local and the destination host.

Flags

Item	Description
-i	Initiation flag. If the -i flag is not used, all the tunnels in the tunnel database (or those listed with the -t flag) will be activated. If the -i flag is used, only the tunnels whose tunnel definitions status in the tunnel database "active" will be activated.
-I	If the -I flag is specified, manual tunnels will be activated.
-t	If the -t flag is specified, only the tunnel(s) that follows this flag will be activated. If the -t flag is not used, all tunnel(s) currently defined in the tunnel database will be activated. The <i>tid_list</i> can be a single tunnel ID or a sequence of tunnel IDs separated by "," or "-" (1, 3, 5-7).
-v	The IP version of the tunnels to be activated. The value of 4 specifies IP version 4 tunnels. The value of 6 specifies IP version 6 tunnels. If the -v flag is not used, all tunnels for IP version 4 and IP version 6 will be activated.

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.

mkuser Command

Purpose

Creates a new user account.

Syntax

```
mkuser [ -R load_module ] [ -a ] [ Attribute=Value ... ] Name
```

Description

The **mkuser** command creates a new user account. The *Name* parameter must be a unique string (whose length is administrator-configurable using the **chdev** command). You cannot use the **ALL** or **default** keywords in the user name. By default, the **mkuser** command creates a standard user account. To create an administrative user account, specify the **-a** flag.

To create a user with an alternate Identification and Authentication (I&A) mechanism, you can use the **-R** flag to specify the I&A load module. If you create users without the **-R** flag, you create the users locally. Load modules are defined in the **/usr/lib/security/methods.cfg** file.

The **mkuser** 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 **mkuser** command records the time of the user creation in the **/etc/security/uattr** file in the Unix time. |<

You can use the System Management Interface Tool (SMIT) **smit mkuser** fast path to run this command.

The **mkuser** command always checks the target user registry to make sure the ID for the new account is unique to the target registry. You can also configure the **mkuser** command 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 non-target 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 **mkuser** 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 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 **mkuser** command bypasses a remote registry if the remote registry is not reachable at the time the command is run.

Restrictions on Creating User Names

To prevent login inconsistencies, you should avoid composing user names entirely of uppercase alphabetic characters. While the **mkuser** command supports multi-byte user names, it is recommended that you 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 - (dash), + (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:

Ite	Description
m	
:	Colon
"	Double quote
#	Pound sign
,	Comma
=	Equal sign
\	Back slash
/	Slash
?	Question mark
'	Single quote
`	Back quote

Finally, the *Name* parameter cannot contain any space, tab, or new-line characters.

Flags

Item	Description
-a	Specifies that the user is an administrator. Only the root user can use this flag or alter the attributes of an administrative user.
<i>username</i>	Specifies that the user is a new user.
-R load_module	Specifies the loadable I&A module used to create the user.

Parameters

Item	Description
<i>Attribute=Value</i>	Initializes a user attribute. Refer to the chuser command for the valid attributes and values.
<i>Name</i>	Specifies a unique string. The length of this string is set by an administrator by using the chdev command.

Exit Status

This command returns the following exit values:

Item	Description
0	The command runs successfully and all requested changes are made.
>0	An error occurred. The printed error message lists 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.

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 following authorizations:

- **aix.security.user.audit**
- **aix.security.role.assign**
- **aix.security.group.change**
- **aix.security.user.change**

Files Accessed:

Mode	File
rw	/etc/passwd
rw	/etc/security/user
rw	/etc/security/user.roles
rw	/etc/security/limits
rw	/etc/security/environ

Mode	File
rw	/etc/group
rw	/etc/security/group
> > rw	/etc/security/uattr < <
r	/usr/lib/security/mkuser.default
x	/usr/lib/security/mkuser.sys

Auditing Events:

Event	Information
USER_Create	user

Limitations

Creating a user may not be supported by all loadable I&A modules. If the loadable I&A module does not support creating a user, an error is reported.

Examples

1. To create the `davis` user account with the default values in the `/usr/lib/security/mkuser.default` file, type:

```
mkuser davis
```

2. To create the `davis` account with `davis` as an administrator, type:

```
mkuser -a davis
```

Only the root user or users with the UserAdmin authorization can create `davis` as an administrative user.

3. To create the `davis` user account and set the `su` attribute to a value of `false`, type:

```
mkuser su=false davis
```

4. To create the `davis` user account that is identified and authenticated through the LDAP load module, type:

```
mkuser -R LDAP davis
```

Error Codes

Item	Description
0	The command is successful.
EINVAL	The username argument is not valid (containing characters not valid).
EACCES	The invoker does not have write access to the database files.
EPERM	The user identification and authentication fails if the <code>-a</code> flag is specified and the invoker is not root.
EEXIST	The user already exists.
ENAMETOOLONG	The user name is too long.
other errno	There are other system errors.

Files

Item	Description
<u>/usr/bin/mkuser</u>	Contains the mkuser command.
<u>/usr/lib/security/mkuser.default</u>	Contains the default values for new users.
<u>/etc/passwd</u>	Contains the basic attributes of users.
<u>/etc/security/user</u>	Contains the extended attributes of users.
<u>/etc/security/user.roles</u>	Contains the administrative role attributes of users.
<u>/etc/security/passwd</u>	Contains password information.
<u>/etc/security/limits</u>	Defines resource quotas and limits for each user.
<u>/etc/security/environ</u>	Contains the environment attributes of users.
<u>/etc/group</u>	Contains the basic attributes of groups.
<u>/etc/security/group</u>	Contains the extended attributes of groups.
<u>/etc/security/.ids</u>	Contains standard and administrative user IDs and group IDs.
<u>> > /etc/security/uattr</u>	Contains further user attributes. < <

mkuser.sys Command

Purpose

Customizes a new user account.

Syntax

mkuser.sys *Directory User Group Shell*

Description

The **mkuser.sys** command customizes the new user account specified by the *User* parameter. The **mkuser** command calls the **mkuser.sys** command after it has created and initialized the new account. The **tsm**, **login**, and **getty** commands and the **pam_mkuserhome** module call the **mkuser.sys** command at your login time if you do not have a home directory already.

The program as shipped creates the home directory specified by the *Directory* parameter, with the owner specified by the *User* parameter, the primary group specified by the *Group* parameter, and a copy of the appropriate profile for the user's shell. The shipped program can be replaced at installation by another program to customize local new-user creation. The installation-specific program should adhere to the error conventions of the supplied program.

Note: The shipped **mkuser.sys** file must not be customized directly. If a customized version is required, a new file **/etc/security/mkuser.sys.custom** must be created. The **mkuser.sys** program detects this new program and if it is present on the system, it runs it instead of the original **mkuser.sys**. The shipped **mkuser.sys** file is now a non-volatile file and must not be modified. The installation-specific program must adhere to the error conventions of the supplied program.

Security

Access Control: This command should grant read (r), write (w), and execute (x) access for the root user and members of the security group.

Files Accessed:

Mode	File
r	/etc/passwd
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.

Files

Item	Description
/usr/lib/security/mkuser.sys	Contains the mkuser.sys command.

Note: You cannot use the **/etc/security/mkuser.sys** file to edit with the **chuser**, and **rmuser** commands. To assign default attributes such as *primary group*, *home directory*, and *login shell* to a user, use the **/etc/security/mkuser.default** file.

mkusil Command

Purpose

Creates or attaches a new user-specified installation location (USIL) instance.

Syntax

```
mkusil -R RelocatePath -c Comments [-XFa]
```

Description

The **mkusil** command creates or attaches a new USIL instance.

A user-specified installation location (USIL) is a tracked, relocated installation path that is created by the administrator. The location is tracked by the system and can be used as an alternate installation path for packages that use relocation of file sets or software. Multiple instances or versions of the same software package can be installed on a single system by delegating each installation to a separate USIL. An existing USIL instance can be attached or detached from any given system.

Each USIL instance maintains its own set of Software Vital Product Data (SWVPD) in three **installp** parts:

- *InstallRoot/et*c/objrepos
- *InstallRoot/usr/lib*/objrepos
- *InstallRoot/usr/share/lib*/objrepos

Tip: Current SWVPD object classes include product, lpp, inventory, history, fix, vendor, and lag. Each USIL instance mirrors the default SWVPD structure within the relocated path.

Flags

Item	Description
-a	Attaches an existing installation as a USIL instance.
-c Comments	Specifies the comments to include in the USIL definition.

Item	Description
-F	Overwrites the existing USIL SWVPD in the target path without prompting you. It is appropriate to use this flag for USIL SWVPDs that are detached or removed by the rmusil command.
-R RelocatePath	Specifies the path to a new USIL location, which must be a valid directory. The directory cannot belong to a file system with the CIO mount option.
-X	Expands the space needed automatically.

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/mkusil	Contains the <i>mkusil</i> command.

mkvg Command

Purpose

Creates a volume group.

Syntax

```
mkvg [ -a ] [ -B ] [ -t factor ] [ -S [ -v logicalvolumes ] [ -P partitions ] ] [ -C ] [ -G ] [ -f ] [ -i ] [ -I ] [ -c ] [ -X none|SSD ] [ -L ltgsize ] [ -n ] [ -s size ] [ -V majornumber ] [ -y volume ] [ -M y|s ] [ -p mirrorpool ] [ -O y|n ] [ -N o|n ] [ -r y|n ] [ -e y|n ] > [ -k y|n ] < physicalvolume ...
```

Description

The **mkvg** command creates a new volume group, by using the physical volumes that are represented by the *physicalvolume* parameter. Once the volume group is created, the **mkvg** command automatically varies on the new volume group by using the **varyonvg** command. The exception to this fact is when the volume group is created with the **-C** flag. When the volume group is successfully created, the volume group is not varied on automatically. Instead, the user must manually vary on the volume group.

>| The **mkvg** command by default creates a scalable type of volume group that can accommodate up to 1024 physical volumes, 256 logical volumes, and 32768 physical partitions. By default, the **mkvg** command enables the data encryption in the volume group. |<

The **mkvg** command attempts to determine a proper partition size (**-s**) and factor (**-t**) if none is specified on the command line.

Note:

1. The physical volume is checked to verify that it is not already in another volume group. If the **mkvg** command determines that the physical volume belongs to a volume group that is varied on, it exits without creating the volume group. If the **mkvg** command determines the physical volume belongs to a volume group that is not varied on, the force option (**-f**) must be used to create the volume group. If

the force option is used, the previous contents of the physical volume are lost, so use the force option with caution.

2. To use this command, you must either have root user authority or be a member of the system group.
3. When creating a `small` volume group type (with a maximum of 32 PVs) or a `big` volume group type (with a maximum of 128 PVs), there is a limitation of 1016 physical partitions per PV. When specifying the physical partition size (`-s`), make sure that the value is set large enough so that 1016 physical partitions per PV limit is not violated. For example, a partition size of at least 16 MB would be needed to create a volume group with a 10-GB disk. Using a factor size (`-t`) of 2, a smaller partition size of 8 MB can be used. If a factor value is specified, the maximum number of PVs that can be included in the volume group is `MaxPVs` or factor value.
4. Whenever you create a volume group, the operating system automatically does a `varyon`. However, if you create a volume group with the `-C` flag, the system does not `autovaryon` the volume group at the end of the Concurrent Capable volume group creation. Instead, the `mkvg` command notifies you to manually `varyonvg` the volume group in either nonconcurrent or concurrent mode.
5. If the disk indicates that it is managed by a third-party volume manager, the `mkvg` command fails to add the disk to the volume group. To override and clear the disk of the third-party volume manager use `chpv -C HDiskName`.
6. Only Enhanced Concurrent Capable volume groups are created when the `-c` or `-C` flags are specified.
7. Do not mix 4 KB block physical volumes (PV) with PV blocks of other sizes. The block size of all PVs in the volume group must be the same. You cannot import the volume group that is created with 4 KB block PVs on a version of AIX that does not support 4 KB block PVs.
8. Bad block relocation policy of a volume group is not supported on a volume group that is created with 4 KB block PVs.

Because the VGDA space is increased substantially, every VGDA update operation (creating a logical volume, changing a logical volume, adding a physical volume, and so on) might take considerably longer to run.

You can use the System Management Interface Tool (SMIT) `smit mkvg` fast path to run this command.

Flags

Item	Description
<code>> > -a</code>	Creates a volume group of <code>small</code> type. This type can accommodate up to 32 physical volumes and 255 logical volumes. <<
<code>-B</code>	Creates a <code>big</code> type of volume group. This type can accommodate up to 128 physical volumes and 512 logical volumes. Note: Because the VGDA space is increased substantially, every VGDA update operation (creating a logical volume, changing a logical volume, adding a physical volume, and so on) might take considerably longer to run.
<code>-c</code>	Same as <code>-C</code> flag. Only Enhanced Concurrent Capable volume groups are created.

Item	Description
-C	<p>Creates an Enhanced Concurrent Capable volume group. Use the -C flag with the PowerHA® SystemMirror® enhanced scalability (ES). It is not usable on volume groups and systems that do not use the PowerHA SystemMirror ES product.</p> <p>Use this flag to create an Enhanced Concurrent Capable volume group.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. Enhanced Concurrent volume groups use Group Services. Group Services ships with PowerHA SystemMirror ES and must be configured before activating a volume group in this mode. 2. Only Enhanced Concurrent Capable volume groups are supported when running with a 64-bit kernel. Concurrent Capable volume groups are not supported when running with a 64-bit kernel. 3. Enhanced Concurrent Capable volume groups have multinode varyon protection enabled. See the -N flag for details about multinode varyon protection.
-e y n	<p>Enables the <code>Critical PVs</code> option of the volume group. This flag is available in IBM AIX 7.2 with Technology Level 1, or later.</p>
	<p>y</p> <p>Enables the <code>Critical PVs</code> option of the volume group. If write request failures occur in the mirrored logical volume, the PV is marked as missing and it stops sending I/O requests to the failed mirrored logical volume. If the <code>Critical PVs</code> option is enabled in a volume group, you can import only the volume group into IBM AIX 7.2 with Technology Level 1, or later.</p>
	<p>n</p> <p>The <code>Critical PVs</code> option is not used. This option is the default value.</p>
-f	<p>Forces the volume group to be created on the specified physical volume unless the physical volume is part of another volume group in the Device Configuration Database or a volume group that is active.</p>
-G	<p>Same as -B flag.</p>
-i	<p>Reads the <i>PhysicalVolume</i> parameter from standard input.</p>
-I	<p>Creates a volume group that can be imported to AIX Version 6.1. The <i>LTGSize</i> behaves as if the volume group is created before AIX Version 6.1. If the logical volumes are later created with a strip size that is larger than the supported strip size on AIX Version 6.1 or AIX Version 6.1 (a strip size that is multiplied by the number of disks in an array equals the stripe size), then attempting to import the volume group back to AIX Version 6.1 or AIX Version 6.1 is not supported.</p>

Item	Description
>> -k y n	<p>Enables the data encryption option in the volume group. The -k flag is available in IBM AIX 7.2 with Technology Level 5, or later. You can specify the following values for this flag:</p> <p>y Enables the data encryption option in the volume group. If the data encryption option is enabled in a volume group, you can import the volume group into an AIX LPAR that is running AIX 7 with 7200-05, or later. This option is the default value.</p> <p>n Does not enable the data encryption option in the volume group.</p>
-L ltgsize	<p><<<</p> <p>For volume groups created on AIX Version 6.1 without the -I flag, the -L flag is ignored. When the volume group is varied on, the logical track group size is set to the common maximum transfer size of the disks.</p> <p>For volume groups created on AIX Version 6.1 with the -I flag or for volume groups that are created before AIX Version 6.1, the logical track group size is set to the <i>ltgsize</i>. The valid values for <i>ltgsize</i> are 128, 256, 512, or 1024. In addition, it must be less than or equal to the maximum transfer size of all disks in the volume group. The default <i>ltgsize</i> is 128 KB.</p>
-M y s	<p>Enables mirror pool strictness for the volume group.</p> <p>y Mirror pools must be used on each logical volume in the volume group.</p> <p>s Super-strict mirror pools are enforced on this volume group.</p>
-N o n	<p>o Creates a volume group that is allowed to varyon in nonconcurrent mode in more than one node at the same time. This option is the default value.</p> <p>n Creates a volume group that is not allowed to varyon in nonconcurrent mode in more than one node at the same time. This volume group can no longer be imported on a version of the AIX operating system that does not support the -N flag.</p>
-n	<p>Specifies that the volume group is not automatically available during a system restart. The default value activates the volume group automatically.</p>
-O y/n	<p>Enables the infinite retry option of the logical volume.</p> <p>n The infinite retry option of the logical volume is not enabled. The failing I/O of the logical volume is not retried. This option is the default value.</p> <p>y The infinite retry option of the logical volume is enabled. The failed I/O request is retried until it is successful.</p> <p>Note: The infinite retry option is not supported in the Geographic Logical Volume Manager (GLVM) environment.</p>

Item	Description
-p <i>mirrorpool</i>	Assigns each of the physical volumes that are being added to the specified mirror pool. After mirror pools are enabled in a volume group, the volume group can no longer be imported into a version of AIX that does not support mirror pools.
-P <i>partitions</i>	Total number of partitions in the volume group, where the <i>partitions</i> variable is represented in units of 1024 partitions. Valid values are 32, 64, 128, 256, 512 768, 1024 and 2048. The default is 32 k (32768 partitions). The chvg command can be used to increase the number of partitions up to the maximum of 2048 k (2097152 partitions). This option is only valid with the -S option.
-r <i>y n</i>	Enables the <code>Critical VG</code> option of the volume group. The -r flag can have the following values: <p data-bbox="613 615 1446 926">y The <code>Critical VG</code> option of the volume group is enabled. If the volume group is created with the <code>Critical VG</code> option turned on, any I/O request failure starts writing the logical volume manager (LVM) metadata to check the state of the disk before returning the I/O failure. If the rootvg volume group is set to the <code>Critical VG</code> option and if the volume group loses access to quorum set of disks (or all disks if quorum is disabled), instead of forcing the volume group to an offline state the node crashes and a message is displayed on the console.</p> <p data-bbox="613 947 1446 1029">n The <code>Critical VG</code> option of the volume group is not enabled. It is the default value.</p>
-S	Creates a scalable type of volume group. By default, this volume group can accommodate up to 1024 physical volumes, 256 logical volumes, and 32768 physical partitions. To increase the number of logical volumes, use the -v option. To increase the number of physical partitions, use the -P option. <p data-bbox="613 1230 1471 1451">Note: Increasing <code>maxlvs</code> and <code>maxpps</code> beyond the default values for a scalable volume group can significantly increase the size of the VGDA proportionately. The <code>maxlvs</code> and <code>maxpps</code> values must be increased only as needed because they cannot be decreased. Meanwhile, as the VGDA space increases all VGDA update operations (creating a logical volume, changing a logical volume, adding a physical volume, and so on) can take longer and longer to run.</p>
-s <i>Size</i>	Sets the number of megabytes in each physical partition, where the <i>Size</i> variable is expressed in units of megabytes from 1 (1 MB) through 131072 (128 GB). The <i>Size</i> variable must be equal to a power of 2 (example 1, 2, 4, 8). The default value for 32 and 128 PV volume groups is the lowest value to remain within the limitation of 1016 physical partitions per PV. The default value for scalable volume groups is the lowest value to accommodate 2040 physical partitions per PV.

Item	Description
-t <i>factor</i>	Changes the limit of the number of physical partitions per physical volume, which is specified by the <i>factor</i> variable. The <i>factor</i> must be 1 - 16 for 32 PV volume groups and 1 and 64 for 128 PV volume groups. The maximum number of physical partitions per physical volume for this volume group changes to <i>factor</i> x 1016. The default is the lowest value to remain within the physical partition limit of <i>factor</i> x 1016. The maximum number of PVs that can be included in the volume group is MaxPVs or <i>factor</i> value. The -t option is ignored with the -S option.
-V <i>majornumber</i>	Specifies the major number of the volume group that is created.
-v	Number of logical volumes that can be created. Valid values are 256, 512, 1024, 2048 and 4096. The default is 256. The chvg command can be used to increase the number of logical volumes up to the maximum of 4096. This option is only valid with the -S option. The last logical volume is reserved for metadata.
-X none SSD	Enables PV type restriction for the volume group. This option allows a volume group to be created with a specific restriction based on the PV type. "none" is the default value. "SSD" requires that all PVs in the volume group must be SSD media type PVs. When the PV restriction is turned on, the mkvg command verifies that all PVs meet this condition. Once a PV restriction is turned on, the volume group can no longer be imported on a version of AIX that does not support PV type restrictions. <p>none As an option, there are no PV restriction. Volume group can be formed on any disk type. This option is the default value.</p> <p>SSD As an option, volume group is restricted to SSD PV types. Disks that are listed in physical volume argument must be of type SSD.</p>
-y <i>volume group</i>	Specifies the volume group name rather than 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 that is already defined in the PdDv class in the Device Configuration database for other devices. The volume group name that is created is sent to standard output. <p>The volume group name can contain the following characters: A through Z, a through z, 0 through 9, _ (underscore), - (minus sign), or . (period). All other characters are considered invalid.</p>

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 volume group that contains 3 physical volumes with partition size set to 1 megabyte, type:

```
mkvg -s 1 hdisk3 hdisk5 hdisk6
```

The volume group is created with an automatically generated name, which is displayed and available at system restart time.

```
mkvg -s 2 -t 2 -y newvg hdisk1
```

The volume group `newvg` is created with a physical partition size of 2 MB and maximum number of physical partitions per physical volume of 2032. The configuration that is mentioned in the example means that the size of `hdisk1` cannot be larger than 4064 MB (2032*2)

2. To create a volume group that can accommodate a maximum of 1024 physical volumes and 2048 logical volumes, type:

```
mkvg -S -v 2048 hdisk6
```

Files

Item	Description
<code>/usr/sbin</code>	Directory where the mkvg command is stored.
<code>/tmp</code>	Directory where the temporary files are stored while the command is running.
<code>/dev</code>	Directory where the character device entry for the volume group is created.

mkvgdata Command

Purpose

Creates a file containing information about a volume group for use by the **savevg** and **restvg** commands.

Syntax

```
mkvgdata [ -X ] [ -m ] [ -x file ] VGName
```

Description

The **mkvgdata** command creates a file containing information about a volume group for use by the **savevg** and **restvg** commands. The information includes the list of logical volumes, file systems and their sizes, and the volume group name. One of the following files is created, depending on the type of volume group:

Item	Description
<code>/image.data</code>	Created for information about the root volume group (rootvg). The savevg command uses this file to create a backup image that can be used by the bosinstall routine to reinstall the volume group to the current system or to a new system. The mkvgdata command overwrites this file if it already exists. The /image.data file is located in the <code>/</code> directory.
<code>vgname.data</code>	Created for information about a user volume group. The <code>vgname</code> 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 reinstall the user volume group. The mkvgdata command overwrites this file if it already exists. The vgname.data file is located in the <code>/tmp/vgdata/vgname</code> directory, where <code>vgname</code> is the volume group name.

The information in either of these files can be edited by the user before issuing the **savevg** command.

Flag

Item	Description
-m	Creates map files that specify the mapping of the logical-to-physical partitions for each logical volume in the volume group. This mapping can be used to allocate the same logical-to-physical mapping when the image is restored. The map file locations are stored in the MAPFILE field in the /image.data file for each logical volume. Sufficient space would exist in the /tmp file system for map creation because the installation routines place the maps in the /tmp file system before issuing the mklv command. For example, for the hd7 logical volume, the location of the map file is /tmp/vgdata/rootvg/hd7.map . The MAPFILE field in the /image.data file for the hd7 logical volume is under the entry MAPFILE=/tmp/vgdata/rootvg/hd7.map. The map files in the backup image are copied after the image.data or vgname.data files.
-X	Expands /tmp if needed.
<i>vgname</i>	Name of volume group to backup.
-x file	Excludes the file systems that are listed in the file from the output file. File system mount points must be listed one per line.

Note: Use care when excluding file systems because a resulting backup can be unusable for system restoration.

Files

Item	Description
/image.data	Created when the volume group is rootvg .
/tmp/vgdata/vgname/vgname.data	Created when the volume group is not rootvg and where <i>vgname</i> is the name of the volume group.

mkvirprt Command

Purpose

Makes a virtual printer.

Syntax

```
mkvirprt [ -A AttachmentType ] -d QueueDevice -n Device -q PrintQueue -s DataStream -t PrinterType -T  
mkvirprt -A AttachmentType
```

Description

The **mkvirprt** command creates a virtual printer definition and assigns it to the specified print queue and queue device. A virtual printer definition is a set of attribute values that describe a particular data stream for a particular printer. Before a print job can be queued with the **enq** command, **qprt** command, **lp** command, or **lpr** command, a virtual printer definition must be created for the printer's print queue and queue device.

Printers that support only one printer data stream, such as the 4201-3 Proprinter III, need only one virtual printer defined. Printers that support multiple printer data streams, such as the IBM 4216-31 Page Printer II, need a virtual printer defined for each data stream.

To create a virtual printer definition for a printer attached to an ASCII terminal, use the **-T** flag with the **mkvirprt** command.

After a virtual printer definition is created, its attribute values can be displayed with the **lsvirprt** command and changed with the **chvirprt** command.

The **mkvirprt** command becomes interactive if only the **-A** flag is specified with the command. Prompts are issued requesting the necessary parameter values. Prerequisite spooler queues and spooler queue devices are generated automatically, and all virtual printer definitions needed for the printer are defined with a single invocation of the **mkvirprt** command for the specified attachment type.

When the first prompt asks for a device name, if the device name entered is not that of a printer, or if an * (asterisk) precedes the device name, a list of printers is displayed. Otherwise, the printer type is assumed to be the same as that of the device.

Also, when a prompt asks for a print queue name, the queue name entered may optionally be followed by a colon and a queue device name. If no queue device name is provided, the queue device name is assumed to be the same as the device name.

Note: Queue and device names must begin with an alphabetic character.

You can use the System Management Interface Tool (SMIT) **smit mkvirprt** fast path to run this command.

Flags

Item	Description
-A <i>AttachmentType</i>	<p>Specifies the type of printer attachment. The most common values for the <i>AttachmentType</i> variable value are:</p> <p>Attachment Type Represents</p> <p>local Locally connected printers</p> <p>remote Remote print queues</p> <p>ascii Printers attached to an ASCII terminal</p> <p>file Print output redirected to a regular file.</p> <p>This flag is optional, and if the -A flag is not specified the default attachment type is file. If the -A flag is the only flag specified on the command line, the mkvirprt command goes into interactive mode and executes steps specified in the corresponding .config file.</p>
-d <i>QueueDeviceName</i>	Specifies the name of an existing queue device to which the virtual printer is assigned.
-n <i>DeviceName</i>	Specifies the name of the printer device. Device names include lp0 for printer 0, lp1 for printer 1, and so on.
-q <i>PrintQueueName</i>	Specifies the special file name of an existing print queue to which the virtual printer is to be assigned. Note that you do not have to specify the path name to the file, such as the /dev/lp0 file, you just need to specify lp0 .

Item	Description
-s <i>DataStreamType</i>	Specifies the printer data stream type. Data stream types include: Type Description asc Extended ASCII ps PostScript pcl Hewlett-Packard PCL 630 Diablo 630 855 Texas Instruments 855 gl Hewlett-Packard GL kji Kanji
-t <i>PrinterType</i>	Specifies the printer type. Printer types include 4201-3, ti2115, and so on.
-T	Specifies that the printer is attached to an ASCII terminal.

Examples

1. To make a virtual printer for the asc printer data stream for the 4029 printer attached locally, enter:

```
mkvirprt -A local -d mypro -n lp0 -q proq -s asc -t 4019
```

2. To make a virtual printer for a printer connected to an ENA 4033 network adapter, and to be prompted for the parameter values, enter:

```
mkvirprt -A ena
```

Files

Item	Description
/usr/sbin/mkvirprt	Contains the mkvirprt command.
/etc/qconfig	Contains configuration files.
/usr/lib/lpd/pio/predef/*	Contains predefined printer attribute files.
/var/spool/lpd/pio/@local/custom/*	Contains customized virtual printer attribute files.
/usr/lib/lpd/pio/etc/*.attach	Contains attachment type files.
/usr/lib/lpd/pio/etc/*.config	Contains the configuration file for the printer.
/var/spool/lpd/pio/@local/ddi*	Contains digested virtual printer attribute files.

mkwpar Command

Purpose

Creates a system workload partition (WPAR), or a WPAR specification file.

Syntax

```
/usr/sbin/mkwpar [-a] [-A] [-b devexportsfile ] [-c] [-C] [-E directory] [-d directory ] [-B wparbackupdevice] [-D attribute=value ...] ... [-F] [-g vg ] [-h hostname] [-H architecture] [-i] [-I attribute=value ...] [-k]... [-l] [-L attribute=value...] [-M attribute=value ...] ... [-N attribute=value ...] ... [-P] [-r] [-R attribute=value ...] [-S attribute[+|-]=value ...] [-t ] [-T attribute=value ... ] [-s] [-u userscript] [-X attribute=value ...] [-U [uuid]] { -n wparname [-p [name]] [-e existingwparname -W | -f infile] [-o outfile [-w]] | -p [name] [-n wparname] [-e existingwparname -W | -f infile] [-o outfile [-w]] | -f infile [-n wparname] [-p [name]] [-o outfile [-w]] | -w -o outfile [-n wparname] [-p [name]] [-e existingwparname -W | -f infile] }
```

Restriction:

- White space must be included between a flag and its argument for *attribute=value* type flags. The **mkwpar** command is not supported on the TCB systems. Regardless of locale, only ASCII characters are allowed as arguments to **mkwpar**, **chwp**, or **wparexec**.
- You must not run the **mkwpar** command during the AIX Live Update operation.

In addition to the previous command restrictions, more restrictions follow for the WPAR name:

- Must not be more than 25 bytes.
- Must not contain white space or any of the following symbols:

```
= : / ! ; ` ' " < > ~ & ( ) * + [ ] , . ^ 0 { } | \
```

- Must not start with hyphen (-) or 0.

Description

The **mkwpar** command builds the infrastructure to prepare a system workload partition for use. This command includes the following tasks:

- Creating the configuration data of the workload partition in the workload partition database
- Creating and populating file systems of the workload partition
- Creating an SRC subsystem for the init process of the workload partition
- Defining the resource control profile of the workload partition through Workload Manager

The following options are also available:

- Writing a specification file to simplify creation of other, similar workload partitions
- Starting the workload partitions
- Specifying whether the workload partitions must be automatically started on system start or when **/etc/rc.wpars** is started
- Specifying WPAR specific routing, by using the **-i** and **-I** flags

The **mkwpar** command supports advanced logical volume and file system options by specifying the **image.data** file as an argument to the **mkwpar -L** flag.

The **mkwpar** command supports creating a rootvg WPAR, in which the root file systems are located solely in WPAR storage devices.

Flags

Item	Description
-a	Automatically resolves conflicting static settings if required. Resolvable settings are base directory, host name, and network configuration.
-A	Specifies that the workload partition must be started each time <code>/etc/rc.wpars</code> is run, which is added to the global <code>/etc/inittab</code> to run each time that the system starts. The default is not to start the workload partition automatically. Tip: The workload partition is started immediately upon completion of the <code>mkwpar</code> command. To start the workload partition immediately, use the <code>-s</code> flag.
-b <i>devexportsfile</i>	Specifies an alternative file to use 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, <code>/etc/wpars/devexports</code> is used.
-B <i>wparbackupdevice</i>	Specifies a device that contains a workload partition backup image. This image is used to populate the workload partition file systems. The <code>wparBackupDevice</code> parameter is a workload partition image that is created with the <code>savewpar</code> , <code>mkcd</code> , or <code>mkdvd</code> command. The -B flag is used by the <code>restwpar</code> command as part of the process of creating a workload partition from a backup image. Note: The -B flag is mutually exclusive with the -p flag.
-c	Configures the workload partition to be checkpointable. This option is valid only when more checkpoints or restart software are installed and configured. When you specify this flag, any file systems that are associated with only this flag (for example, through the -M flag) must be remote (for example, <code>vfs=nfs</code>).
-C	Creates a versioned workload partition. This option is valid only when more versioned workload partition software is installed.
-d <i>directory</i>	Specifies a base directory for the workload partition. If you do not specify a directory name, <code>/wpars/<wparname></code> is used.
-D [<i>devname=device name</i> <i>devid=device identifier</i>] [<i>rootvg=yes no</i>] [<i>devtype=[clone pseudo disk adapter cdrom tape]</i>]	Configures exporting or virtualization of a global device into the workload partition every time the system starts. You can specify more than one -D flag to allocate multiple devices. Separate the attribute=value by blank spaces. You can specify the following attributes for the -D flag: devname=device name Specifies the device name to allocate to the workload partition. For pseudo and clone type devices, this command is the full path to the device (that is, <code>/dev/pty10</code>). For storage type devices, it is the logical device short name. devid=device identifier Specifies the unique device identifier of a disk type device to allocate to the workload partition. This attribute applies only to disk, cdrom, or tape type devices. devtype=[clone pseudo disk adapter cdrom tape] Specifies the device type of the device to allocate to the workload partition. rootvg= [yes no] Used to indicate whether the specified disk device is to be used as a rootvg workload partition device. If the <code>rootvg</code> attribute is not specified, the command takes the default of number.
-e <i>existingwparname</i>	Uses an existing workload partition as the source for specification data. This flag is mutually exclusive with the <code>-f</code> flag. Any values that you specify using other <code>mkwpar</code> flags override those from the existing workload partition.
-E	Specifies a directory which contains additional filesets to install when a versioned workload partition is created. If you do not specify a directory name, <code>/usr/sys/inst.images</code> is used. This option is used only during creation of a versioned WPAR.
-f <i>infile</i>	Indicates a specification file from which default values are read. This flag is mutually exclusive with the <code>-e</code> flag. Any values that you specify by using other <code>mkwpar</code> flags, override those flags from the loaded specification file.
-F	Forces the command to continue rather than fail for most error conditions.
-g <i>vg</i>	Indicates the default volume group. If you do not specify a value, <code>rootvg</code> is used. This volume group is used for each localfs file system whose volume group is not specified by using the <code>vg</code> parameter of the -M flag. The volume group for file systems that you specified in the <code>image.data</code> file, supersedes the volume group that is specified with the -g flag.
-h <i>hostname</i>	Specifies a host name for the workload partition. you do not specify a value, the <code>mkwpar</code> command uses the workload partition name for the host name.

Item	Description														
-H <i>architecture</i>	<p>Creates an architecture compatible workload partition. The valid architecture values are <i>{pwr4, ppc970, pwr5, pwr6, pwr7, and pwr8}</i>. The architecture value must be lower than, or equal to, the system hardware level. The applications in the workload partition are presented with the lowest common denominator of the specified architecture. If the workload partition is checkpointable, the workload partition must be able to migrate between systems with hardware levels greater than, or equal to, the workload partition architecture.</p> <p>Note: Values <i>pwr5</i> and <i>ppc970</i> are not compatible with each other. You cannot create a <i>ppc970</i> compatible WPAR on a POWER5 processor-based system even though the <i>ppc970</i> processor preceded the POWER 5 processor.</p>														
-i	<p>Enables WPAR specific routing for the workload partition.</p> <p>A default route is not created automatically. The -I flag is used to specify routes, including the default route.</p> <p>By default, outgoing network traffic from a workload partition is routed as if it is being sent from the global environment:</p> <ul style="list-style-type: none"> • 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. <p>If you enable WPAR specific routing by specifying the -i flag, the workload partition creates and uses its own routing table for outgoing traffic.</p> <p>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 -N flag. You can create explicit additions to the routing table of the workload partition using the -I flag.</p>														
-I <i>attribute=value ...</i>	<p>Adds routing table entries to those tables that are automatically created when WPAR specific routing is in effect. You can specify more than one -I flag to configure multiple routes. Using the -I flag automatically enables WPAR specific routing as described under the -i flag.</p> <p>You can specify the following attributes with the -I flag:</p> <table border="0" style="width: 100%;"> <tr> <td style="vertical-align: top;">rtdest=destination</td> <td>(Required) Identifies the host or network to which you are directing the route. You can specify the value by using either symbolic name or 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.</td> </tr> <tr> <td style="vertical-align: top;">rtgateway=gateway</td> <td>(Required) Identifies the gateway to which packets are addressed. You can specify the value by using either symbolic name or numeric address.</td> </tr> <tr> <td style="vertical-align: top;">rtnetmask=A.B.C.D</td> <td>Specifies the network mask to the destination address.</td> </tr> <tr> <td style="vertical-align: top;">rtprefixlen=n</td> <td>Specifies the length of a destination prefix, which is the number of bits in the netmask. The value must be a positive integer.</td> </tr> <tr> <td style="vertical-align: top;">rttype={net host}</td> <td>Forces the rtdest attribute to be interpreted as the specified type.</td> </tr> <tr> <td style="vertical-align: top;">rtinterface=if</td> <td>Specifies the interface, for example, en0, to associate with the route so that packets are sent by using the interface when the route is chosen.</td> </tr> <tr> <td style="vertical-align: top;">rtfamily={inet inet6}</td> <td>Specifies the address family.</td> </tr> </table>	rtdest=destination	(Required) Identifies the host or network to which you are directing the route. You can specify the value by using either symbolic name or 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	(Required) Identifies the gateway to which packets are addressed. You can specify the value by using either symbolic name or 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, en0, to associate with the route so that packets are sent by using the interface when the route is chosen.	rtfamily={inet inet6}	Specifies the address family.
rtdest=destination	(Required) Identifies the host or network to which you are directing the route. You can specify the value by using either symbolic name or 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	(Required) Identifies the gateway to which packets are addressed. You can specify the value by using either symbolic name or 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, en0, to associate with the route so that packets are sent by using the interface when the route is chosen.														
rtfamily={inet inet6}	Specifies the address family.														
-k	<p>Specifies the path to a user provided post installation customization script. The script is run in the global environment after the WPAR is created while WPAR file systems are mounted. The post customization script is called with the WPAR name as the first argument, and the WPAR base directory as the second argument. If the script exits with a nonzero return code, a warning is printed, but the mkwpar command has not failed.</p>														
-l	<p>Creates private and writable versions of the /usr and /opt file systems.</p>														

Item

-L [**image_data**= *imagedatafile*] [**shrink**=
{**yes**|**no**}] [**ignore_maps**= {**yes**|**no**}]

Description**image_data**

Specifies the path to the **image.data** file to be used for logical volume and file system options. The format of the **image.data** file is described in *Files Reference* and the **/usr/lpp/bosinst/image.template** file. File system specifications in the **image.data** file supersede file system specifications in the **Specifications** File. The **-c** flag and **-L image_data=** flags are mutually exclusive.

shrink

Specifies that the **LV_MIN_LPS** attribute, rather than the **LPS** attribute, must be used to determine the number of logical partitions for the logical volume. The **LV_MIN_LPS** attribute is from the **lv_data** stanzas from the file that the **image_data** attribute specifies. This attribute can minimize the amount of disk space that is required for a workload partition file system. This attribute has no effect if the **image_data** attribute is not specified.

ignore_maps

Specifies that the **MAPFILES** attribute must not be used to provide a disk mapping for the logical volumes that are associated with a workload partition. The **MAPFILES** attribute is from the **lv_data** stanzas from the file that the **image_data** attribute specifies. This attribute has no effect if the **image_data** attribute is not specified.

ignore_lvs

Specifies that the information from the **lv_data** stanzas is not used when the **image.data** file is being processed. The logical volumes are created with the default characteristics when the file systems are created. This attribute has no effect if the **image.dita** attribute is not specified. If this attribute is specified, the **ignore_maps** attribute is ignored.

-M **directory**=*dir* [**vfs**=*type*]
[**size**=*sizespec*] [**vg**=*volume**group*]
[**logname**=*loglv*] [**dev**=*device**path*]
[**host**=*remote**host*] [=]
[**mountopts**=*mountopts*]

Specifies mount configuration attributes. Attributes must be separated by a blank space. You can specify more than one **-M** flag. By default, the **/usr** and **/opt** file systems of the workload partition are mounted over the global **/usr** and **/opt** file systems in read-only mode. The **/proc** file system of the workload partition is mounted over the global **/proc** file system in read/write mode. New logical volumes that are created in **rootvg** for **/**, **/var**, **/tmp**, and **/home**. The default settings for a specified file system can be overridden by using the **-M** flag with the **directory** attribute set to the file system name. You can specify more file systems with additional **-M** flags. The **directory** attribute denotes the directory within the workload partition where the device must be mounted.

File system specifications in the **-M** flag supersede file system specifications in the **image.data** file.

There are 4 basic workload partition mount forms:

localfs

Disk-based file system (**vfs**=**jfs** or **vfs**=**jfs2**) to be created at the location that is specified by the value of the **directory** within the directory structure of the workload partition. If you specify a **dev** attribute, it denotes an existing logical volume in the global environment, which is to be used to host the file system. For localfs file systems, you must specify the **size** attribute. Other optional attributes, which are of the form **attr=value**, include those attributes in the following list:

logname

Specifies the log device to use for this file system. This attribute must be specified only if the default log device that the file system uses is insufficient.

- For **vfs=jfs2**, the default is to use an inline log.
- For **vfs=jfs**, the default is that the file system uses an existing log device if available. Otherwise, it creates one. When the **logname** attribute is being specified, make sure that the named log device exists.

mode

Specifies the octal permission mode to assign to the base directory of this file system. The default is 755.

size

Specifies the size of the file system that is created in a format acceptable to the **crfs** command.

vg

Specifies the volume group in which the file system (if no existing logical volume device is specified by using the **dev** attribute) is created. If you do not specify a value, the volume group that is specified in the **-g** flag is used. If you do not specify the **-g** flag, **rootvg** is assumed.

Specifies other options to pass to the **crfs** command when the file system is being created. Options are passed directly to the **crfs** command so the value must be in the form that is required by the **crfs** command.

Item	Description
-M directory = <i>dir</i> [vfs = <i>type</i>] [size = <i>sizespec</i>] [vg = <i>volume</i> group] [logname = <i>loglv</i>] [dev = <i>device</i> path] [host = <i>RemoteHost</i>] [=] [mountopts = <i>mountopts</i>] (continued)	<p>Restriction:</p> <p>Do not specify any options to the crfs command that correspond to the flags in the mkwpar command.</p> <p>The mkwpar command must not be specified by using the attribute because incorrect results might occur:</p> <ul style="list-style-type: none"> • -a logname=<i>lvname</i> (<i>logname</i>) • -a size=<i>value</i> (<i>size</i>) • -d device (<i>dev</i>) • -g volumegroup (<i>vg</i>) • -m mountpoint (<i>directory</i>) • -v vfstype (<i>vfs</i>) <p>For more information, see crfs documentation for further information about the crfs command.</p>
mountopts	<p>Specifies the mount options (corresponding to the "options" attribute in an <code>/etc/filesystems</code> stanza). If you do not specify a mount option, by default, no mount flags are used. Option values that you can specify correspond to the -o options of the mount command.</p>
namefs	<p>Specifies that the global directory that is specified by the dev attribute is mounted over the directory that is specified by the directory attribute in the file system structure of the workload partition. The only other attribute that is applicable to a namefs mount is mountopts. For the namefs type, you cannot map the <code>/var</code>, <code>/opt</code>, <code>/usr</code>, <code>/tmp</code>, or <code>/proc</code> file system of a workload partition with write privileges to a real <code>/</code>, <code>/var</code>, <code>/opt</code>, <code>/usr</code>, <code>/tmp</code>, or <code>/proc</code> file system.</p> <p>The namefs mount can also be used with rootvg workload partitions. In this case, the content of the namefs mount is not saved by using the savewpar command.</p>
nfs	<p>Specifies that the directory that is specified by the dev attribute on the system that is exported by the host attribute is mounted over the workload partition directory. The only other attribute that is applicable to a nfs mount is mountopts.</p> <p>Requirement: The global system and the workload partition must both have root permissions to the NFS device. You can give the global and the WPAR root permission to the NFS device, when you export the NFS mount, by specifying the root access for the host names of both the global system and the workload partition. When an NFS device is mounted, you cannot map the <code>/</code>, <code>/var</code>, <code>/opt</code>, or <code>/usr</code> file system of a workload partition with write privileges to a real <code>/</code>, <code>/var</code>, <code>/opt</code>, or <code>/usr</code> file system.</p>
directory	<p>Specifies that the directory that is specified by the directory attribute is added to the file system structure of the workload partition. No file system is created. Use this attribute to reduce the number of file systems to manage in a workload partition, such as by eliminating the separate file systems for <code>/tmp</code> and <code>/var</code>. Ensure that the size of the containing file system is adjusted accordingly.</p> <p>Note: A directory mount cannot be used for <code>/usr</code> or <code>/opt</code>.</p>
-n wparname	<p>Specifies the name for the workload partition to be created. You must specify a name, either by using the -n flag, or in a specification file by using the -f flag, unless the -p name or both -w and -o flags are used.</p>

Item	Description
-N <i>attribute=value</i>	<p>Specifies network configuration attributes. Separate the <i>attribute=value</i> pairs by blank spaces. You can specify more than one -N flag to configure multiple IP addresses. You must always specify the address or the address6 attribute when you use the -N flag. Any other values that are not specified are taken from the settings of the global system. If you do not specify the -N flag, the mkwpar command attempts to discover an appropriate IP address for the workload partition. To do that, the mkwpar command performs the gethostbyname subroutine on the workload partition host name (specified with the -h flag). If no -N flag is specified and no host name is specified, the mkwpar command attempts to discover the IP address by performing the gethostbyname subroutine on the workload partition name (specified with the -n flag). If you can find an address on the same subnet as any global interface, use that interface settings with the resolved IP address to create the default network entry. You can specify the following attributes for the -N flag:</p> <ul style="list-style-type: none"> • interface= <i>if</i> or interface=<i>namemappedif</i> • address=<i>A.B.C.D</i> • netmask=<i>A.B.C.D</i> • broadcast=<i>A.B.C.D</i> • address6=<i>S:T:U:V:W:X:Y:Z</i> • prefixlen=<i>n</i> <p>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:</p> <pre style="background-color: #f0f0f0; padding: 5px;"># 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</pre> <p>To define an IPv6 network configuration, specify the -N flag with the address6, prefixlen, and interface attributes. 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 4 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.</p>
-o <i>outfile</i>	Indicates an output path and file name to which to write specification data. This specification file can then be used to create a workload partition later, by using the -f flag.
-O	This flag is used to force an existing volume group to be overwritten on a particular set of devices, specified with the -D rootvg=yes flag directive. If not specified, the overwrite value defaults to FALSE. This flag must be specified only once, as its setting is applied to all devices specified with the -D rootvg=yes flag directive.
-p [<i>name</i>]	<p>Indicates that the workload partition's file systems exist and must be preserved, which means the root part must not be populated. You must specify the existing file systems to the mkwpar command in one of the following two ways:</p> <ul style="list-style-type: none"> • Use the <i>name</i> parameter to specify an existing mount group in /etc/filesystems. Such a mount group usually exists because a previous workload partition was removed by using rmwpar -p. If you specify the <i>name</i> parameter, it cannot match the name of an existing workload partition on the system. If you specify the -d flag, the mount points of the file systems are adjusted accordingly. If you do not specify the -d flag, the base directory of the workload partition is determined based on the mount points that are associated with the discovered file systems. • If the file systems are not defined in /etc/filesystems, use the -p flag with the -M flag or mount stanzas in the specification file to define the attributes of the file systems. <p>If you specify the -p flag with the <i>name</i> parameter, and no workload partition name is provided by using other means (for example, the -n flag or general.name in the specification file), you can also use the <i>name</i> parameter as the workload partition name.</p> <p>Note: The -p flag is mutually exclusive with the -B flag.</p> <p>The -p flag can be used to re-create a versioned workload partition. The workload partition's file systems must be preserved. To re-create such a versioned workload partition, the -f infile flag is included to provide a specification file that is saved from the original workload partition. The specification file is required to preserve the ostype value of the workload partition.</p>
-P	Sets the root password for the workload partition. The mkwpar command prompts you for the password interactively.

Item	Description
-r	<p>Duplicates the network name resolution configuration from the global system. The following files, if they exist, are copied into the workload partition:</p> <ul style="list-style-type: none"> • /etc/resolv.conf • /etc/hosts • /etc/netshconf • /etc/irs.conf • /etc/networks
-R attribute=value	<p>If the NSORDER environment variable is defined in the calling environment, it is added to the workload partition's /etc/environment file.</p> <p>Allows specification of resource control attributes. You specify only one -R flag. Most resource controls are similar to those resource control attributes that are supported by Workload Manager. You can use the following attributes:</p> <p>active={yes no}</p> <ul style="list-style-type: none"> • Active=no means that the resource attributes are defined but the resource controls are not activated when the WPAR is started. • Active=yes means that the resource control attributes are activated when the WPAR starts. <p>Tip: If this field is set to 'no', performance metrics such as processor and memory usage are not available by using such commands as topas and wlmstat, either inside and outside of the workload partition.</p> <p>rset=rset</p> <p>Configures the workload partition to use a resource set that was created by the mkrsset command.</p> <p>shares_CPU=n</p> <p>Specifies the number of processor shares that are available to the workload partition.</p> <p>CPU=m%-SM%,HM%</p> <p>Specifies the percentage processor limits for the processes of the workload partition.</p> <p>shares_memory=n</p> <p>Specifies the number of memory shares that are available to the workload partition.</p> <p>memory=m%-SM%,HM%</p> <p>Specifies the percentage memory limits for the processes of the workload partition.</p> <p>procVirtMem=n[M MB G GB T TB]</p> <p>Specifies the maximum amount of virtual memory that a single process can consume. 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 that is allowed is 1 MB. The maximum limit that can be specified is 8796093022207M, 8589934591G, or 8388607T. If you set the value to -1 (no units), the limit is disabled. See Workload Manager limits File.</p> <p>totalVirtMem=n[M MB G GB T TB]</p> <p>The maximum amount of virtual memory that can be consumed by the WPAR as a whole. Processes that cause the specified limit to be exceeded are terminated. The valid range and units are the same as procVirtMem. If you set the value to -1 (no units), the limit is disabled. See Workload Manager limits File.</p> <p>totalProcesses=n</p> <p>Specifies the total number of processes that are allowed in the workload partition. See Workload Manager limits File.</p> <p>totalPTYs=n</p> <p>Specifies the total number of pseudo terminals that are allowed in the workload partition. See pty Special File.</p> <p>totalLargePages=n</p> <p>Specifies the number of large pages that can be allowed for the workload partition. See Large Pages.</p> <p>pct_msgIDs=n%</p> <p>Specifies the percentage of the maximum number of message queue IDs of the system that are allowed in the workload partition. See Message Queue Kernel Services.</p> <p>pct_semIDs=n%</p>

Item	Description
	Specifies the percentage of the maximum number of semaphore IDs of the system that are allowed in the workload partition.
	pct_shmIDs=n%
	Specifies the percentage of the maximum number of shared memory IDs of the system that are allowed in the workload partition. See Shared Memory .
	pct_pinMem=n%
	Specifies the percentage of the maximum pinned memory of the system that can be allocated to the workload partition. See Support for pinned memory .
	totalThreads=n
	Specifies the total number of threads that are allowed in the workload partition. See Workload Manager limits File .
rootvg=yes/no	Used to indicate whether the specified disk device is to be used as a rootvg WPAR device. If the rootvg option is not specified, the command takes the default of <i>no</i> .
-s	Starts the workload partition after it is created.
-S secfile = /path/to/secattr privs[+ -] = list	Configures the set of privileges that can be assigned to processes that are running in a system workload partition.
	<p>You can provide privileges in a specification file (see the -f flag), in a separate security attributes file through -S secfile=/path/to/secattr, or on the command line by using the -S privs=list flag. If you do not provide security attributes through one of these mechanisms, the /etc/wpars/secattr file is used by default. When you use a separate security attributes file (either the default file or the file that is supplied through -S secfile), this file is read once when the workload partition is created to determine the privileges that are associated with the workload partition. Subsequent changes to the file have no effect on existing workload partitions. The default security attributes file /etc/wpars/secattr must not be modified directly as it might be overwritten in the future.</p> <p>If you use a base list of privileges from a specification file or security attributes file (including the default), individual privileges can be added to or removed from the list by specifying -S privs+=list, -S privs-=list, or both. Separate attributes must be separated by a blank space and must be unique, which means secfile=, privs=, privs+=, and privs-= cannot be specified more than once. Privileges must be comma-separated (without spaces) and must be unique. Attributes are processed in the following order regardless of the order that is specified in either the command line or the specification file:</p> <ol style="list-style-type: none"> 1. The first attribute to be processed is the privs attribute without the + or - modifier. For example, privs=PV_AZ_READ,PV_AZ_ADMIN. If this attribute is found, no other attributes can be used. 2. The next attribute to be processed is the secfile attribute. See the security stanza of the Specification File Format for details on the format of this file. 3. If none of the attributes that are listed previously are specified, the /etc/wpars/secattr file is used to populate the list of privileges. 4. The next attribute to be processed is an attribute with the + modifier. For example, privs+=PV_DAC_UID,PV_AZ_ROOT. This command adds the specified privileges to the list of privileges that are specified in the security file. 5. The final attribute to be processed is an attribute with the - modifier. For example, privs-=PV_AZ_ROOT. This command removes the specified privileges from the list of privileges that are specified in the security file.
	Tip: If you specify the -S flag on the command line, any security attributes in the specification file are ignored.

Item	Description
-u <i>userscript</i>	<p>Specifies the path to a user script to be run by workload partition commands at various administration points. The parameter of the -u flag can be a string that is enclosed in quotation marks, including more arguments to be passed to the script. The first component of the parameter of the -u flag must be an absolute path to an existing executable file. The script is started in the following manner:</p> <pre>/path/to/userscript <action> <wparName></pre> <p>The first argument indicates the administrative action that is being performed, as follows:</p> <p>WPAR_LOAD A script runs in the global environment after the kernel is configured, and before the tracked process is created. If the script returns a value other than zero, the workload partition cannot be started.</p> <p>WPAR_START A script runs in the global environment as soon as the workload partition becomes active. For system workload partitions, the script runs after the device configuration is complete. For application workload partitions, the script runs as soon as the tracked process is started. In the latter case, 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 other than zero, a warning is logged, but no other behavior changes.</p> <p>WPAR_STOP A script runs in the global environment after all workload partition processes finish before the kernel is unconfigured. Note: This code path can be started by a dissociated process with its standard I/O streams closed or redirected to SRC logs. If the script returns a value other than zero, a warning is logged, but no other behavior changes</p> <p>The second argument is the name of the workload partition. The script can use the lswpar command to obtain any other necessary configuration data.</p>
-U [<i>Workload Partition UUID</i>]	Specifies the Workload Partition UUID. If you do not specify the value, the UUID is automatically generated for the corresponding Workload Partition.
-w	Writes the specification file only. Used with the -o flag, the -w flag causes the mkwpar command to quit after the new specification file is written, without actually creating the workload partition.
-W	Filters the bootset related information from the WPAR specification file. When the -W flag is used with the -e flag to generate a specification file from an existing WPAR, the resulting specification file does not include the bootset related attributes, such as bootset and bootlist.
-X [exportfile =/path/to/file [[kext =/path/to/extension/ALL]] [local =yes no] [major =yes no]	<p>Configures exporting kernel extensions that will be allowed to load inside a workload partition. You can specify more than one -X flag to allocate multiple kernel extensions. Separate the attribute=value by blank spaces. This flag is not valid for application workload partitions. You can specify the following attributes for the -X flag:</p> <p>exportfile=/path/to/file Specify a file containing valid extension stanzas that will be exported. An extension stanza should contain at least the kext attribute. The local and major attribute can also be specified in the stanza which are described below. The exportfile attribute is mutually exclusive with the kext attribute. It is also mutually exclusive with the local and major attribute because these can be specified for each extension stanza in the exportfile.</p> <p>This is a file that can be created by a user to use with exportfile=/path/to/file for mkwpar and chwp. It can contain multiple extension stanzas. The kext attribute is required for each extension stanza. The local and major are optional as they both have default value no. The exportfile will look similar to the following.</p> <pre>extension: major = "yes" local = "no" kext = "/usr/lib/drivers/ldterm"</pre> <p>kext=/path/to/extension Specify a kernel extension that will be exported. This is a kernel extension located in the global system's filesystem. The keyword ALL can also be specified. This will allow a workload partition to load any extension. When ALL is specified, the local and major attributes are restricted to local=yes and major=no. Additional -X flags can be specified to override the restricted local and major values. The kext attribute is mutually exclusive with the -X exportfile attribute.</p> <p>local=yes/no Specifying local=yes will make an instance of the kernel extension accessible to only the workload partition that is loading it. Specifying local=no will share the instance of the kernel extension loaded in the global system. By default, local=no.</p> <p>major=yes/no This attribute should only be used for kernel extensions that have an associated device major. By default, major=no.</p>

Item	Description
-t	Copies the file systems from the rootvg volume groups from a system backup image specified by the -B flag or the global system.
-T [preserve_private ={yes no}] [preserve_wpars ={yes no}]	Controls behavior when copying the file systems from a rootvg volume group or system backup.
preserve_private ={yes no}	Controls whether filesets that are designated as not visible within WPARs remain in a WPAR that is created by copying a rootvg volume group from a system backup or global system. The default value is no.
preserve_wpars ={yes no}	Indicates whether the file systems associated with WPARs are available in the source system in a WPAR, which is created by copying a rootvg volume group from a global system. The default value is no.

Security

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

Examples

1. To create a workload partition called `roy`, enter the following command:

```
mkwpar -n roy -N address=192.168.0.51
```

All values that are not specified are generated or discovered from the global system settings.

2. To create a workload partition based on an existing specification file, enter the following command:

```
mkwpar -f /tmp/wpar1.spec
```

3. To create a modified copy of a specification file with a new IP address, host name, and workload partition name (without creating a workload partition), enter the following command:

```
mkwpar -f /tmp/wpar1.spec -N address=219.168.45.132 -h www.flowers.com -n wpar2 -o /tmp/wpar2.spec -w
```

4. To create a specification file, which is based on an existing workload partition, enter the following command:

```
mkwpar -e wpar1 -o /tmp/wpar2.spec -w
```

5. To recreate a workload partition that was previously removed with the **rmwpar -p** command, enter the following command:

```
mkwpar -p wparname
```

6. To create a rootvg workload partition, enter the following command:

```
mkwpar -n test -D devname=hdisk1 rootvg=yes -0
```

7. To create a rootvg workload partition called `wpar1` with the storage device on an adapter, enter the following command (assuming that `hdisk3` is attached to the adapter, `fcs2`):

```
mkwpar -n wpar1 -D devname=fcs2 -D devname=hdisk3 rootvg=yes
```

8. To create a specification file from an existing workload partition, without including bootset related information, enter the following command:

```
mkwpar -e <existing wparname> -W -w -o <path to spec file>
```

9. To create a WPAR with a default route, enter the following command:

```
mkwpar -n wparB -N address=192.162.1.2 interface=en0 netmask=255.255.255.0 -i -I rtdst=0.0.0.0 rtgateway=192.162.1.1
```

10. To create a WPAR with its own routing table but no default route, enter the following command:

```
mkwpar -n wparA -N address=192.152.1.2 interface=en0 netmask=255.255.255.0 -i
```

Files

Item	Description
<code>/etc/wpars/devexports</code>	Default device export control file for workload partitions.
<code>/etc/wpars/secattrs</code>	Default security file for workload partitions.
<code>/usr/samples/wpars/sample.spec</code>	An annotated workload partition specification file.

mkwpardata Command

Purpose

Creates a file containing information about a workload partition for use by the **savewpar** and **restwpar** commands.

Syntax

```
mkwpardata [ -X ] [ -m ] WparName
```

Description

The **mkwpardata** command creates a file containing information about a workload partition (WPAR) for use by the **savewpar** and **restwpar** commands. The information includes the list of logical volumes, file systems and their sizes, the list of volume groups, and the WPAR name. The following files are created: You can edit the information in the file before issuing the **savewpar** command.

Flags

Item	Description
-m	Creates map files that specify the mapping of the logical-to-physical partitions for each logical volume in the WPAR. This mapping can be used to allocate the same logical-to-physical mapping when the image is restored. The map file locations are stored in the MAPFILE field in the image.data file for each logical volume. For example, for the hd7 logical volume, the location of the map file is /tmp/wpardata/WparName/hd7.map . The MAPFILE field in the image.data file for the hd7 logical volume is under the entry MAPFILE=/tmp/wpardata/WparName/hd7.map . The map files in the backup image are copied after the image.data file.
-X	Expands the /tmp file system if needed.

Parameters

Item	Description
<i>WparName</i>	Specifies the name of workload partition to back up.

Files

Item	Description
<code>/tmp/wpardata/WparName/image.data</code>	Created for general and storage information about a WPAR. The <i>WparName</i> 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 reinstall the WPAR. The mkwpardata command overwrites this file if it already exists. The image.data file is located in the <code>/tmp/wpardata/WparName</code> directory, where <i>WparName</i> is the workload partition name.

mm Command

Purpose

Prints documents formatted with memorandum macros.

Syntax

```
mm [ -M Media ] [ -c ] [ -e ] [ -E ] [ -t ] [ -12 ] [ -TName ] { File ... | - }
```

Description

The **mm** command formats documents that use the **nroff** command and the **mm** macro package. The **mm** command has flags that specify preprocessing by the **tbl** and **neqn** commands and postprocessing by various terminal-oriented output filters. The proper pipelines and the required flags for the **nroff** command are generated depending on the flags that are selected.

Notes:

1. Use the **-oList** flag of the **nroff** command to specify ranges of output pages. Remember that if the **mm** command is called with the **-e**, **-t**, or **-** (minus sign) flags together with the **-oList** flag, and if the last page of the document is not specified by the *List* variable, you may receive a `broken pipe` message. This message is not an indication of any problem and can be ignored.
2. The **mm** command calls the **nroff** command with the **-h** flag. With this flag, the **nroff** command assumes that the workstation has tabs set every 8 character positions.
3. If you use the **-s** flag of the **nroff** command (to stop between pages of output), use a linefeed (rather than the Enter key or a newline character) to restart the output. The **-s** flag of the **nroff** command does not work with the **-c** flag of the **mm** command or if the **mm** command automatically calls the **col** command.
4. Providing inaccurate information to the **mm** command about the kind of workstation its output is to be printed on will produce unsatisfactory results. However, if you are redirecting output to a file, use the **-T37** flag. Then, use the appropriate workstation filter when you print the file.

To obtain a list of **mm** command flags, enter the command name with no parameters. The flags can occur in any order, but they must come before the *File* parameter. Any other flags (for instance, **-rANumber**) are passed to the **nroff** command.

Flags

Item	Description
-M <i>Media</i>	<p>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:</p> <p>A4 Specifies a paper size of 8.27 X 11.69 inches (210 X 297 mm).</p> <p>B5 Specifies a paper size of 6.93 X 9.84 inches (176 X 250 mm).</p> <p>EXEC Specifies a paper size of 7.25 X 10.5 inches (184.2 X 266.7 mm).</p> <p>LEGAL Specifies a paper size of 8.5 X 14 inches (215.9 X 355.6 mm).</p> <p>LETTER Specifies a paper size of 8.5 X 11 inches (215.9 X 279.4 mm). This is the default value.</p> <p>Note: The <i>Media</i> variable is not case sensitive.</p>
-c	<p>Calls the col command. Note that the col command is called automatically by the mm command for the following terminal names. The following devices can be specified by the -TName flag, the \$TERM shell variable, or by using the default:</p> <ul style="list-style-type: none">• ppds• lp• 2631• 8510
-e	<p>Calls the neqn command; also causes the neqn command to read the /usr/share/lib/pub/eqnchar file. See the eqnchar file format.</p>
-E	<p>Calls the -e flag of the nroff command.</p>
-t	<p>Calls the tbl command.</p>
-12	<p>Uses 12-pitch font. Use this when the \$TERM shell variable is set to 300, 300s, 450, or 1620. (The pitch switch on the DASI 300 and 300s workstations must be manually set to 12 if this flag is used.)</p>
-TName	<p>Uses the workstation type specified by the <i>Name</i> variable.</p> <p>By default, the mm command uses the value of the \$TERM shell variable from the environment as the value of the <i>Name</i> variable. If the \$TERM shell variable is not set, the mm command uses lp (the generic name for printers that can underline and tab). If several workstation types are specified, the last one listed applies.</p>
-	<p>Forces input to be read from standard input.</p>

Parameters

Item	Description
<i>File</i>	Specifies the file that the mm command formats.

Examples

1. When the **\$TERM** shell variable is set in the environment to the **hplj** command, the following two command lines are equivalent:

```
mm -t -rC3 File
tbl File | nroff -mm -Thplj -h -rC3
```

2. The **mm** command reads the standard input when you specify a - (minus sign) flag instead of a value for the *File* variable. This option allows you to use the **mm** command as a filter, as follows:

```
cat File | mm -
```

Note: Using other files together with a - (minus sign) flag leads to undesired results.

Environment Variables

Item	Description
------	-------------

\$TERM	Specifies the terminal names.
---------------	-------------------------------

Files

Item	Description
------	-------------

<code>/usr/share/lib/pub/eqnchar</code>	Contains special character definitions for the eqn command and the neqn command.
---	--

mmt Command

Purpose

Typesets documents.

Syntax

```
mmt [ -M Media ] [ -a ] [ -c ] [ -e ] [ -g ] [ -p ] [ -t ] [ -z ] [ -TName | -DDestination ] [ File | - ]
```

Description

Similar to the **mm** command, the **mmt** command typesets its input using the **troff** command, rather than formatting it with the **nroff** command. The **mmt** command uses the **mm** macro package. There are flags to specify preprocessing by the **tbl**, **pic**, **eqn**, and **grap** commands. The proper pipelines, required parameters, and flags for the **troff** command and the **mm** macro package are generated, depending on the flags selected.

There are several flags that are specific to the **mmt** command. Any other parameters or flags (for instance, **-rANumber** or **-a**) that you give the **mmt** command are passed to the **troff** command. You can put flags in any order, but they must be listed before any input files. *File* specifies the file that the **mmt** command formats. If you do not give *File* parameters or other flag variables, the **mmt** command prints a list of its flags.

The **mmt** command, unlike the **troff** command, automatically pipes its output to a postprocessor, unless specifically requested not to do so. The user should not specify a postprocessor when using the **mmt** command. The precedence is as follows:

1. The **-z** flag; no postprocessor is used.
2. The **-TName** flag.
3. The **TYPESETTER** environment variable is read.
4. The default is set to **ibm3816**.

The **mmt** command reads standard input when you specify a - (minus sign) instead of any *File* parameters.

Use the **-oList** flag of the **troff** command to specify ranges of pages to be output.

Note: If you call the **mmt** command with one or more of the **-e**, **-c**, **-t**, **-p**, **-g**, and - (minus sign) flags together with the **-oList** flag of the **troff** command, you may receive a broken pipe message if the last page of the document is not specified by the *List* variable. This broken pipe message is not an indication of any problem and can be ignored.

Flags

Item	Description
-M Media	<p>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:</p> <p>A4 Specifies a paper size of 8.27 X 11.69 inches (210 X 297 mm).</p> <p>A5 Specifies a paper size of 5.83 X 8.27 inches (148 X 210 mm).</p> <p>B5 Specifies a paper size of 6.93 X 9.84 inches (176 X 250 mm).</p> <p>EXEC Specifies a paper size of 7.25 X 10.5 inches (184.2 X 266.7 mm).</p> <p>LEGAL Specifies a paper size of 8.5 X 14 inches (215.9 X 355.6 mm).</p> <p>LETTER Specifies a paper size of 8.5 X 11 inches (215.9 X 279.4 mm). This is the default value.</p> <p>Note: The <i>Media</i> variable is not case sensitive.</p>
-a	Displays readable troff output to the terminal.
-c	Preprocesses the input files with the cw command.
-e	Calls the eqn command; also causes the eqn command to read the /usr/share/lib/pub/eqnchar file (see the eqnchar file format).
-g	Calls the grap command, which in turn calls the pic command.
-p	Calls the pic command.
-t	Calls the tbl command.
-z	Starts no output filter to process or redirect the output of the troff command.
-D<i>Destination</i>	Directs the output to a device specified by the <i>Destination</i> variable. Supported destination devices for English-language output is 4014, which is the Tektronix 4014 terminal by way of the tc command.

Item	Description
-TName	Creates output for a troff device as specified by the <i>Name</i> variable. The output is sent through the appropriate postprocessor.. The default value is ibm3816 . Possible Name variables are: ibm3812 3812 Pageprinter II. ibm3816 3816 Pageprinter. hplj Hewlett-Packard LaserJet II. ibm5587G 5587-G01 Kanji Printer multi-byte language support. psc PostScript printer. X100 AIXwindows display.
-	Forces input to be read from standard input.

mmtu Command

Purpose

Displaying, adding, and deleting maximum transfer unit (MTU) values used for path MTU discovery.

Syntax

```
mmtu { -a Value | -d Value | -s }
```

Description

Use the **mmtu** command to display, add, and delete maximum transfer unit (MTU) values to the list of potential path MTU values. Path MTU discovery uses the list of potential path MTU values to detect the path MTU. The list of potential path MTU values is only used when there are routers in the path that do not comply with RFC 1191. The user must have administrative authority to add or delete MTU values.

Note: The **-a** and **-d** flags used to modify the list of potential path MTU values are disallowed when executed within workload partitions.

Flags

Item	Description
-a Value	Adds the new MTU to the list of potential path MTU values.
-d Value	Deletes the value from the list of potential path MTU values.
-s	Displays the current list of potential path MTU values.

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 add a value to the list of potential path MTU values, enter:

```
mmtu -a mtu-value
```

2. To delete a value from the list of potential path MTU values, enter:

```
mmtu -d mtu-value
```

3. To display the contents of the list of potential path MTU values, enter:

```
mmtu -s
```

Files

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

mobip6ctrl Command

Purpose

Configures and manages the mobile IPv6 home agent and correspondent node functionality.

Syntax

```
mobip6ctrl [ -c ] [ -R ] [ -b ] [ -S { 0 | 1 } ] [ -n { 0 | 1 } ] [ -l LifeTime ] [ -a | -d HomeAddress CareOfAddress MyAddress ]
```

Description

The **mobip6ctrl** command is used to configure and manage the mobile IPv6 home agent and correspondent node. It can enable and disable NDP proxy and IP security checking, and it can be used to display or modify the mobile IPv6 binding cache.

NDP proxy must be enabled if the system is configured as a home agent. This allows the home agent to intercept packets addressed to mobile nodes that are not currently on their home network.

IP security checking enables checking to ensure that IP security is used for the Binding Update and Binding Acknowledgement messages sent for mobile IPv6. Because these two types of messages have the ability to affect the routing of packets addressed to a mobile node, they would represent a significant security vulnerability if not protected by IP security. If checking is enabled, the mobile IPv6 home agent or correspondent node will discard any Binding Update or Binding Acknowledgement packets that are not protected by IP security.

The mobile IPv6 binding cache on a home agent or correspondent node maps home addresses to the current care-of addresses for each mobile node. This allows the home agent to tunnel traffic to the mobile node at its current location, and allows a correspondent node to send packets directly to a mobile node at its current location. The **mobip6ctrl** command can be used to view the binding cache or manually edit it for debugging purposes.

Normally, this command is used from the `/etc/rc.mobip6` script when mobile IPv6 has been configured using system management.

Flags

Item	Description
-a <i>HomeAddress CareOfAddress MyAddress</i>	Adds this entry to the binding cache.
-b	Displays all binding cache entries.
-c	Compatibility option which enables the support of the mobiles implementing the draft #13 of the <i>Mobility support in IPv6</i> specification. Using this option, the home agent or correspondent node will accept the binding update messages sent using a Destination Option and using an Authentication Header (AH) to protect these packets with IPsec.
-d <i>HomeAddress CareOfAddress MyAddress</i>	Delete this entry from the binding cache.
-l <i>LifeTime</i>	Specifies the default life time value for binding cache entries in seconds.
-n 0 1	Activates or deactivates NDP proxy capabilities. A value of 1 activates the NDP proxy capabilities, and a value of 0 disables NDP proxy capabilities. The default value is 0.
-R	Resets all the binding cache entries.
-S 0 1	Enables or disables checking to ensure that IP security is used for all Binding Update and Binding Acknowledgement packets. A value of 1 enables checking, and a value of 0 disables checking. The default value is 0.

Exit Status

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

Security

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

Examples

1. The following example enables NDP proxy and IP security checking for mobile IPv6:

```
mobipctrl -S 1 -n 1
```

2. The following example displays all entries in the binding cache:

```
mobip6ctrl -b
```

The output from this command looks similar to the following:

```
                BINDING CACHE LIST (1 elem)
Home Address.....: 3ffe:300:20:1102::217
Care-Of Address....: 3ffe:300:20:1101::217
My Address.....: 3ffe:300:20:1102::223
Life time.....: 518
Time since last usage: 50
Rate limit time.....: 0
```

```
Retransmit count.....: 0
Sequence number.....: 14
Registered by me.....: 1
Prefix length.....: 64
```

mobip6reqd Daemon

Purpose

Provides the Mobile IPv6 home agent daemon.

Syntax

To run the daemon using the System Resource Controller:

```
startsrc -s mobip6reqd
```

To run the daemon without using the System Resource Controller:

```
mobip6reqd
```

Description

The **mobip6reqd** daemon must be running in order for the system to function as a mobile IPv6 home agent. This daemon enables the home agent to perform NDP proxying for mobile nodes. The daemon is normally started automatically by the **/etc/rc.mobip6** script if the mobile IPv6 home agent has been enabled using system management.

Exit Status

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

Security

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

Examples

1. The following example starts the **mobip6reqd** daemon:

```
startsrc -s mobip6reqd
```

2. The following example stops the **mobip6reqd** daemon:

```
stopsrc -s mobip6reqd
```

monacct Command

Purpose

Performs monthly or periodic accounting.

Syntax

```
/usr/sbin/acct/monacct [ -X ] [ Number ]
```

Description

The **monacct** command performs monthly or periodic accounting. The intervals are set in the **crontab** file. You can set the **cron** daemon to run the **monacct** command once each month or at some other specified time period. The **monacct** example shows how to set up this command for use with the **cron** daemon. See the **crontab** command for more information about setting up **cron** files.

The *Number* parameter indicates which month or other accounting period to process. The default value of the *Number* parameter is the current month. The **monacct** command creates summary files in the **/var/adm/acct/fiscal** file and restarts summary files in the **/var/adm/acct/sum** file, the cumulative summary to which daily reports are appended.

Note: You should not share accounting files among nodes in a distributed environment. Each node should have its own copy of the various accounting files.

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 monacct command to use the /var/adm/acct/sumx and /var/adm/acct/fiscalx directories instead of the /var/adm/acct/sum and /var/adm/acct/fiscal directories.

Security

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

Example

To produce automatically a monthly accounting report, add the following to the **/var/spool/cron/crontabs/root** file:

```
15 5 1 * * /usr/sbin/acct/monacct
```

This example shows the instructions that the **cron** daemon will read and act upon. The **monacct** command will run at 5:15 (15 5) the first day of each month (1). This command is only one of the accounting instructions normally given to the **cron** daemon.

Files

Item	Description
/usr/sbin/acct	Contains the accounting commands.
/var/adm/acct/fiscal	Contains accounting data files.
/var/adm/acct/sum	Cumulative daily accounting records.
/var/spool/cron/crontabs	Contains the commands to be run by the cron daemon at regularly scheduled intervals.

mon-cxma Command

Purpose

Monitor status of 128-port asynchronous subsystem and attached devices.

Syntax

To Display All 128-Port Adapters:

mon-cxma

To Display Syntax or Slots and Bus Information:

mon-cxma { **-h** | **-x** }

To Display Specific Slots and Bus Information:

mon-cxma { [**-l** [*LogFile*]] [**-f** [*DeviceFile*]] [**-s** [*SlotNumber*]] [**-b** [*BusNumber*]] }

Description

The **mon-cxma** command is a software tool which provides a means to monitor the status of serial devices and remote async nodes (RAN) attached to the IBM 128-port asynchronous adapter. It is used for subsystem problem determination and can be accessed locally and remotely via modem. The only restriction on modem access is that the modem can not be physically attached to the 128-port adapter being monitored.

When the user enters the **mon-cxma** command at the command line, it automatically detects and displays all available 128-port adapters in the system. The bus and slot location within the system is displayed for each adapter and the user can select adapter to monitor.

You can use the System Management Interface Tool (SMIT) **smit 128psync** fast path to advance directly to the "128-Port Asynchronous Adapter" menu. When run from SMIT, the **mon-cxma** command automatically displays all available 128-port adapters in the system.

Flags

Item	Description
-b [<i>BusNumber</i>]	Specifies the bus number of the device. Valid values for <i>BusNumber</i> are 0 to (n-1), where n is the number of buses the system has.
-f [<i>DeviceFile</i>]	Specifies the device special file. Use this file to look at a specific device driver without having to make a selection. The default device special file is /dev/cxma0 .
-h	Shows syntax information.
-l [<i>LogFile</i>]	(Lowercase L) Specifies the file to be used as the log. Use this file to store information from the screen when the IMAGE key is pressed. The default log file is /tmp/mon-cxma.log .
-s [<i>SlotNumber</i>]	Specifies the slot number of the device. Valid values for <i>SlotNumber</i> are 0 to (n-1), where n is the number of slots the system has.
-x	Shows the POS (Programmable Select Option) register values for all the slots and buses.

Note: **-x** and **-h** ignore other options.

Security

Access Control: Root authority required to run this command.

Auditing Events: N/A

Examples

1. To run the `mon-cxma` command using the SMITfastpath, enter:

```
smit 128psync
```

2. To display all 128-port adapters, enter:

```
/usr/sbin/tty/mon-cxma
```

Files

Item	Description
<code>/usr/sbin/tty/mon-cxma</code>	Contains the mon-cxma command.
<code>/tmp/mon-cxma.log</code>	Contains the log file.

monitord Daemon

Purpose

Communicates with the License Use Management server and requests a concurrent-use license for each countable login.

Syntax

```
monitord [ -t Minutes ] [ -v Version.Release ]
```

Description

The operating system has multiple ways to access the system, and each of them has a different behavior upon exit. The **monitord** daemon provides a common interface to the License Use Management **netlsd**. **monitord** communicates with the License Use Management server and requests a concurrent-use license for each countable login.

Note: The License Use Management licensing mechanism is used only if the system has the *floating license mode* enabled.

After user logout, **monitord** requests **netlsd** to release the specific license the user was using, in order to make it available for further logins.

monitord is started when the **chlicense -f on** command is used to enable the *floating license mode*. When the *floating license mode* is enabled, **monitord** is started upon system startup via an entry in **/etc/inittab**. The default (invoked without **-t** option) is an interval of fifteen minutes.

The entry in **/etc/inittab** looks like the following:

```
monitord:2:once:/usr/sbin/monitord >/dev/console 2>&1
```

Flags

Item	Description
-t <i>Minutes</i>	Sets the value in minutes of the heartbeat interval. A value of 0 sets an infinite interval. The default is fifteen minutes.
-v <i>Version.Release</i>	Enables the <i>floating license mode</i> for a license of the specified <i>Version</i> and <i>Release</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.

moo Command

Purpose

Starts the number-guessing game.

Syntax

moo

Description

The **moo** command picks a combination of four random, non-repeating numbers. You guess four numbers at the your `guess?` prompt. Each correct number in an incorrect position in the four number combination scores "cow." Each correct number in the correct position in the four number combination scores a "bull." For example:

```
your guess?
1470
bulls = 0 cows = 1
your guess?
```

In this example, one of the four numbers (1, 4, 7, and 0) is correct but in the incorrect position. None of the numbers are correct and in the correct position.

To quit the game, press the Interrupt (Ctrl-C) or End Of File (Ctrl-D) key sequence.

File

Item	Description
<code>/usr/games</code>	Contains the system's games.

more Command

Purpose

Displays file contents one screen at a time.

Syntax

```
more [ -c ] [ -d ] [ -e ] [ -H ] [ -i ] [ -l ] [ -N ] [ -s ] [ -u ] [ -v ] [ -z ] [ -n Number ] [ -p Subcommand ] [ -t Tagstring ] [ -W Option ] [ -x Tabs ] [ File ... ]
```

Description

The **more** command reads files and displays the text one screen at a time. The command pauses after each screen and prints the word `More` at the bottom of the screen. If you then press a carriage return,

the **more** command displays an additional line. If you press the space bar, the **more** command displays another full screen of text.

Note: On some terminal models, the **more** command clears the screen, instead of scrolling.

Instead of naming files to read, you can either redirect or pipe standard output, such as a long directory listing, to the **more** command. The command adds a % (percent sign) to its prompt when reading from a file rather than a pipe. This provides the percentage of the file (in characters, not lines) that the **more** command has read.

The **more** command sets the terminal to NOECHO mode so the output can be continuous. With the exception of the / and ! subcommands, commands that are typed do not normally show up on the terminal. If the standard output is not a terminal, the **more** command will act just like the **cat** command, except that a header will be printed before each file in a series.

Environment Variables

Environment variables affect the way the **more** command works. You can set some environment characteristics in the **/etc/environment** file and system profile files, such as the **.ksh**, **.csh**, and **.profile** files.

The **more** command uses the **TERM** variable to determine terminal characteristics. If this variable is NULL or not set, the command uses the default terminal type. The **/usr/share/lib/terminfo** directory contains definitions for terminal characteristics.

By default, the **more** command window size is 2 lines less than what the system terminal is capable of. The command sets the default window size based on the **LINES** variable. Also, you can easily adjust the window size for each run of the command by adding the **-n** flag.

Use the **MORE** variable to customize the **more** command with your preferred configuration each time the system starts. This variable accepts **more** command flags.

Flags

Flags Description

Item	Description
-c	Prevents the screen from scrolling, which makes text easier to read as the more command writes to the screen. The system ignores the -c flag if the terminal cannot clear to the end of a line.
-d	Prints a message, appended to the More prompt at the bottom of the screen, about which keys continue, quit, and provide help for the more command. Displays error messages rather than ringing the terminal bell if an unrecognized command is used. This is helpful for inexperienced users.
-e	Exits automatically after displaying the last line of the last file.
-H	Disables the search pattern highlighting feature by default.
-i	Searches for patterns without considering uppercase and lowercase.
-l	Pauses after detecting a page break in the input. If the -l flag is not used, the more command pauses to accept commands after any line containing a ^L (CTRL-L) character. Also, if a file begins with a FORMFEED, the screen is cleared before the file is printed.
-N	Suppresses line numbering. The default display, with line numbers, can slow the more command's performance on very large input files. The line numbering feature displays the line number in the = subcommand and passes the line number to the editor (if it is the vi editor).

Item	Description
-n <i>Number</i>	Configures the more command to display the specified number of lines in the window. Without the -n flag, the more command defaults to two lines less than what the terminal is capable of. For example, on a 24-line terminal, the default is 22 lines. The -n option overrides any values obtained from the environment.
-p <i>Subcommand</i>	<p>Starts the more command and specified subcommand for each <i>File</i> operand. For example, more -p 50j text1 text2 displays the text1 file at the fiftieth line; then does the same for the text2 file when you finish the first. See "Subcommands" for descriptions of more subcommands.</p> <p>If the command is a positioning command, such as a line number or a regular expression search, set the current position to represent the final results of the command, without writing any intermediate lines of the file. For example, the two commands:</p> <pre>more -p 1000j filename more -p 1000G filename</pre> <p>are functionally the same and will start the display with the current position at line 1000, passing the lines that j would write and would scroll off the screen if it had been issued during the file examination.</p> <p>If the positioning command is unsuccessful, the first line in the file will be the current position.</p>
-s	Reduces multiple blank lines in the output to only one blank line. The -s flag is particularly helpful in viewing output from the nroff command.
-t <i>Tagstring</i>	Displays the portion of the file that contains the specified tag. This flag works only on files containing tags created with the ctags command.
-u	Prevents the more command from treating a backspace character as a printable control character (displayed as a ^H (CTRL-H)), suppressing backspacing, underlining, or creating reverse video text for underlined information in a source file. The -u flag also forces the more command to recognize a carriage-return character, if it exists, at the end of a line.
-v	Suppresses the graphical translation of nonprinting characters. Without the -v flag, the more command graphically interprets all non-ASCII and most control characters, except Tab, Backspace, and Return. For example, if you do not use the -v flag, the more command displays the non-ASCII characters Ctrl-x as ^X and x as M-x.

Item	Description
-W <i>Option</i>	Provides the specified <i>Option</i> to the more command as an extension: notite Prevents the more command from sending the terminal initialization string (either the ti termcap or the smcup terminfo capability) before displaying the file. This option also prevents the more command from sending the terminal de-initialization string (either the te termcap or the rmcup terminfo capability) before exiting. tite Causes the more command to send the initialization and de-initialization strings. This is the default. These options control whether the more command sends the initialization strings described, which, for certain terminals (such as some xterms), cause the more command to switch to an alternative screen. The effect of switching screens is to erase the display of the file you were viewing.
-x <i>Tabs</i>	Sets tab stops at the specified <i>Tabs</i> position. The default tab setting is 8 columns.
-z	Graphically displays the Tab, Backspace, and Return control characters. With the -z flag, the more command translates the Backspace character as ^H, Return as ^M, and Tab as ^I.

Subcommands

The **more** command accepts subcommands when the command pauses and as parameters for the **-p** flag. Many subcommands take an optional integer, symbolized here by *K*, which you must enter before the subcommand, with no space between. The **more** command, in the paused state, processes subcommands immediately and does not require you to press the Enter key.

The **more** command uses the following subcommands:

more command

Item	Description
h	Displays a help screen that describes the more subcommands.
v	Starts the vi editor, editing the current file in the current line.
r or ^L	Refreshes the display.
R	Refreshes the display and removes buffered input.
[<i>K</i>](Spacebar)	Moves forward <i>K</i> lines when you press the spacebar. If you do not give a value for <i>K</i> , pressing the spacebar displays the next full screen by default. This spacebar subcommand is the same as [<i>K</i>] f or [<i>K</i>] ^F or [<i>K</i>] z .
[<i>K</i>] f or [<i>K</i>] ^F or [<i>K</i>] z	Moves forward <i>K</i> lines, or one full screen if you do not give a value for <i>K</i> .
[<i>K</i>] b or [<i>K</i>] ^B	Moves backward <i>K</i> lines, or one full screen if you do not give a value for <i>K</i> .
[<i>K</i>] d or [<i>K</i>] ^D	Moves forward <i>K</i> lines, or half a screen if you do not give a value for <i>K</i> . If you give a value for <i>K</i> , the more command sets the d and u scroll size to <i>K</i> lines for the session.

Item	Description
[K]u or [K]^U	Moves backward <i>K</i> lines, or half a screen if you do not give a value for <i>K</i> . If you give a value for <i>K</i> , the more command sets the d and u scroll size to <i>K</i> lines for the session.
[K]j or [K](Enter) or [K]^E	Moves forward <i>K</i> lines, or one line if you do not give a value for <i>K</i> .
[K]k or [K]^Y	Moves backward <i>K</i> lines, or one line if you do not give a value for <i>K</i> .
[K]g	Moves to the beginning of the file, unless you give a line number for <i>K</i> . The default for <i>K</i> is line number 1.
[K]G	Moves to the last line in the file, unless you give a line number for <i>K</i> . The default for <i>K</i> is the last line in the file.
[K]p or [K]%	Moves to the point in the file that is <i>K</i> percent of the total file. The default for <i>K</i> is one percent, or the first line in the file.
ma-z	Marks the current position in the file with the specified letter.
'a-z	(Single quote) Moves to the position marked with the specified letter.
"	(Two single quotes) Returns to the position from which the last large movement (moving more than a page) command was run. If no such movements have been made, returns to the beginning of the file.
[K]/pattern	(Slash) Searches forward, from the current position, for the specified occurrence of the specified pattern of characters. The default value for <i>K</i> is the first occurrence.
[K]/!pattern	(Slash, exclamation mark) Searches forward, from the current position, for the specified occurrence of a line that does not contain the specified pattern of characters. The default value for <i>K</i> is the first occurrence.
[K]?pattern	(Question mark) Searches backward, from the current position, for the specified occurrence of the specified pattern of characters. The default value for <i>K</i> is the first occurrence.
[K]?!pattern	(Question mark, exclamation mark) Searches backward, from the current position, for the specified occurrence of a line that does not contain the specified pattern of characters. The default value for <i>K</i> is the first occurrence.
[K]n	Repeats the last search, specifying an occurrence of the pattern (or an occurrence <i>not</i> containing the pattern if the search subcommand included !). The default value for <i>K</i> is the first occurrence.
:a	Lists the file or files you named in the more command line.

more command (continued)

Item	Description
:f or ^G or =	Displays information about the current file: <ul style="list-style-type: none">• file name• order of the file in the list of files• current line number• current position in the file, given as a percentage• current byte number and total bytes to display.
:e [File] or E [File]	Examines the specified file, provided you named it in the more command line.
[K]:n or [K]N	Examines either the next file (if you do not give a value for <i>K</i>) or the file <i>K</i> number of positions forward in the list of files you named in the more command line.
[K]:p or [K]P	Examines either the previous file (if you do not give a value for <i>K</i>) or the file <i>K</i> number of positions backward in the list of files you named in the more command line.
:t <i>Tagstring</i>	Displays the portion of the file that contains the specified tag. This subcommand works only on files containing tags created with the ctags command. The :t subcommand is the interactive version of the -t flag.
:q or q or Q	Exits the more command.
!:command or !command	Starts the specified command in a new shell.
H	Toggles the search pattern highlighting feature on or off.

Exit Status

This command returns the following exit values:

Exit Status

Ite	Description
m	
0	Successful completion.
>0	An error occurred.

Examples

1. To view a file named `myfile`, enter:

```
more myfile
```

2. To view output from the **nroff** command, enter:

```
ls -l | more
```

3. To view each file starting at its last screen, enter:

```
more -p G file1 file2
```

4. To view each file with the 100th line at the current position, enter:

```
more -p 100 file1 file2
```

Typically, the current position in a **more** command display is the third line on the screen. In this example, the first line on the screen is the 98th line in the file.

5. To view each file starting with the first line that contains the foo string, enter:

```
more -p /foo file1 file2
```

The **more** command displays the line in the current position, the third line on the screen.

Files

Files

Item	Description
<code>/usr/share/lib/terminfo</code>	Indicates the terminal information database.

mosy Command

Purpose

Converts the ASN.1 definitions of Structure and Identification of Management Information (SMI) and Management Information Base (MIB) modules into objects definition files for the **snmpinfo** command.

Syntax

```
mosy -o output_defs_file [ -s ] inputfile ...
```

```
mosy -x output_desc_file [ -o output_defs_file ] [ -s ] inputfile ...
```

```
mosy -c output_c_file [ -x output_desc_file ] [ -o output_defs_file ] [ -s ] inputfile ...
```

Description

The **mosy** command reads in the ASN.1 definitions of SMI and MIB modules and produces objects definition files in specific formats. The resulting objects definition files are used by the **snmpinfo** command.

The *inputfile* parameter files are required to be in the **smi.my** or **mibII.my** format. Sample files are the `/usr/samples/snmpd/smi.my` and `/usr/samples/snmpd/mibII.my` files. See the **smi.my** and the **mibII.my** files for information on the required format of the file specified by the *inputfile* parameter.

The **mosy -o** command is used to create the objects definition file specified by the *output_defs_file* parameter for the **snmpinfo** command. This file is normally the `/etc/mib.defs` file.

The objects definition file can be created with one pass of the **mosy** compiler if the **smi.my** and **mibII.my** files are both specified as *inputfile* parameters. The **smi.my** file must precede the **mibII.my** file on the command line.

The **mosy -o** command can also be used to create subfiles. If subfiles are created separately from the SMI and MIB modules, you must concatenate the various subfiles before the **snmpinfo** command can successfully use the resultant **mib.defs** file. The SMI subfile must be at the top of the final objects definition file.

You can add objects definitions for experimental MIB modules or private-enterprise-specific MIB modules to the `/etc/mib.defs` file, but you must first obtain the private MIB module from the vendor that supports those MIB variables.

To update the **/etc/mib.defs** file to incorporate a vendor's private or experimental MIB objects definitions, create a subfile and then concatenate that subfile to the existing MIB II **/etc/mib.defs** file. See example 3.

Flags

Item	Description
-c <i>output_c_file</i>	Creates a C code file.
-o <i>output_defs_file</i>	Defines the path and file name of the MIB objects definition file for the snmpinfo command. There is no default path and file name for this flag. If this flag is not specified, the objects definition file is not created.
-s	Suppresses the conversion verification messages. If this flag is not specified, the conversion verification messages are printed to standard output.
-x <i>output_desc_file</i>	Creates a description file in mib.desc file format.

Parameters

Item	Description
<i>inputfile</i>	Defines the ASN.1 object definitions module for input to the mosy compiler. This file can be formatted according to either the smi.my or mibll.my file format.

Examples

1. To create an objects definition file for use by the **snmpinfo** command with one pass of the **mosy** command, enter:

```
mosy -o /etc/mib.defs /usr/samples/snmpd/smi.my
      /usr/samples/snmpd/mibII.my
```

In this example, **/usr/samples/snmpd/smi.my** and **/usr/samples/snmpd/mibII.my** are both specified as input files and the resultant objects definition file is the **/etc/mib.defs** file.

2. To create objects definition subfiles, enter:

```
mosy -o /tmp/smi.obj /usr/samples/snmpd/smi.my
mosy -o /tmp/mibII.obj /usr/samples/snmpd/mibII.my
cat /tmp/smi.obj /tmp/mibII.obj > /etc/mib.defs
```

In this example, the first command creates an SMI objects file, **/tmp/smi.obj**, from the **/usr/samples/snmpd/smi.my** file. The second command creates the MIB objects definition file, **/tmp/mibII.obj**, from the **/usr/samples/snmpd/mibII.my** file. The final command concatenates the subfiles, placing the SMI objects definition file first in the resultant **/etc/mib.defs** file.

3. To add private enterprise specific MIB objects definitions to an existing **/etc/mib.defs** file for use by the **snmpinfo** command, enter:

```
mosy -o /tmp/private.obj /tmp/private.my
cat /etc/mib.defs /tmp/private.obj > /tmp/mib.defs
mv /tmp/mib.defs /etc/mib.defs
```

In this example, the first command creates the **/tmp/private.obj** objects definition file. The second command concatenates the **/etc/mib.defs** MIB objects definition file with the **/tmp/private.obj** private MIB file and places the concatenated contents into the **/tmp/mib.defs** temporary MIB objects definition file. The final command moves the temporary file to the **/etc/mib.defs** file for use by the **snmpinfo** command.

4. To create a description file in **/tmp/smi.desc**, a C code file named **/tmp/smi.c**, and a Management Information Base (MIB) definition file named **/tmp/smi.defs**, enter:

```
mosy -x /tmp/smi.desc -c /tmp/smi.c -o /tmp/smi.defs -s smi.my mibII.my
```

Files

Item	Description
/etc/mib.defs	Defines the Management Information Base (MIB) variables the SNMP agent should recognize and handle. The format of the /etc/mib.defs file is required by the snmpinfo command.
/usr/samples/snmpd/smi.my	Defines the ASN.1 definitions by which the SMI is defined as in RFC 1155.
/usr/samples/snmpd/mibII.my	Defines the ASN.1 definitions for the MIB II variables as defined in RFC 1213.

mount Command

Purpose

Makes a file system available for use.

Syntax

```
mount [ -f ] [ -n node ] [ -o options ] [ -p ] [ -r ] [ -v vfsmname ] [ -F AltFSfile ] [ -t type | [ device | node:directory ] directory | all | -a ] [ -V [generic_options] special_mount_points ]
```

Description

The **mount** command instructs the operating system to make a file system available for use at a specified location (the mount point). In addition, you can use the **mount** command to build other file trees that are made up of directory and file mounts. The **mount** command mounts a file system that is expressed as a device by using the *device* or *node:directory* parameter on the directory that is specified by the *directory* parameter. After the execution of **mount** finishes, the directory that is specified becomes the root directory of the newly mounted file system.

Only users with root authority or users who are members of the system group with write access to the mount point can perform file or directory mounts. The file or directory might be a symbolic link. The **mount** command uses the real user ID, not the effective user ID, to determine whether the user has appropriate access. System group members can issue device mounts, provided they have write access to the mount point and to those mounts that are specified in the `/etc/filesystems` file. Users with root user authority can issue any **mount** command.

Users can mount a device provided they belong to the system group and have appropriate access. When mounting a device, the **mount** command uses the *device* parameter as the name of the block device and the *directory* parameter as the directory on which to mount the file system.

If you enter the **mount** command without flags, the command displays the following information for the mounted file systems:

- The node (if the mount is remote).
- The object that is mounted.
- The mount point.
- The virtual-file-system type.
- The time mounted.

- Any mount options.

If you specify only the *directory* or *node:directory* parameter, the **mount** command takes it to be the name of the directory or file on which a file system, directory, or file is mounted (as defined in the `/etc/filesystems` file). The **mount** command looks up the associated device, directory, or file and mounts it. It is convenient to use the **mount** command because it does not require you to remember what is normally mounted on a directory or file. You can also specify only the device. In this case, the command obtains the mount point from the `/etc/filesystems` file.

The `/etc/filesystems` file must include a stanza for each mountable file system, directory, or file. This stanza must specify at least the name of the file system and either the device on which it resides or the directory name. If the stanza includes a mount attribute, the **mount** command uses the associated values. It recognizes five values for the mount attributes: **automatic**, **true**, **false**, **removable**, and **readonly**.

The **mount all** command causes all file systems with the **mount=true** attribute to be mounted in their normal places. This command is typically used during system initialization, and the corresponding mount operations are referred to as automatic mount operations.

By default, the **mount** command runs the **wlmcntrl** command to refresh the current assignment rules in the kernel after mounting the file system. In some situations (such as when many file systems are mounted at once, or when a rule for an inaccessible remote mount is present in the workload manager configuration), calling **wlmcntrl** automatically after mount might be undesirable.

If you want to override this behavior, set the environment variable `MOUNT_WLMCNTRL_SELFMANAGE` to any value. Setting the environment variable `MOUNT_WLMCNTRL_SELFMANAGE` to any value avoids calling the **wlmcntrl** command during the mount operation. To refresh the current assignment rules in the kernel, you must manually run `wlmcntrl -u -d ""` command. For more information, see [wlmcntrl](#) command.

Note:

1. If the **cdromd** CD and DVD **automount** daemon is enabled, those devices are automatically mounted as specified in the `/etc/cdromd.conf` file. Use the **cdumount** or **cdeject** command to unmount an automatically mounted CD or DVD. Use **stopsrc -s cdromd** to disable the CD/DVD **automount** daemon.
2. For CacheFS, the remote file system that is to be cached locally must be exported such that the root ID of the local system is not remapped on the remote host to nobody (or the ID that the remote host uses as the anonymous user). For example, if host A has to export a file system `/F` that is mounted with CacheFS on host B then the `/etc/exports` on host A need to have an entry similar to:

```
/F -rw,root=B
or
/F -ro,root=B
```

depending on the mount options used for the local CacheFS mount.

3. Mounting a JFS file system on a read-only logical volume is not supported.

Using mount on a JFS2 File System

The **mount** command can also be used to access a snapshot of a JFS2 file system as a directory tree. The snapshot on *device* is mounted read-only at *directory*. A snapshot can be mounted only once. When mounting a JFS2 file system with snapshots, the snapshots are activated.

You can use the System Management Interface Tool (SMIT) `smit mount` fast path to run this command.

Note: If the **mount** command encounters a Journaled File System (JFS) or Enhanced Journaled File System (JFS2) which was not unmounted before reboot, a replay of any JFS or JFS2 log records is attempted. To move a compatible JFS file system to a system that runs an earlier release of the operating system, the file system must always be unmounted cleanly before its movement. Failure to unmount first might result in an incompatible JFS log device. If the movement results in an unknown

log device, the file system must be returned to the system that runs the latter operating system release, and **fsck** must be run on the file system.

Flags

Item	Description
-a	Mounts all file systems in the <code>/etc/filesystems</code> file with stanzas that contain the true mount attribute.
all	Same as the -a flag.
-f	Requests a forced mount during system initialization to enable mounting over the root file system.
-F AltFSfile	Mounts on a file of an alternate file system, other than the <code>/etc/filesystems</code> file.
-n node	Specifies the remote node that holds the directory to be mounted. The node can be specified as a colon-separated IPv6 address. If the node is specified with the <code>node:directory</code> format, the colon-separated IPv6 address must be enclosed in square brackets.
-p	Mounts a file system as a removable file system. While open files are on it, a removable mounted file system behaves the same as a normally mounted file system. However, when no files are open (and no process has a current directory on the file system), all of the file system disk buffers in the file system are written to the medium, and the operating system forgets the structure of the file system.
-r	Mounts a file system as a read-only file system, regardless of its previous specification in the <code>/etc/filesystems</code> file or any previous command-line options.
-t type	Mounts all stanzas in the <code>/etc/filesystems</code> file that contain the type=type attribute and are not mounted. The <i>type</i> parameter specifies the name of the group.
-v vfsname	Specifies that the file system is defined by the <i>vfsname</i> parameter in the <code>/etc/vfs</code> file.

File System Specific Options

Item	Description
-o options	Specifies options. Options that are entered on the command line must be separated only by a comma. The following file system-specific options do not apply to all virtual file system types: atime Turns on access-time updates. If neither atime nor noatime option is specified, then atime option is the default value. bsy This option prevents the mount operation if the directory to be mounted over is the current working directory of a process.

Item**Description****cio**

Specifies the file system to be mounted for concurrent readers and writers. I/O on files in this file system behaves as if they are opened with **O_CIO** specified in the **open()** system call. Using the **cio** option prevents access in any manner other than CIO. It is impossible to use cached I/O on a file system mounted with the **cio** option. Mapping commands such as **mmap()** and **shmat()** fails with **EINVAL** when used on any file in a file system mounted with the **cio** option. One side-effect of this is that it is impossible to run binaries out of a **cio** mounted file system, since the loader might use **mmap()**.

Note: When you mount the file system by using the **cio** option, all applications must manage the serialization of files. Quotas are not supported by the **cio** option because quotas have their own serialization code.

dev

Specifies that you can open devices from this mount. If neither **dev** nor **nodev** option is specified, then **dev** option is the default value.

dio

Specifies that I/O on the file system behaves as if all the files were opened with **O_DIRECT** specified in the **open()** system call.

Note: Using the **-odio** or **-ocio** flags can help performance on certain workloads, but users must be aware that using these flags prevents file caching for these file systems. Because readahead is disabled for these file systems, it might decrease performance for large sequential reads.

fmode=octal

Specifies the mode for a file and directory. The default is 755.

gid=gid

Specifies the GID that is assigned to files in the mount. The default is **bin**.

log=lvname

Specifies the full path name of the file system logging logical volume name where the following file-system operations are logged.

log=NULL

Turns off logging and flushing of metadata for JFS2 file systems. Metadata is not flushed to the disk until the file system is unmounted. If the system stops abnormally before the file system is unmounted, the metadata changes are lost.

The JFS2 file system depends on the log information for metadata consistency. If the system stops abnormally during the metadata flush process for the JFS2 file system when the unmount operation is in progress, the file system cannot be recovered to a consistent state upon system reboot. In this case, the file system must be re-created.



Attention: Because of the risk of data loss, use this flag with caution.

maxpout=value

Specifies the pageout level for files on this file system at which threads must be slept. If **maxpout** option is specified, the **minpout** option must also be specified. Value must be nonnegative and greater than **minpout**. The default is the kernel **maxpout** level.

Item**Description****minpout=value**

Specifies the pageout level for files on this file system at which threads must be readied. If **minpout** is specified, the **maxpout** must also be specified. Value must be nonnegative. The default is the kernel **minpout** level.

nBufferPerPagerDevice=value

Specifies the number of buffers for JFS2 file system on a per-file system basis. The **nBufferPerPagerDevice** mount option specifies only the number of file system buffers that start on the paging device. If the **nBufferPerPagerDevice** mount option is not specified, then the value of the **j2_nBufferPerPagerDevice** tunable parameter is used to allocate the file system buffers. The range for **nBufferPerPagerDevice** mount option is identical to that of the **j2_nBufferPerPagerDevice** tunable parameter.

The **j2_dynamicBufferPreallocation** tunable parameter of the **ioo** command must be tuned to specify the file system buffers for JFS2 file system on a per-system basis. In a server with many file systems, a system might run out of buffers if a file system has a larger number of concurrent I/O than the other file systems. Currently, increasing the value of the **j2_nBufferPerPagerDevice** tunable parameter applies to all the file systems. If you set the **j2_nBufferPerPagerDevice** tunable parameter to a large value, a high amount of pinned memory is used when only one file system requires a higher buffer value than the rest. In such cases, you can use the **nBufferPerPagerDevice** mount option to specify the file system buffers for JFS2 file system on a per-file system basis.

After tuning the **j2_dynamicBufferPreallocation** tuning parameter of the **ioo** command, run the **vmstat -v** command. In the output displayed after running the **vmstat -v** command, check the number of external pager filesystem I/Os blocked with no fsbuf counter. If the value of this counter increases quickly and keeps increasing, then the **nBufferPerPagerDevice** mount option can be used to specify the number of file system buffers for JFS2 on a per-file system basis. If the kernel must wait for free buffers, it puts the process on a wait list before the I/O operation starts and wakes up the process when buffers are available. If you are using striped logical volumes or disk arrays, consider increasing the number of buffers for JFS2 file system.

noatime

Turns off access-time updates. Using this option can improve performance on file systems where many files are read frequently and seldom updated. If you use the option, the last access time for a file cannot be determined. If neither **atime** nor **noatime** option is specified, then **atime** option is the default value.

nocase

Turns-off case mapping. This option is useful for CDRoms that uses the ISO 9660:1998/HSG standard.

nodev

Specifies that you cannot open devices from this mount. This option returns a value of **ENXIO** if a failure occurs. If neither **dev** nor **nodev** option is specified, then **dev** option is the default value.

Item**Description****noguard**

Mount the file system regardless of the current `mountguard` setting that would otherwise guard the file system against unsupported concurrent mounts in a PowerHA or other clustering environment. If `mountguard` is enabled by the `chfs` or `crfs` command, the file system cannot be mounted if it appears to be mounted on another node or system. Specifying the **noguard** option temporarily overrides the `mountguard` setting.

norbr

Mounts the file system without the release-behind-when-reading capability. If none of the release-behind options are specified, the **norbrw** is the default value.

norbrw

Mounts the file system without both the release-behind-when-reading and release-behind-when-writing capabilities. If none of the release-behind options are specified, the **norbrw** is the default value.

norbw

Mounts the file system without the release-behind-when-writing capability. If none of the release-behind options are specified, the **norbrw** is the default value.

nosuid

Specifies that execution of **setuid** and **setgid** programs by way of this mount is not allowed. This option returns a value of **EPERM** if a failure occurs. If neither **suid** nor **nosuid** option is specified, then **suid** option is the default value.

rbr

Mount the file system with the release-behind-when-reading capability. When sequential reading of a file in this file system is detected, the real memory pages that are used by the file are released once the pages are copied to internal buffers. If none of the release-behind options are specified, the **norbrw** is the default.

Note: When **rbr** is specified, the **D_RB_READ** flag is ultimately set in the `_devflags` field in the `pdentry` structure.

rbw

Mount the file system with the release-behind-when-writing capability. When sequential writing of a file in this file system is detected, the real memory pages that are used by the file are released once the pages written to disk. If none of the release-behind options are specified, the **norbrw** is the default.

Note: When **rbw** is specified, the **D_RB_WRITE** flag is set.

rbrw

Mount the file system with both release-behind-when-reading and release-behind-when-writing capabilities. If none of the release-behind options are specified, the **norbrw** is the default.

Note: If **rbrw** is specified, both the **D_RB_READ** and the **D_RB_WRITE** flags are set.

Item**Description****remount**

Changes the mount options of a mounted file system. For JFS2 file systems, you can specify the following mount options with the **remount** option to change the settings of a mounted file system. For any mount options not specified, no change is made to the current corresponding settings of the file system.

atime, noatime; dev, nodev; logdev; maxpout, minpout; rbr, norbr; rbw, norbw; rbrw, norbrw, rw, ro, rox; suid, nosuid.

Note:

1. External-snapshot mounted file systems cannot be remounted to read/write file systems.
2. You cannot use the **rw** and **ro** remount options on a file system that is managed by a data management application programming interface (DMAPI).
3. If **logdev** is specified, the new log device must be in the same volume group as the existing log device. You cannot change an external log device to an internal log device or vice-versa if you specify the **logdev** option. Use the **logshuffle** option in the **chfs** command for that functionality.

For NFS, there are three types of mount requests.

duplicate mount

If the node, object, mount point, and the options that are specified in the **mount** command are the same as those for an existing mount, the **mount** command returns information about a successful mount, but a new mount is not created.

new mount

If the **remount** option is not specified, the **mount** command creates a new mount. If the node, object, mount point, or the constant options that are specified in the **mount** command are different than those for the existing mounts, the **mount** command fails if the **remount** option is specified.

remount

If the node, object, and mount point are the same as those for a top-most mount, but the remount options are different, the remount operation modifies the mount options of an existing mount. In this case, NFS performs the remount operation.

A top-most mount does not have another mount on top of it. For remount requests, the following options can be modified: **acdirmax, acdirmin, acregmax, acregmin, actimeo, fastattr, grpuid, hard, intr, noac, nocto, nodev, nointr, nosuid, posix, retrans, ro, rsize, rw, secure, sec, soft, timeo, wsize, biops, extraattr, nodircahce, prefer, otwattr, maxgroups, and proto**. Other options are classified as constant options.

ro

Specifies that the mounted file is read-only, regardless of its previous option specification in the `/etc/filesystems` file or any previous command-line options. The default value is **rw**.

rw

Specifies that the mounted file is read/write accessible, regardless of its previous option specification in the `/etc/filesystems` file or any previous command-line options. The default value is **rw**.

Item	Description
snapshot	Specifies the <i>device</i> to be mounted is a snapshot. The snapped file system for the specified snapshot must already be mounted or an error message is displayed.
snapto=snapshot	Specifies the location to start a snapshot with the value of <i>snapshot</i> when mounting the specified JFS2 file system. The <i>snapshot</i> parameter specifies the name of an internal snapshot if the <i>snapshot</i> parameter does not included a forward slash (/), that is, no path information.
suid	Specifies that execution of setuid and setgid programs by way of this mount is allowed. If neither suid nor nosuid option is specified, then suid option is the default value.
upcase	Changes case mapping from default lowercase to uppercase. This option is useful for CDROMs that uses the ISO 9660:1998/HSG standard.
uid=uid	Specifies the UID that is assigned to files in the mount, the default is bin.
wrkgrp=workgroup	Specifies the workgroup that the SMB server belongs.

NFS Specific Options

Item	Description
-o options	Specifies options. Options that you enter on the command line must be separated only by a comma without any space. The following NFS-specific options do not apply to all virtual file system types:
acdirmax=n	Holds cached attributes for no more than <i>n</i> seconds after directory update. The default is 60 seconds.
acdirmin=n	Holds cached attributes for at least <i>n</i> seconds after directory update. The default is 30 seconds.
acl	Requests by using the Access Control List RPC program for this NFS mount. If the acl option is used, the ACL RPC program is used only if the NFS server provides it. The default is noacl .
acregmax=n	Holds cached attributes for no longer that <i>n</i> seconds after file modification. The default is 60 seconds.
acregmin=n	Holds cached attributes for at least <i>n</i> seconds after file modification. The default is 30 seconds.

Item	Description
actimeo=<i>n</i>	Sets minimum and maximum times for regular files and directories to <i>n</i> seconds. If this option is set, it overrides any settings for the acregmin , acregmax , acdirmin , and acdirmax options.
bg	Attempts mount in the background if first attempt is unsuccessful. The default value is fg .
biops=<i>n</i>	Sets the maximum number of biops threads that perform asynchronous I/O RPC requests for an NFS mount. The maximum value that can be set is 128. Values greater than 128 are limited to 128 within the NFS client. The NFS client dynamically manages the number of running biops threads up to the maximum based on activity. The default maximums for the different NFS protocols are 7 for NFS version 2 and 32 for NFS version 3 and NFS version 4. These defaults are subject to change in future releases.
cio	Specifies the file system to be mounted for concurrent readers and writers. I/O on files in this file system behaves as if they were opened with O_CIO specified in the open() system call. Using this option prevents access in any manner other than CIO. It is impossible to use cached I/O on a file system mounted with the cio option. Mapping commands such as mmap() and shmat() fails with EINVAL when used on any file in a file system mounted with the cio option. One side-effect of this is that it is impossible to run binaries out of a cio mounted file system, since the loader might use mmap() . Note: When you mount the file system by using the cio option, all applications must manage the serialization of files. Quotas are not supported by the cio option because quotas have their own serialization code.
cior	Specifies to allow read-only files to open in the file system. I/O on files in this file system behaves as if they were opened with O_CIO O_CIOR specified in the open() system call. Using this option prevents access in any manner other than O_CIO O_CIOR and read-only. An attempt to open only with O_CIO also fails. This option can be used only along with cio .
dio	Specifies that I/O on the file system behaves as if all the files were opened with O_DIRECT specified in the open() system call. Note: Using the -odio or -ocio flags can help performance on certain workloads, but users must be aware that using these flags prevents file caching for these file systems. Because readahead is disabled for these file systems, it might decrease performance for large sequential reads.
fastattr	This option bypasses the requirement that a file, which is currently written be sent to the server before the attributes of the file are read. This option is to be used with caution, since it causes the client to assume that the file data that has not yet reached the server is written without problem. If there are write errors, the client and server have different opinions on what the size of the file really is. Likewise, a client is not aware of attribute changes to the file that are made by another client, so this option must not be used in environments where two clients are writing to the same files.

Item	Description
fg	Attempts mount in foreground if first attempt is unsuccessful. fg is the default value.
grpuid	Directs any file or directory that is created on the file system to inherit the group ID of the parent directory.
hard	Retries a request until the server responds. The option is the default value.
intr	This option allows keyboard interrupts on hard mounts.
llock	Requests that files lock locally at the NFS client. NFS network file locking requests are not sent to the NFS server if the llock option is used.
maxgroups=<i>n</i>	<p>This option indicates that NFS RPC calls by using AUTH_UNIX might include up to <i>n</i> member groups of information. Using this option to increase the number of member groups beyond the RPC protocol standard of 16 only works against servers that support more than 16 member groups. Otherwise, the client experiences errors.</p> <p>Values less than 16 or greater than 64 are ignored. By default, the protocol standard maximum of 16 is adhered to. AIX NFS servers accept and process AUTH_UNIX credentials with up to 64 groups that start with AIX 5L Version 5.2 with the 5200-01 Recommended Maintenance package. The actual number of member groups that are sent by the NFS client depends on the number of groups the involved user is a member of, and might be limited by the length of the NFS client's hostname (which is included in the AUTH_UNIX information).</p>
noac	<p>Specifies that the mount command performs no attribute or directory caching. If you do not specify this option, the attributes (including permissions, size, and timestamps) for files and directories are cached to reduce the need to perform over-the-wire NFSPROC_GETATTR Remote Procedure Calls (RPCs). The NFSPROC_GETATTR RPC enables a client to prompt the server for file and directory attributes. The acregmin, acregmax, acdirmin, and acdirmax options control the length of time for which the cached values are retained.</p>
noacl	Specifies not to use the Access Control List RPC program for this NFS mount request. The default is noacl .
nointr	Specifies no keyboard interrupts allowed on hard mounts.
port=<i>n</i>	Sets server Internet Protocol (IP) port number to <i>n</i> . The default value is the 2049.
posix	Requests that pathconf information is exchanged and made available on an NFS Version 2 mount. Requires a mount Version 2 rpc.mountd at the NFS server.

Item**Description****proto=[udp|tcp]**

Specifies the transport protocol. The default is **tcp**. Use the **proto=[udp|tcp]** option to override the default.

proto=udp cannot be specified if **vers=4**.

retrans=n

Sets the number of NFS transmissions to *n*. The default value is 5. The **retrans** setting determines how many times the NFS client retransmits a given UDP RPC request to an NFS server for file system operations. The **retrans** setting is not used during communication with the NFS server **rpc.mountd** service when processing NFS version 2 and 3 mounts. Retries to **rpc.mountd** are controlled with the **retry** mount option.

retry=n

Sets the number of times the mount is attempted to *n*; the default value is 1000. When the retry value is 0, the system makes 10,000 attempts.

rsize=n

Sets the read buffer size to *n* bytes. Beginning with AIX Version 6.1, the default value is 64 KB and the maximum value is 512 KB when using Version 3 and Version 4 of the NFS protocol.

secure

Specifies that the **mount** command uses Data Encryption Standard (DES) for NFS transactions. The Data Encryption Standard (DES) is not supported in NFS Version 4, use **krb5** instead.

sec=[flavor[:flavor...]]

Specifies a list of security methods that might be used to access files under the mount point. Allowable flavor values are:

sys

UNIX authentication. This value is the default method.

dh

DES authentication. The Data Encryption Standard (DES) is not supported in NFS Version 4, use **krb5** instead.

krb5

Kerberos. Authentication only.

krb5i

Kerberos. Authentication and integrity.

krb5p

Kerberos. Authentication, integrity, and privacy.

The **secure** option might be specified, but not along with a **sec** option. The **secure** option is deprecated and might be eliminated in a future release. Use **sec=dh** instead.

sec=[flavor1:...:flavorn]

The **sec** option specifies the security flavor list for the NFS mount. The available flavors are **des**, **unix**, **sys**, **krb5**, **krb5i**, and **krb5p**. This option applies to AIX 5.3 or later.

shortdev

Specifies that you are mounting a file system from a host that does not support 32-bit device special files.

Item	Description
	<p>soft This option returns an error if the server does not respond. The default value is hard.</p>
	<p>timeo=<i>n</i> Sets the Network File System (NFS) time out period to <i>n</i> tenths of a second. For TCP mounts, the default timeout is 100, which equals 10 seconds. For UDP mounts, the default timeout is 11, which equals 1.1 seconds, but varies depending on the NFS operation that takes place. For UDP mounts, the timeout increases for each failed transmission, with a maximum value of 20 seconds. Each transmission will be attempted twice after which the timeout value is updated. The timeo option does not apply to communication from the NFS client to the rpc.mountd service on NFS servers. A timeout of 30 seconds is used when making calls to rpc.mountd.</p>
	<p>vers=[2 3 4] Specifies NFS version. The default is the version of the NFS protocol that is used between the client and server and is the highest one available on both systems. If the NFS server does not support NFS Version 3, the NFS mount uses NFS Version 2. Use the vers=[2 3 4] option to select the NFS version. By default, the NFS mount never uses NFS Version 4 unless specified. The vers=4 applies to AIX 5.3 or later.</p>
	<p>wsiz=<i>n</i> Sets the write buffer size to <i>n</i> bytes. Beginning with AIX Version 6.1, the default value is 64 KB and the maximum value is 512 KB when using Version 3 and Version 4 of the NFS protocol.</p>

CacheFS Specific Options

The CacheFS-specific version of the **mount** command mounts a cached file system; if necessary, it NFS-mounts its back file system. It also provides several CacheFS-specific options for controlling the caching process.

To mount a CacheFS file system, use the **mount** command with the **-V** flag followed by the argument. The following **mount** flags are available.

The following arguments to the **-o** flag are specifically for CacheFS mounts. Options that you enter on the command line must be separated only by a comma without any space.

Note: The **backfstype** argument must be specified.

Item	Description
-o	Specifies options.
	<p>acdirmax=<i>n</i> Specifies that cached attributes are held for no more than <i>n</i> seconds after a directory update. Before <i>n</i> seconds, CacheFS checks to see whether the directory modification time on the back file system changes. If the directory modification time on the back file system changes, all information about the directory is purged from the cache and new data is retrieved from the back file system. The default value is 60 seconds.</p>

Item	Description
acdirmin=<i>n</i>	<p>Specifies that cached attributes are held for at least <i>n</i> seconds after directory update. After <i>n</i> seconds, CacheFS checks to see whether the directory modification time on the back file system changes. If the directory modification time on the back file system changes, all information about the directory is purged from the cache and new data is retrieved from the back file system. The default value is 30 seconds.</p>
acregmax=<i>n</i>	<p>Specifies that cached attributes are held for no more than <i>n</i> seconds after file modification. After <i>n</i> seconds, all file information is purged from the cache. The default value is 30 seconds.</p>
acregmin=<i>n</i>	<p>Specifies that cached attributes are held for at least <i>n</i> seconds after file modification. After <i>n</i> seconds, CacheFS checks to see whether the file modification time on the back file system changes. If the file modification time on the back file system changes, all the information about the file is purged from the cache and new data is retrieved from the back file system. The default value is 30 seconds.</p>
actimeo=<i>n</i>	<p>Sets acregmin, acregmax, acdirmin, and acdirmax to <i>n</i>.</p>
backfstype=<i>file_system_type</i>	<p>The file system type of the back file system (for example, nfs).</p>
backpath=<i>path</i>	<p>Specifies where the back file system is already mounted. If this argument is not supplied, CacheFS determines a mount point for the back file system.</p>
cachedir=<i>directory</i>	<p>The name of the cache directory.</p>
cacheid=<i>ID</i>	<p>ID is a string that specifies an instance of a cache. If you do not specify a cache ID, CacheFS constructs one.</p>
demandconst	<p>Enables maximum cache consistency checking. By default, periodic consistency checking is enabled. When you enable demandconst, it checks on every read and write.</p> <p>Note: If this option is used the first time a specific CacheFS is mounted, then this option must also be specified for subsequent mounts. There is state information that is stored in the cache control files that enforces consistent use of this option.</p>
local_access	<p>This option causes the front file system to interpret the mode bits used for access checking instead of having the back file system verify access permissions. Do not use this argument with secure NFS.</p>

Item	Description
	<p>noconst Disables cache consistency checking. By default, periodic consistency checking is enabled. Specify noconst only when you know that the back file system is not modified. Trying to perform cache consistency check by using cfsadmin-s results in error. demandconst and noconst are mutually exclusive.</p> <p>Note: If this option is used the first time a specific CacheFS is mounted, then the option must also be specified for subsequent mounts. There is state information that is stored in the cache control files that enforces consistent use of this option.</p>
	<p>purge Purges any cached information for the specified file system.</p> <p>Note: If this option is used the first time a specific CacheFS is mounted, then the option must also be specified for subsequent mounts. There is state information that is stored in the cache control files that enforces consistent use of this option.</p>
	<p>rw ro Read-write (default) or read-only.</p>
	<p>suid nosuid Allows (default) or disallows set-uid execution.</p>
	<p>write-around non-shared Writes modes for CacheFS. The write-around mode (the default) handles the writes the same as NFS does. Writes are made to the back file system, and the affected file is purged from the cache. You can use the non-shared mode when you are sure that no one else is writing to the cached file system.</p> <p>Note: If this option is used the first time a specific CacheFS is mounted, then the option must also be specified for subsequent mounts. There is state information that is stored in the cache control files that enforces consistent use of this option.</p>
	<p>mfsid Turns on global view. In the NFS v4 system, you can traverse through the exported namespace on the server side. You need to specify this option to go over the file system.</p> <p>Restriction: mfsid is an option if the backend file system for CacheFS is NFS v4.</p>
-V	Mounts a CacheFS file system.

Server Message Block (SMB) client file system specific options

Item	Description
-o options	<p>Specifies options for mounting the SMB client file system. Options that you enter on the command line must be separated only by a comma. Do not insert a space before or after a comma. The following options are available for the SMB client file system:</p> <p>fmode Sets a file or directory to octal mode for access permissions. The default value is 755.</p> <p>uid Assigns a user ID to files during the mount operation. The default value is root.</p> <p>gid Assigns a group ID to files during the mount operation. The default value is system.</p> <p>wrkgrp Specifies the workgroup to which the SMB server belongs. This parameter is mandatory to mount the SMB client file system.</p> <p>port Specifies the port number. The valid values are 445 and 139. The default value is 445. Port 139 is supported only when the specified server address is in IPv4 format.</p> <p>Note: encryption option is not supported when the port specified is 139.</p> <p>pver 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. For the value auto, the SMB protocol version 2.1 or version 3.0.2 is used based on the specified SMB server.</p> <p>signing Specifies whether the file system in the SMB client needs a digital signature for communication with the SMB server file system. The valid values are <code>enabled</code> and <code>required</code>. When this parameter is set to <code>enabled</code>, the file system in the SMB client does not digitally sign the data packets unless the file system in the SMB server needs digital signatures for communication with the file system in the SMB server. When this parameter is set to <code>required</code>, the file system in the SMB client must digitally sign the data packets for communication with the file system in the SMB server. If you do not specify the value for the signing parameter by using the mount command, a default value is used from the tunable parameter values of the kernel that are set by using the <u>smbctune</u> command.</p> <p>secure_negotiate Specifies whether the file system in the SMB client needs a secure dialect negotiation capability. SMB Dialect 3.0.2 implements secure dialect negotiation to protect against security-downgrade attacks. The valid values are <code>desired</code>, <code>required</code>, and <code>disabled</code>. If you do not specify the value by using the mount command, a default value is used from tunable parameter values of the kernel that are set by using the <u>smbctune</u> command.</p> <p>encryption Specifies whether the file system in the SMB client requires data encryption. The valid values are <code>desired</code>, <code>required</code>, and <code>disabled</code>. If you do not specify the value by using the mount command, a default value is used from the tunable parameter values of the kernel that are set by using the <u>smbctune</u> command.</p> <p>Note: encryption option is not supported when the port specified is 139.</p>

Item	Description
-o options (continued)	spn Specifies the <u>service principal name</u> (SPN) that must be used in the SMB client mount points. The format of the spn parameter is <code>cifs/< smbServerHostName ></code> , where <code>smbServerHostName</code> is the fully qualified domain name (FQDN) of the SMB server or the name that the Kerberos resolves as the SMB server. By default, SPN is constructed automatically by the SMB client file system as <code>cifs/< smbServerHostName ></code> .

If the options that are used with the **mount** command (`pver`, `signing`, `secure_negotiate`, or `encryption`) are unspecified by using the **-o** flag, the default values for the **mount** command options are initialized by using the new values of the kernel tunable parameters (`smbc_protocol_version`, `smbc_signing`, `smbc_secure_negotiate`, `smbc_encryption`). The kernel tunable parameters are initialized from tunable parameters that are defined in the `smbctune.conf` file. These parameters can also be modified by using the **smbctune** command.

The following table shows the kernel tunable parameters of the **mount** command and the corresponding kernel tunable parameters that can be set in the `smbctune.conf` file:

Options of the -o flag (mount command)	Corresponding kernel tunable parameter of the smbctune.conf file	Valid values
pver	smbc_protocol_version	2.1, 3.0.2, auto
signing	smbc_signing	enabled, required
secure_negotiate	smbc_secure_negotiate	desired, required, disabled
encryption	smbc_encryption	desired, required, disabled

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 the mounted file systems, enter the following command:

```
mount
```

This command produces output similar to the following output:

```
node   mounted          mounted over  vfs    date           options
-----
      /dev/hd0         /             jfs    Dec 17 08:04   rw, log =/dev/hd8
      /dev/hd3         /tmp          jfs    Dec 17 08:04   rw, log =/dev/hd8
      /dev/hd1         /home         jfs    Dec 17 08:06   rw, log =/dev/hd8
      /dev/hd2         /usr          jfs    Dec 17 08:06   rw, log =/dev/hd8
sue    /home/local/src  /usr/code    nfs    Dec 17 08:06   ro, log =/dev/hd8
```

For each file system, the **mount** command lists the node name, the device name, the name under which it is mounted, the virtual-file-system type, the date and time it was mounted, and its options.

- To mount all default file systems, enter the following command:

```
mount all
```

This command sequence mounts all standard file systems in the `/etc/filesystems` file that is marked by the **mount=true** attribute.

3. To mount a remote directory, enter the following command:

```
mount -n nodeA /home/tom.remote /home/tom.local
```

This command sequence mounts the `/home/tom.remote` directory that is located on `nodeA` onto the local `/home/tom.remote` directory. It assumes the default `VfsName` parameter=**remote**, which must be defined in the `/etc/vfs` file.

4. To mount a file or directory from the `/etc/filesystems` file with a specific type, enter the following command:

```
mount -t remote
```

This command sequence mounts all files or directories in the `/etc/filesystems` file that have a stanza that contains the **type=remote** attribute.

5. To CacheFS-mount the file system that is already NFS-mounted on `/usr/abc`, enter the following command:

```
mount -V cachefs -o backfstype=nfs,backpath=/usr/abc,
cachedir=/cache1 server1:/user2 /xyz
```

The lines similar to the following appear in the `/etc/mnttab` file after the mount command is executed:

```
server1:/user2 /usr/abc nfs
/usr/abc /cache1/xyz cachefs backfstype=nfs
```

6. To mount a snapshot, enter the following command:

```
mount -o snapshot /dev/snapsb /home/janet/snapsb
```

This command mounts the snapshot that is contained on the `/dev/snapsb` device onto the `/home/janet/snapsb` directory.

7. To mount a file system and create a snapshot, enter the following command:

```
mount -o snapto=/dev/snapsb /dev/sb /home/janet/sb
```

This command mounts the file system that is contained on the `/dev/sb` device onto the `/home/janet/sb` directory and creates a snapshot for the file system on the `/dev/snapsb` device.

8. To access files on an SMB server as a local file system, enter the following command:

```
mount -v cifs -n pezman/user1/pass1 -o uid=201,fmode=750 /home /mnt
```

9. To mount an SMB client file system as a local mount point, enter the following command:

```
mount -v smbclient -n llm140.xyz.com/cec102usr1/Passw0rd \
-o wkggrp=SMB_21.FVT,port=445,signing=required /some_share /mnt
```

Where, `llm140.xyz.com` is the Windows server, `cec102usr1` is the Kerberos user name, `Passw0rd` is the password of the Kerberos user, `SMB_21.FVT` is the workgroup, `some_share` is the share point on the Windows system, and `/mnt` is the local mount point.

10. To remount the mounted read-only JFS2 file system to a read/write file system, enter the following command:

```
mount -o remount,rw fsname
```

11. To mount all on a file `/tmp/fs1` of an alternate file system, enter the following command:

```
mount -F /tmp/fs1 all
```

Files

Item	Description
<code>/etc/filesystems</code>	Lists the known file systems and defines their characteristics.
<code>/etc/vfs</code>	Contains descriptions of virtual-file-system types.

mountd Daemon

Purpose

Answers requests from clients for file system mounts.

Syntax

```
/usr/sbin/rpc.mountd [ -n ] [ -N ] [ -x ] [ -r ]
```

Description

The **mountd** daemon is a Remote Procedure Call (RPC) that answers a client request to mount a file system. The **mountd** daemon finds out which file systems are available by reading the `/etc/xtab` file.

In addition, the **mountd** daemon provides a list of currently mounted file systems and the clients on which they are mounted. You can display this list by using the **showmount** command.

The **mountd** daemon listens for requests on the port specified in the `/etc/services` file for the service **mountd**. If the `/etc/services` file does not specify a port, one will be chosen when the daemon starts. For example, adding the lines:

```
mountd 6666/tcp
mountd 6666/udp
```

will cause **mountd** to listen for requests on port 6666.

Examples

The **mountd** daemon is started from the `/etc/rc.nfs` file. The **mountd** daemon can be started and stopped by the following System Resource Controller (SRC) commands:

```
startsrc -s rpc.mountd
stopsrc -s rpc.mountd
```

To change the parameters passed to the **mountd** daemon, use the **chssys** command. For example:

```
chssys -s rpc.mountd -a Argument
```

The change will not take effect until the daemon is restarted.

Flags

Item	Description
-n	Allows clients that use older versions of NFS to mount file systems. This option makes the system less secure. It is the default.
-N	Deny mount requests originating from non-privileged ports. This is the opposite of using the -n flag, and is not enabled by default.
-x	Allows mount request without reverse lookup check.
-r	Turn off the <code>/etc/rmtab</code> file updates.

Files

Item	Description
<u>/etc/exports</u>	Lists the directories that the server can export.
<u>/etc/inetd.conf</u>	Defines how the inetd daemon handles Internet service requests.
<u>/etc/xtab</u>	Lists currently exported directories.
<u>/etc/services</u>	Defines the sockets and protocols used for Internet services. Contains information about the known services used in the DARPA Internet network.

mpcstat Command

Purpose

Displays operational information about a Multi-Protocol Over ATM (MPOA) Client.

Syntax

```
mpcstat [ -a -c -e -i -m -r -s -t -v] [ Device_Name ]
```

Description

This command displays Multi-Protocol Over ATM (MPOA) Client operational information gathered by a specified MPOA Client device. If a MPOA Client (MPC) device name is not entered, information for the available MPC appear. You can use the flags to narrow down your search to specify specific categories of information such as Configuration, Egress Cache Entries, Ingress Cache Entries, MPOA Servers, Shortcut Virtual Connections, and Statistics, or you can elect to have all of the information categories display.

You can also toggle debug tracing on or off and reset statistics counters.

Parameters

Item	Description
<i>Device_Name</i>	The name of the MPOA Client device name, for example, <i>mpc0</i> .

Flags

Item	Description
-a	Requests that all of the MPOA Client information appear. Note that this flag does not reset statistics counters or toggle trace. If a flag is not entered, the -a flag is the default flag.
-c	Requests the configuration
-e	Requests the egress (incoming) cache
-i	Requests the ingress (outgoing) cache.
-m	Requests the list of MPOA Servers in use.
-r	Resets the statistics counters after reading.
-s	Requests the statistics counters.
-t	Toggles full debug trace on or off.
-v	Requests the list of Shortcut Virtual Connections.

The following information appears for all valid calls and contains the following fields:

Device Name

Displays the device name of the MPOA Client.

MPC State

Displays the current state of the MPOA Client.

Example States:

Idle	Registering with the ELAN.
Initializing	Registering with the switch.
Operational	Fully operational.
Network Down	Network is currently unavailable.

MPC Address

Displays the MPOA Client's 20-byte ATM address for a specific ATM adapter port device name. The adapter port device name is also displayed.

Elapsed Time

Displays the real time period which has elapsed since statistics were last reset.

MPC Configuration

Selected with the **-a** or **-c** flags. Displays the network administrator's pre-configured attributes for the MPOA Client, or the values provided by a Lan Emulation Configuration Server (LECS).

MPC Egress Cache

Selected with the **-a** or **-e** flags. Displays the current egress cache entries. Included are the state of the entry, its Level-3 address, and ATM shortcut address, as well as additional descriptive values associated with each entry.

Example States

Active	Has active shortcut connection.
Purging	Purging the egress MPOA Server entry.
DP Purge	Purging the remote MPOA Client data plane.
Inactive	No current activity on shortcut connection.

MPC Ingress Cache:

Selected with the **-a** or **-i** flags. Displays the current ingress cache entries. Included are the state of the entry, its Level-3 address, and ATM shortcut address, as well as additional descriptive values associated with each entry.

Example States

Flow Detect	Waiting for packet threshold to enable shortcut.
Resolution	Packet threshold reached, resolving shortcut.
Hold Down	Shortcut resolution failed, waiting for retry.
Resolved	Shortcut resolution sequence complete.

MPOA Server List

Selected with the **-a** or **-m** flags. Displays a list of MPOA Servers currently known by this MPC. Included in each entry are the name of the LE Client that identified the MPS, the MPS ATM address, and the MPS LAN MAC address.

MPC Statistics

Selected with the **-a** or **-s** flags. Displays the current Transmit, Receive, and General statistics for this MPOA Client.

Shortcut Virtual Connection

Selected with the **-a** or **-v** flags. Displays the current list of shortcut virtual circuits in use by the MPOA client. Included are virtual path and channel values, VC state, ATM device name, as well as additional descriptive values associated with each entry.

Example States:

Idle	Call idle.
Signalling	Call placed but not established.

Operational	Call connected; data path valid.
Released	Call released.
Retry	Temporary call failure; will be retried.
Hold Down	Call failure; will be suspended for hold down period.

Exit Status

If an invalid *Device_Name* is specified, this command produces error messages stating that it could not connect to the device. Note that MPOA is a protocol extension to the ATM LAN Emulation protocol, and must have a corresponding and available LE Client to be operational. Examples of an invalid device error message are:

```
MPCSTAT: Device is not an MPOA device.
MPCSTAT: No LEC device with MPOA enabled.
MPCSTAT: Device is not available.
```

mpstat Command

Purpose

Collects and displays performance statistics for all logical processors in the system.

Syntax

```
mpstat [{ -d | -i | -s | -a | -h | -v | -E } ] [ -w ] [ -O Options ] [ -@ wparname ] [ interval [ count ] ]
mpstat [-X [-o filename]] [interval[count]]
```

Restriction: The *wparname* parameter is restricted to use inside workload partitions.

Description

The **mpstat** command collects and displays performance statistics for all logical processors in the system. Users can define both, the number of times the statistics are displayed, and the interval at which the data is updated.

The *interval* parameter specifies the amount of time in seconds between each report. If you do not specify the *interval* parameter, the **mpstat** 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, its 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 **mpstat** command with no options generates a single report that contains the performance statistics for all logical processors since boot time.

When the **mpstat** command is invoked, it displays two sections of statistics. The first section displays the System Configuration, which is displayed when the command starts and whenever there is a change in the system configuration. The second section displays the Utilization Statistics which are displayed in intervals and at any time the values of these metrics are deltas from previous interval.

The following information is displayed in the system configuration section:

lcpu

Indicates the number of online logical processors.

ent

Indicates the entitled processing capacity in processor units. This information is displayed only when the partition type is shared.

mode

Indicates whether the partition processor capacity is capped or uncapped allowing it to consume idle cycles from the shared pool. Dedicated LPAR is capped or donating.

rset

Indicates the resource-set type (regular or exclusive) that is associated with the WPAR. This information is displayed only when there is a resource set that is associated with the WPAR.

The performance statistics displayed by `mpstat` are listed below:

CPU

(All flags) Logical processor ID.

Note: The logical processor ID that is associated with the resource set of a WPAR is prefixed by an asterisk (*) when you run the `mpstat` command inside a WPAR with the `-s` or `-@` flag.

min

(Default, `-a` flag) Minor page faults (page faults with no IO).

maj

(Default, `-a` flag) Major page faults (page faults with disk IO).

mpcs

(`-a`, `-i` flag) Number of mpc send interrupts.

mPCR

(`-a`, `-i` flag) Number of mpc receive interrupts.

mpc

(Only default) Total number of inter-processor calls .

dev

(`-a`, `-i` flag) Number of device interrupts.

soft

(`-a`, `-i` flag) Number of software interrupts.

dec

(`-a`, `-i` flag) Number of decremter interrupts.

ph

(`-a`, `-i` flag) Number of phantom interrupts.

int

(Only default) Total number of interrupts.

cs

(Default, `-a` flag) Total number of context switches.

ics

(Default, `-a` flag) Total number of involuntary context switches.

bound

(`-a`, `-d` flag) Total number of threads that are bound.

rq

(Default, `-a`, `-d` flag) Run queue size.

push

(`-a`, `-d` flag) Number of migrations due to starvation load balancing .

S3pull

(`-a`, `-d` flag) Number of migrations outside the scheduling affinity domain 3 due to idle stealing.

S3grd

(`-a`, `-d` flag) Number of dispatches from global runqueue, outside the scheduling affinity domain 3.

mig

(Only default) Total number of thread migrations (to another logical processor).

S0rd

(`-a`, `-d` flag) The percentage of thread redispaches within the same logical processor with scheduling affinity domain 0.

S1rd

(-a, -d flag) The percentage of thread redispaches within the same physical processor or core with scheduling affinity domain 1.

S2rd

(-a, -d flag) The percentage of thread redispaches within the same chip set, but not within the same processor core with scheduling affinity domain 2.

S3rd

(-a, -d flag) The percentage of thread redispaches within the same MCM (multiple chip module) , but not within the same chip set with scheduling affinity domain 3.

S4rd

(-a, -d flag) The percentage of thread redispaches on different MCMs within the same CEC or Plane with scheduling affinity domain 4.

S5rd

(-a, -d flag) The percentage of thread redispaches on a different CEC or Plane with scheduling affinity domain 5.

S3hrd

(-a, -d flag) The percentage of local thread dispatches on this logical processor.

S4hrd

(-a, -d flag) The percentage of near thread dispatches on this logical processor.

S5hrd

(-a, -d flag) The percentage of far thread dispatches on this logical processor.

lpa

(Only default) Logical processor affinity. The percentage of logical processor re-dispatches within the scheduling affinity domain 3.

sysc

(Default, -a flag) Number of system calls.

us

(Default, -a flag, -v flag) The percentage of physical processor utilization that occurred while executing at the user level (application).

If the -v flag is used, then utilization is based on the virtual processor.

sy

(Default, -a flag, -v flag) The percentage of physical processor utilization that occurred while executing at the system level (kernel).

If the -v flag is used, then utilization is based on the virtual processor.

wa

(Default, -a flag, -v flag) The percentage of time that the logical processor was idle during which it had an outstanding disk I/O request.

If the -v flag is used, then utilization is based on the virtual processor.

id

(Default, -a flag, -v flag) The percentage of time that the logical processor was idle and it did not have an outstanding disk I/O request.

If the -v flag is used, then utilization is based on the virtual processor.

pc

(Default, -a flag, -h flag, -v flag) The number or fraction of physical processor consumed. It is displayed in both a shared partition and a dedicated partition. For the default flag in the dedicated partition, it is not displayed when both donation and simultaneous multithreading are disabled.

The pc of the cpuid U row represents the number of unused physical processors.

%ec

(Default, -a flag) The percentage of entitled capacity consumed by the logical processor. The %ec of the ALL CPU row represents the percentage of entitled capacity consumed. 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. The attribute is displayed only in a shared partition.

ilcs

(-a, -d, -h flag) Number of involuntary logical processor context switches, displayed only in shared partition. For the -h and -a flags, it is also displayed in dedicated partition.

vlcs

(-a, -d, -h flag) Number of voluntary logical processor context switches. Displayed only in shared partition. For the -h and -a flags, it is also displayed in dedicated partition.

lcs

(Default) Total number of logical processor context switches. Displayed only in shared partition or when a dedicated partition is donating.

%idon

(-a, -h flag) Shows the percentage of physical processor utilization that occurs while explicitly donating idle cycles. Displayed only in dedicated partition that is donating.

%bdon

(-a, -h flag) Shows the percentage of physical processor utilization that occurs while donating busy cycles. Displayed only in dedicated partition that is donating.

%istol

(-a, -h flag) Shows the percentage of physical processor utilization that occurs while the Hypervisor is stealing idle cycles. Displayed only in dedicated partition.

%bstol

(-a, -h flag) Shows the percentage of physical processor utilization that occurs while the Hypervisor is stealing busy cycles. Displayed only in dedicated partition.

%nsp

(-a, -d flag) Shows the current average processor speed as a percentage of nominal speed. Displayed only if the hardware uses Scaled Processor Utilization Resource Register (SPURR).

vcpu

(-v flag) Shows the virtual processor id.

lcpu

(-v flag) Shows the logical processor id.

pbusy

(-v, -E flag) Shows the percentage of physical time during which the physical processor is busy.

VTB

(-v flag) Shows the time taken for a virtual processor in milliseconds.

physc

(-E flag) Shows the number of physical processors that are used by the logical processor.

freq

(-E flag) Shows the operating frequency in GHz.

scaled physc

(-E flag) Shows the number of physical processors that are used by the logical processor based on SPURR.

The `mpstat` command displays all the above statistics for every logical processor in the partition. When running a shared partition, a special processor row with the `cpu id U` can be displayed when the entitled processing capacity has not entirely been consumed.

The `mpstat` command also displays a special processor row with the `cpu id ALL` which shows the partition-wide utilization. On that row, except for uncapped partitions with current physical processor consumption above their entitled capacity, the percentages are relative to the entitled processing capacity. For dedicated partitions, the entitled processing capacity is the number of physical processors.

And for a WPAR, the processors present in the associated resource set, if any, are indicated by asterisks (*) only when the **-s** or **-@** flag is used.

When the **-s** flag is specified, the **mpstat** command reports simultaneous multithreading utilization, if it is enabled. This report displays the virtual processor engines utilization and utilization of each thread (logical processor) associated with the virtual processor engine.

If **mpstat** is running in a dedicated partition and simultaneous multithreading is enabled, then only the thread (logical processor) utilization is displayed.

If **mpstat** is running on an interval based mode, then it would be average value calculated per second.

Flags

Item	Description
-a	Displays all the statistics.
-d	Displays detailed affinity and migration statistics for AIX threads and dispatching statistics for logical processors.
-i	Displays detailed interrupts statistics.
-s	Displays simultaneous multithreading threads utilization, this flag is available only when mpstat runs in a simultaneous multithreading enabled partition.
-h	Displays pc and processor switches, with stolen and donation statistics for dedicated partitions.
-w	Displays wide column output, switches to wide output mode. Default is 80 column output mode.
@ <i>wparname</i>	Displays the statistics for the specified WPAR.
-O <i>Options</i>	Specifies the command option. -O options=value... Following are the supported options: <ul style="list-style-type: none">• sortcolumn = Name of the metrics in the mpstat command output• sortorder = [asc desc]• topcount = Number of CPUs to be displayed in the mpstat command sorted output
-X	Generates the XML output. The default file name is mpstat_DDMMYYHHMM.xml unless you specify a different file name by using with the -o option.
-o	Specifies the file name for the XML output.
-v	Displays utilization statistics at the virtual processor level. Note: The -v flag is available only for POWER8® processors, and later.
-E	Displays SPURR-based utilization metrics on a SPURR-capable processor.

Note:

1. The **-a**, **-d**, and **-i** flags implicitly turn on wide-column output.
2. Inside a WPAR, the **-@** flag reports statistics of all processors.
3. Processor statistics that are displayed inside a WPAR is always system wide.
4. Only **-o** option is allowed with **-X** option.

Parameters

Item	Description
<i>interval</i>	Specifies the interval between the iterations. If <i>interval</i> is not specified, just one snapshot of metrics is displayed which actually reports the values from the time system is up. If <i>interval</i> is specified, the tool waits for that duration before printing the first set of data. Each set of data is followed by a separation line, a line with average values for each columns (except the processor, which is replaced by ALL), followed by an empty line.
<i>count</i>	Specifies number of iterations. If <i>interval</i> is specified and <i>count</i> is not specified then <code>mpstat</code> runs infinitely. <i>count</i> can not be specified without specifying <i>interval</i> .

Examples

1. To see the default set of utilization metrics, enter the following command:

```
mpstat 1 1
```

2. To see the default set of utilization metrics in wide display mode, enter the following command:

```
mpstat -w 1 1
```

3. To see the detailed dispatch & affinity metrics, enter the following command:

```
mpstat -d 1 1
```

4. To see the detailed interrupts report, enter the following command:

```
mpstat -i 1 1
```

5. To see all the statistics, enter the following command:

```
mpstat -a 1 1
```

6. To see simultaneous multithreading utilization, enter the following command:

```
mpstat -s 1 1
```

7. To see all the processor metrics of a WPAR, enter the following command:

```
mpstat -@ wparname
```

Note: To see all the processor metrics of a WPAR inside the WPAR, enter the following command:

```
mpstat -@
```

8. To see the sorted output for the column **cs**, enter the following command:

```
mpstat -d -0 sortcolumn=cs
```

9. To see the list of the top 10 CPUs, enter the following command:

```
mpstat -a -0 sortcolumn=min,sortorder=desc,topcount=10
```

10. To see metrics based on the virtual processor, enter the following command:

```
mpstat -v
```

Files

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

mrouterd Daemon

Purpose

Forwards a multicast datagram.

Syntax

```
/usr/sbin/mrouterd [ -p ] [ -c Config_File ] [ -d [ Debug_Level ] ]
```

Description

The **mrouterd** daemon is an implementation of the Distance Vector Multicast Routing Protocol (DVMRP), an earlier version of which is specified in RFC 1075. It maintains topological knowledge using a distance vector routing protocol (like RIP, described in RFC 1058), on which it implements a multicast datagram forwarding algorithm called Reverse Path Multicasting.

The **mrouterd** daemon forwards a multicast datagram along a shortest (reverse) path tree rooted at the subnet on which the datagram originates. The multicast delivery tree may be thought of as a broadcast delivery tree that has been pruned back so that it does not extend beyond those subnetworks that have members of the destination group. Hence, datagrams are not forwarded along those branches that have no listeners of the multicast group. The IP time-to-live of a multicast datagram can be used to limit the range of multicast datagrams.

To support multicasting among subnets that are separated by (unicast) routers that do not support IP multicasting, the **mrouterd** daemon includes support for tunnels, which are virtual point-to-point links between pairs of the **mrouterd** daemons located anywhere in an Internet. IP multicast packets are encapsulated for transmission through tunnels, so that they look like typical unicast datagrams to intervening routers and subnets. The encapsulation is added on entry to a tunnel, and stripped off on exit from a tunnel. By default, the packets are encapsulated using the IP-in-IP protocol (IP protocol number 4). Older versions of the **mrouterd** tunnel use IP source routing, which puts a heavy load on some types of routers. This version does not support IP source-route tunneling.

The tunneling mechanism allows the **mrouterd** daemon to establish a virtual Internet, for the purpose of multicasting only, which is independent of the physical Internet and which may span multiple Autonomous Systems. This capability is intended for experimental support of Internet multicasting only, pending widespread support for multicast routing by the regular (unicast) routers. The **mrouterd** daemon suffers from the well-known scaling problems of any distance-vector routing protocol and does not support hierarchical multicast routing.

The **mrouterd** daemon automatically configures itself to forward on all multicast-capable interfaces (that is, interfaces that have the IFF_MULTICAST flag set, excluding the loopback interface), and it finds other **mrouterd** daemons directly reachable using those interfaces.

The **mrouterd** daemon does not initiate execution if it has fewer than two enabled virtual interfaces, where a virtual interface (Vif) is either a physical multicast-capable interface or a tunnel. It logs a warning if all of its virtual interfaces are tunnels; such an **mrouterd** daemon's configuration would be better replaced by more direct tunnels.

The **mrouterd** daemon handles multicast routing only; there might be unicast-routing software running on the same machine as the **mrouterd** daemon. With the use of tunnels, it is unnecessary for the **mrouterd** daemon to have access to more than one physical subnet to perform multicast forwarding.

Flags

Item	Description
-c <i>Config_File</i>	Starts the mROUTED command using an alternate configuration file specified by the <i>Config_File</i> variable.

There are five types of configuration entries:

```
phyint local-addr [disable] [metric m] [threshold t] [rate_limit b]
[boundary (boundary-name|scoped-addr/mask-len)] [altnet
network/mask-len]

tunnel local-addr remote-addr
[
metric m
] [
threshold t
] [
rate_limit b
]

[
boundary
(
boundary-name
|
scoped-addr
/
mask-len
)]

cache_lifetime ct

pruning off
|
on

name boundary-name scoped-addr
/
mask-len
```

-d	Sets the debug level. If no -d option is given, or if the debug level is specified as 0, the mROUTED daemon detaches from the invoking terminal. Otherwise, it remains attached to the invoking terminal and responsive to signals from that terminal. If -d is given with no argument, the debug level defaults to 2. Regardless of the debug level, the mROUTED daemon always writes warning and error messages to the system log demon. Non-zero debug levels have the following effects:
-----------	--

level 1

All syslog'ed messages are also printed to **stderr**.

level 2

All level 1 messages plus notifications of significant events are printed to **stderr**.

level 3

All level 2 messages plus notifications of all packet arrivals and departures are printed to **stderr**.

Upon startup, the **mROUTED** daemon writes its pid to the file **/etc/mROUTED.pid**.

-p	Turns off pruning. Default is pruning enabled.
-----------	--

Signals

The following signals can be sent to the **mROUTED** daemon:

Item Description

- HUP** Restarts the **mROUTED** daemon. The configuration file is reread every time this signal is evoked.
- INT** Terminates execution gracefully (that is, by sending good-bye messages to all neighboring routers).
- TER** Same as **INT**.
- M**
- USR1** Dumps the internal routing tables to **/usr/tmp/mROUTED.dump**.
- 1**
- USR2** Dumps the internal cache tables to **/usr/tmp/mROUTED.cache**.
- 2**
- QUIT** Dumps the internal routing tables to **stderr** (if the **mROUTED** daemon was invoked with a nonzero debug level).

For convenience in sending signals, the **mROUTED** daemon writes its pid to **/etc/mROUTED.pid** on startup.

Examples

1. To display routing table information, type:

```
kill -USR1 `cat /etc/mROUTED.pid`
```

This produces the following output:

```
Virtual Interface Table
Vif Local-Address      Metric  Thresh  Flags
0 36.2.0.8      subnet: 36.2          1      1      querier
   groups: 224.0.2.1
           224.0.0.4
   pkts in: 3456
   pkts out: 2322323
1 36.11.0.1     subnet: 36.11         1      1      querier
   groups: 224.0.2.1
           224.0.1.0
           224.0.0.4
   pkts in: 345
   pkts out: 3456
2 36.2.0.8     tunnel: 36.8.0.77     3      1
   peers: 36.8.0.77 (2.2)
   boundaries: 239.0.1
               : 239.1.2
   pkts in: 34545433
   pkts out: 234342
3 36.2.0.8     tunnel: 36.6.8.23     3      16

Multicast Routing Table (1136 entries)
Origin-Subnet  From-Gateway  Metric  Tmr  In-Vif  Out-Vifs
36.2           1 45 0 1* 2 3*
36.8           36.8.0.77    4 15 2 0* 1* 3*
36.11          1 20 1 0* 2 3*
.
.
.
```

In this example, there are four virtual interfaces connecting to two subnets and two tunnels. The Vif 3 tunnel is not in use (no peer address). The Vif 0 and Vif 1 subnets have some groups present; tunnels never have any groups. This instance of the **mROUTED** daemon is the one responsible for sending periodic group membership queries on the Vif 0 and Vif 1 subnets, as indicated by the **querier** flags. The list of boundaries indicate the scoped addresses on that interface. A count of the no. of incoming and outgoing packets is also shown at each interface.

Associated with each subnet from which a multicast datagram can originate is the address of the previous hop router (unless the subnet is directly connected), the metric of the path back to the origin,

the amount of time since an update for this subnet was last received, the incoming virtual interface for multicasts from that origin, and a list of outgoing virtual interfaces. The * (asterisk) means that the outgoing virtual interface is connected to a leaf of the broadcast tree rooted at the origin, and a multicast datagram from that origin will be forwarded on that outgoing virtual interface only if there are members of the destination group on that leaf.

The **mrouted** daemon also maintains a copy of the kernel forwarding cache table. Entries are created and deleted by the **mrouted** daemon.

2. To display cache table information, type:

```
kill -USR2 *cat /etc/mrouted.pid*
```

This produces the following output:

```
Multicast Routing Cache Table (147 entries)
Origin      Mcast-group  CTmr  Age  Ptmr  IVif  Forwvifs
13.2.116/22 224.2.127.255 3m    2m   -     0     1
>13.2.116.19
>13.2.116.196
138.96.48/21 224.2.127.255 5m    2m   -     0     1
>138.96.48.108
128.9.160/20 224.2.127.255 3m    2m   -     0     1
>128.9.160.45
198.106.194/24 224.2.135.190 9m    28s  9m    0P
>198.106.194.22
```

Each entry is characterized by the origin subnet number and mask and the destination multicast group. The **CTmr** field indicates the lifetime of the entry. The entry is deleted from the cache table when the timer decrements to zero. The **Age** field is the time since this cache entry was originally created. Because cache entries get refreshed if traffic is flowing, routing entries can grow very old. The **Ptmr** field is a hyphen if no prune was sent upstream or the amount of time until the upstream prune will time out. The **IVif** field indicates the incoming virtual interface for multicast packets from that origin. Each router also maintains a record of the number of prunes received from neighboring routers for a particular source and group. If there are no members of a multicast group on any downward link of the multicast tree for a subnet, a prune message is sent to the upstream router. They are indicated by a P after the virtual interface number. The **Forwvifs** field shows the interfaces along which datagrams belonging to the source group are forwarded. A p indicates that no datagrams are being forwarded along that interface. An unlisted interface is a leaf subnet with are no members of the particular group on that subnet. A b on an interface indicates that it is a boundary interface, that is, traffic will not be forwarded on the scoped address on that interface. An additional line with a > (greater-than sign) as the first character is printed for each source on the subnet. There can be many sources in one subnet.

Files

Item	Description
/etc/mrouted.conf	Contains the configuration information for the mrouted daemon.
/usr/tmp/mrouted.dump	Contains the internal routing tables for the mrouted daemon.
/etc/mrouted.pid	Contains the process ID for the mrouted daemon.
/usr/tmp/mrouted.cache	Contains the internal cache tables for the mrouted daemon.

msgchk Command

Purpose

Checks for messages.

Syntax

`msgchk [User ...]`

Description

The **msgchk** command checks mail drops for messages. The **msgchk** command reports whether the mail drop for the specified user contains messages and indicates if the user has already seen these messages. By default, the **msgchk** command checks the mail drop for the current user.

Flags

Item	Description
------	-------------

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

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

Examples

1. To check to see if you have any new messages, enter:

```
msgchk
```

If you have new messages, the system responds with a message similar to the following:

```
You have new Internet mail waiting
```

If you have no messages, the system responds with a message similar to the following:

```
You don't have any mail waiting
```

2. To check to see if user karen on your local system has any new messages, enter:

```
msgchk karen
```

In this example, if user karen on your local system has new messages, the system responds with a message similar to the following:

```
karen has new Internet mail waiting
```

If user karen on your local system has no messages, the system responds with a message similar to the following:

```
karen doesn't have any mail waiting
```

Files

Item	Description
\$HOME/.mh_profile	Contains the user's MH profile.
/etc/mh/mtstailor	Contains the MH tailor file.
/var/spool/Mail/\$USER	Defines the location of the mail drop.
/usr/bin/msgchk	Contains the msgchk command.

msh Command

Purpose

Creates a Message Handler (MH) shell.

Syntax

```
msh [ File ] [ -prompt String ] [ -notopcur | -topcur ]
```

Description

The **msh** command creates an MH shell for use with messages that are packed in a file. By default, this command looks for the **msgbox** file in the current directory. Within the MH shell, you can use the following MH commands:

ali	burst	comp	dist
folder	forw	inc	mark
mhmail	msgchk	next	packf
pick	prev	refile	repl
rmm	scan	send	show
sortm	whatnow	whom	

These commands operate with limited functionality in the MH shell. To see how a command operates in the MH shell, enter the command name followed by the **-help** flag. Entering **help** or a ? (question mark) displays a list of the MH commands you can use.

To leave the **msh** shell, press the Ctrl-D key sequence or enter **quit**.

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.
-notopcur	Makes the current message track the center line of the vmh scan window when the msh command is started using the vmh command. This flag is the default.
-prompt <i>String</i>	Prompts for the msh commands with the specified string.
-topcur	Makes the current message track the top line of the vmh scan window when the msh command is started using the vmh command.

Profile Entries

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

Item	Description
fileproc :	Specifies the program used to refile messages.
Msg-Protect :	Sets the protection level for your new message files.
Path :	Specifies the user's MH directory.
showproc :	Specifies the program used to show 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

1. To start an **msh** shell, enter:

```
msh
```

If the **msgbox** file exists in the current directory, the system responds with a message similar to the following:

```
Reading ./msgbox, currently at message 1 of 10
```

Then, the system prompt appears as follows:

```
(msh)
```

In this example, the current message is message 1 in the **msgbox** file. You can now enter a modified subset of MH commands.

2. To start an **msh** shell to manipulate the messages stored in the **meetings** file, enter:

```
msh meetings
```

Files

Item	Description
\$HOME/.mh_profile	Specifies the user's MH profile.
/etc/mh/mtstailor	Contains the MH tailor file.
/usr/bin/msh	Contains the msh command.

mt Command (BSD)

Purpose

Gives subcommands to streaming tape device.

Syntax

```
mt [ -f TapeName ] Subcommand [ Count ]
```

Description

The **mt** command gives subcommands to a streaming tape device. If you do not specify the **-f** flag with the *TapeName* parameter, the **TAPE** environment variable is used. If the environment variable does not exist, the **mt** command uses the **/dev/rmt0.1** device. The *TapeName* parameter must be a raw (not block) tape device. You can specify more than one operation with the *Count* parameter.

Subcommands

Item	Description
eof, weof	Writes the number of end-of-file markers specified by the <i>Count</i> parameter at the current position on the tape.
fsf	Moves the tape forward the number of files specified by the <i>Count</i> parameter and positions it to the beginning of the next file.

Item	Description
bsf	Moves the tape backwards the number of files specified by the <i>Count</i> parameter and positions it to the beginning of the last file skipped. If using the bsf subcommand would cause the tape head to move back past the beginning of the tape, then the tape will be rewound, and the mt command will return 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.
rewoff1, rewind	Rewinds the tape. The <i>Count</i> parameter is ignored.
status	Prints status information about the specified tape device. The output of the status command may change in future implementations.

Flag

Item	Description
-f <i>TapeName</i>	Specifies the <i>TapeName</i> parameter.

Examples

1. To rewind the `rmt1` tape device, enter:

```
mt -f /dev/rmt1 rewind
```

2. To move forward two files on the default tape device, enter:

```
mt fsf 2
```

3. To write two end-of-file markers on the tape in the `/dev/rmt0.6` file, enter:

```
mt -f /dev/rmt0.6 weof 2
```

Exit Status

Item	Description
0	Indicates a successful completion.
>0	Indicates an error occurred.

Files

Item	Description
<code>/dev/rmt/n.n</code>	Specifies the raw streaming tape interface.
<code>/usr/bin/mt</code>	Contains the mt command file.

mtrace Command

Purpose

Prints a multicast path from a source to a receiver.

Syntax

```
mtrace [ -l ] [ -M ] [ -n ] [ -p ] [ -s ] [ -U ] [ -g gateway ] [ -i if_addr ] [ -m max_hops ] [ -q nqueries ] [ -r resp_dest ] [ -S statint ] [ -t ttl ] [ -w wait ] source [ receiver ] [ group ]
```

Description

A trace query is passed hop-by-hop along the path from the receiver to the source, collecting hop addresses, packet counts, and routing error conditions along the path, and then the response is returned to the requestor. The default receiver is the host running the **mtrace** command, and the default group is 0.0.0.0.

Note: The **mtrace** command is intended for use in network testing, measurement, and management. Because the **mtrace** command heavily loads on the network, avoid using the **mtrace** command during typical operations or from automated scripts. It should be used primarily or with manual fault isolation. If the **-g** flag is specified, the source defaults to the host running **mtrace** and the receiver defaults to the router being addressed.

By default, the **mtrace** command first attempts to trace the full reverse path, unless the number of hops to trace is explicitly set with the **-m** flag. If there is no response within a 3-second timeout interval (changed with the **-w** flag), an * (asterisk) is printed and the probing switches to hop-by-hop mode. Trace queries are issued starting with a maximum hop count of one and increasing by one until the full path is traced or no response is received. At each hop, multiple probes are sent (default is three, changed with **-q** flag). The first half of the attempts (default is two) are made with the reply address set to standard multicast address, `mtrace.mcast.net` (224.0.1.32) with the *ttl* set to 32 more than what is needed to pass the thresholds seen so far along the path to the receiver. For each additional attempt, the *ttl* is increased by another 32 each time up to a maximum of 192. Because the desired router may not be able to send a multicast reply, the remainder of the attempts request that the response be sent via unicast to the host running the **mtrace** command.

Alternatively, the multicast *ttl* can be set explicitly with the **-t** flag, the initial multicast attempts can be forced to use unicast instead with the **-U** flag, the final unicast attempts can be forced to use multicast instead with the **-M** flag, or if you specify **-UM**, the **mtrace** command will first attempt using unicast and then multicast. For each attempt, if no response is received within the timeout, an * (asterisk) is printed. After the specified number of attempts have failed, the **mtrace** command will try to query the next hop router with a **DVMRP_ASK_NEIGHBORS2** request to see what kind of router it is. The **mtrace** command will try to query three (changed with the **-e** flag) hops past a non-responding router. Even though the **mtrace** command is incapable of sending a response, it might be capable of forwarding the request.

Flags

Item	Description
-g gateway	Sends the trace query via unicast directly to the multicast router <i>gateway</i> rather than multicasting the query. This must be the last-hop router on the path from the intended source to the receiver.
-i if_addr	Uses <i>if_addr</i> as the local interface address (on a multi-homed host) for sending the trace query and as the default for the receiver and the response destination.
-l	Loops indefinitely printing packet rate and loss statistics for the multicast path every 10 seconds (see -S stat_int).
-m max_hops	Sets <i>max_hops</i> to the maximum number of hops that will be traced from the receiver to the source. The default is 32 hops and infinity for the DVMRP routing protocol).

Item	Description
-M	Always requests the response using multicast rather than attempting unicast for the last half of the tries.
-n	Prints hop addresses numerically rather than symbolically and numerically (saves a name server address-to-name lookup for each router found on the path).
-p	Listens passively for multicast responses from traces initiated by others. This works best when run on a multicast router.
-q nqueries	Sets the maximum number of query attempts for any hop to <i>nqueries</i> . The default is 3.
-r resp_dest	Sends the trace response to <i>dhost</i> rather than to the host on which the mtrace command is being run, or to a multicast address other than the one registered for this purpose (224.0.1.32).
-s	Prints a short form output including only the multicast path and not the packet rate and loss statistics.
-S statint	Changes the interval between statistics gathering traces to <i>statint</i> seconds (default 10 seconds).
-t ttl	Sets the <i>ttl</i> (time-to-live, or number of hops) for multicast trace queries and responses. The default is 127, except for local queries to the <code>all_routers</code> multicast group that use the <i>ttl</i> of 1.
-U	Forces initial multicast attempts to use unicast instead.
-w wait	Sets the time to wait for a trace response to <i>wait</i> seconds (default 3 seconds).

Parameters

Item	Description
<i>source</i>	Specifies the host for which the multicast path from a particular receiver is sought. This is a required parameter.
<i>receiver</i>	Specifies the host from which the multicast path is sought for a particular source. Default is the host in which the mtrace command is running. This is an optional parameter.
<i>group</i>	Specifies the multicast group. This is an optional parameter.

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 example, the two machines, 10.27.41.57 and 10.27.40.20, are on two different subnets separated by a router having two interfaces, 10.27.40.11 and 10.27.41.11. To find the multicast path from 10.27.40.20 to 10.27.41.57, type the following:

```
# mtrace 10.27.41.57 224.2.0.1
```

The following is displayed:

```
Mtrace from 10.27.41.57 to 10.27.40.20 via group 224.2.0.1
Querying full reverse path...
 0 ? (10.27.40.20)
-1 ? (10.27.40.11) DVMRP thresh^ 1
```

```

-2 ? (10.27.41.57)
Round trip time 1 ms; total ttl of 2 required.

Waiting to accumulate statistics... Results after 10 seconds:

  Source      Response Dest    Overall   Packet Statistics For Traffic From
10.27.41.57   224.0.1.32  Packet    10.27.41.57 To 224.2.0.1
    v         --/  rtt    1 ms    Rate    Lost/Sent = Pct  Rate
10.27.41.11
10.27.40.11   ?
    v         \--  ttl    2        0 pps
10.27.40.20   10.27.40.20
Receiver      Query Source

```

multibos Command

Purpose

Creates, updates, and manages multiple versions of the Base Operating System (BOS) on a rootvg.

Syntax

```

multibos -s [-l device {-a | -f file | -b file | -x file}] [-e file] [-i file] [-L file] [-pntNX]
multibos -c -l device {-a | -f file | -b file | -x file} [-pnNX]
multibos -m [-pnX]
multibos -u [-pnX]
multibos -B [-ntX]
multibos -S [-nX]
multibos -R [-ptX]
multibos -C [VG name]
multibos -s -M file [-pntNX]

```

Description

>| The `multibos` command allows the root user to create an instance of the Base Operating System (BOS) in the `rootvg` volume group. |<

Note: You cannot use the `multibos` command with a mirrored `rootvg` volume group.

>| The `multibos` setup operation creates a standby Base Operating System (BOS) that boots from a distinct boot logical volume (BLV). |< This creates two bootable sets of BOS on a given `rootvg`. The administrator can boot from either instance of BOS by specifying the respective BLV as an argument to the `bootlist` command or using system firmware boot operations. Two bootable instances of BOS can be simultaneously maintained. The instance of BOS associated with the booted BLV is referred to as the *active* BOS. The instance of BOS associated with the BLV that has not been booted is referred to as the *standby* BOS. Currently, only two instances of BOS are supported per `rootvg`.

The `multibos` command allows the administrator to access, install maintenance and technology levels for, update, and customize the standby BOS either during setup or in subsequent customization operations. Installing maintenance and technology updates to the standby BOS does not change system files on the active BOS. This allows for concurrent update of the standby BOS, while the active BOS remains in production.

In addition, the `multibos` command copies or shares logical volumes and file systems. By default, the BOS file systems (currently `/`, `/usr`, `/var`, and `/opt`), and the boot logical volume are copied. The administrator can make copies of additional BOS objects (using the `-L` flag).

All other file systems and logical volumes are shared between instances of BOS. Separate log device logical volumes (for example, those that are not contained within the file system) are not supported for copy and are shared.

In AIX 5L Version 5.3 with the 5300-09 Technology Level, you can populate the standby instance with a later version such as AIX Version 6.1 with the 6100-02 Technology Level. The function is done by creating a **mksysb** backup of a system at a later version and then using the backup to populate the standby instance. For example, system A is at a level of 5.3.9.0 and system B is at a level of 6.1.2.0. You can create a backup of system B using the `mksysb -M` command and use the **mksysb** backup to populate a standby instance of the operating system on system A.

Notes:

1. While the **multibos** command is running, ensure that system activity is minimal.
2. Any logical volume or file system attributes that are new to the higher level are not implemented when the standby instance is created because the operating system at a lower level has no knowledge of the attributes.
3. Do not keep both an AIX 6.1 instance and an AIX 5.3 instance for an extended period. You might not be able to switch between the instances because of incompatibilities. Commit to one of the instances and remove the other.
4. If your operating system is running with the logical volumes in the active BOS that have the multibos-created `bos_hd*` names, and no standby BOS in the `rootvg` directory, then a preservation or migration type of installation can occur starting with AIX 7200-00. The logical volumes that have the `bos_hd*` names on the system are `bos_hd5`, `bos_hd4`, `bos_hd2`, `bos_hd9var`, and `bos_hd10opt`. The operating system must not have `hd5`, `hd4`, `hd2`, `hd9var`, or `hd10opt` logical volumes. If you created the multibos instance from an **mksysb** image, which was created by using the **mksysb** command with the **-M** flag, the `hd8` logical volume might also have been renamed to `bos_hd8`. You can check this prerequisite with the `lsvg -l rootvg` command. Always back up your system before migrating. Also, copy the `/usr/lpp/bos/pre_migration` file from the media or your network installation manager (NIM) spot of the level to which you are migrating, to the target system and execute the file on the target system to check for any migration warnings.

Before you perform a migration or a preservation type of operating system installation in this environment, verify that the disk control block has a valid level for your `rootvg`. You can run the `/usr/lpp/bosinst/blvset -d /dev/hdiskN -g level` command, where `hdiskN` is the disk that contains the `bos_hd5` logical volume. If this command returns `0.0` or an unexpected level, run the `bosboot -ad /dev/ipldevice` command to correct it, and rerun the `blvset` command to verify. It must return `6.1` or `7.1`.

5. In addition to the flags mentioned in the syntax section, the **-V** flag performs the verify operation from the `inittab` during boot. It is important that you do not modify this entry. The verify operation enables the **multibos** utility to synchronize changes in logical volumes and file systems between the active and standby instances. This entry also synchronizes the ODM and devices on initial boot after a **mksysb** restore. Without this operation, both the active and standby instances could become inconsistent with normal file system and logical volume operations.

The file system types (JFS or JFS2) of the **mksysb** backup need to be the same as that of the system where the **multibos** command is to be run. For example, if the `/usr` file system is a JFS2 file system, the `/usr` file system on the **mksysb** backup needs to be a JFS2 file system.

A log is stored in the `/etc/multibos/logs/op.alog` file after you run the **multibos** command. You can view the log file using the `alog -f /etc/multibos/logs/op.alog -o` command.

Note: You can create a backup that contains both instances by first mounting the standby instance (using the `-m` flag), and then creating the backup. However, you can restore the backup onto a disk only by using the `alt_disk_mksysb` command.

Restrictions

- The `multibos` command is supported on AIX 5L Version 5.3 with the 5300-03 Recommended Maintenance package and later.

- The current `rootvg` must have enough space for each BOS object copy. BOS object copies are placed on the same disk or disks as the original.
- The total number of copied logical volumes cannot exceed 128. The total number of copied logical volumes and shared logical volumes are subject to volume group limits.
- >| Using the **multibos** command in a mirrored `rootvg` volume group is not supported. You can remove the mirror of the `rootvg` volume group by running the **unmirrorvg** command. You can run the **alt_disk_copy** command to have a copy of the `rootvg` volume group that can be upgraded as needed. When you have the required instance of AIX, remove the volume group no longer needed by using the `alt_rootvg_op` operation, and re-mirror the `rootvg` volume group. |<

Flags

Item	Description
-a	Specifies the <code>update_all</code> install option. Valid with <code>setup</code> and <code>customization</code> operation.
-B	Build boot image operation. The standby boot image is created and written to the standby BLV using the AIX <code>bosboot</code> command.
-b <i>file</i>	Specifies the install bundle to be installed during the <code>setup</code> or <code>customization</code> operation. The install bundle syntax should follow <code>geninstall</code> conventions.
-c	Performs a customized update of the software in standby BOS.
-C <i>VG name</i>	Allows you to vary on volume groups for syncing when it is appropriate, as the auto varied off volume groups are not varied on during the reboot, like the <code>rootvg</code> volume group, in order to be synced. The <code>multibos -C VG name</code> command should be used on auto varied off volume groups after the <code>multibos</code> command has created an alternative root volume group (operating system) on the disk and has been booted.
-e <i>file</i>	Lists active BOS files to be excluded during the <code>setup</code> operation in regular expression syntax.
-f <i>file</i>	Lists fixes (such as APARs) that are to be installed during the <code>setup</code> or <code>customization</code> operation. The syntax of the list follows <code>instfix</code> conventions.
-i <i>file</i>	Specifies optional <code>image.data</code> file to use instead of the default <code>image.data</code> file created from the current <code>rootvg</code> .
-L <i>file</i>	Specifies a file that contains a list of additional logical volumes to include in standby BOS.
-l <i>device</i>	Installs <code>device</code> or <code>directory</code> for software update during the <code>setup</code> or <code>customization</code> operation.
-m	Mounts standby BOS.
-M <i>file</i>	Specifies a file that contains a mksysb image. The mksysb image must have been created using the <code>mksysb -M</code> command beginning with AIX 6.1 with 6100-02.
-N	Skips boot image processing. This flag should only be used by experienced administrators that have a good understanding of the AIX boot process.

Item	Description
-n	Does not perform cleanup upon failure. This option is useful to retain multibos data after a failed operation.
-p	Performs a preview of the given operation. Valid with setup, remove, mount, unmount, and customization operations.
-R	Removes all standby BOS objects.
-S	Initiates an interactive shell with chroot access to the standby BOS file systems.
-s	Creates an instance of standby BOS.
-t	Prevents multibos from changing the bootlist.
-u	Unmounts standby BOS.
-x <i>file</i>	Runs the optional customization script before any other customized parameters, such as, update_all (-a), install bundle file (-b), and fix list file (-f). You must use a full pathname for the script.
-X	Allows for automatic file system expansion if space is needed to perform tasks related to multibos. It is recommended that all multibos operations are executed with this flag.

Exit Status

Item	Description
0	All the multibos command operations completed successfully.
>0	An error occurred.

Security

Only the root user can run the multibos command.

Examples

1. To perform a standby BOS setup operation preview, enter the following command:

```
multibos -Xsp
```

2. To set up standby BOS, enter the following command:

```
multibos -Xs
```

3. To set up standby BOS with optional image.data file /tmp/image.data and exclude list /tmp/exclude.list, enter the following command:

```
multibos -Xs -i /tmp/image.data -e /tmp/exclude.list
```

4. To set up standby BOS and install additional software listed as bundle file /tmp/bundle and located in the images source /images, enter the following command:

```
multibos -Xs -b /tmp/bundle -l /images
```

5. To execute a customization operation on standby BOS with the `update_all` install option, enter the following command:

```
multibos -Xac -l /images
```

6. To mount all standby BOS file systems, enter the following command:

```
multibos -Xm
```

7. To perform a standby BOS remove operation preview, enter the following command:

```
multibos -RXp
```

8. To remove standby BOS, enter the following command:

```
multibos -RX
```

9. To use an existing **mksysb** file **/backups/mksysb1** to populate the standby instance of rootvg, enter the following command:

```
multibos -M /backups/mksysb1 -sX
```

Files

Item	Description
<code>/usr/sbin/multibos</code>	Contains the multibos command.
<code>/etc/multibos</code>	Contains multibos data and logs.

mv Command

Purpose

Moves files.

Syntax

To Move and Rename a File

```
mv [-d] [-e] [-E{force|ignore|warn}] [-i] [-f] [-I] SourceFile ... TargetFile
```

To Move and Rename a Directory

```
mv [-d] [-e] [-E{force|ignore|warn}] [-i] [-f] [-I] SourceDirectory ... TargetDirectory
```

To Move Files or Directories to a Directory Maintaining Original File Names

```
mv [-d] [-e] [-E{force|ignore|warn}] [-i] [-f] [-I] SourceFile/SourceDirectory TargetDirectory
```

Description



Attention: The **mv** command can overwrite many existing files unless you specify the **-i** flag. The **-i** flag prompts you to confirm before it overwrites a file. If both the **-f** and **-i** flags are specified in combination, the last flag specified takes precedence.

The **mv** command moves files and directories from one directory to another or renames a file or directory. If you move a file or directory to a new directory, it retains the base file name. When you move a file, all links to other files remain intact, except when you move it to a different file system. When you move a directory into an existing directory, the directory and its contents are added under the existing directory.

When you use the **mv** command to rename a file or directory, the *TargetDirectory* parameter can specify either a new file name or a new directory path name.

If moving the file would overwrite an existing file that does not have write-permission set and if standard input is a workstation, the **mv** command displays the file-permission code and reads a line from standard input. If that line begins with a y or the locale's equivalent of a y, the **mv** command moves the file. If the response is anything other than a y, the **mv** command does nothing to that file and continues with the next specified file. The file-permission code displayed may not fully represent the access permission if the *TargetFile* is associated with an ACL. When the parent directory of the *SourceFile* is writable and has the sticky bit set, one or more of the following conditions are true:

- The user must own the file.
- The user must own the directory
- The user must be a privileged user.
- The file must be writable by the user.

This warning message and prompt for input can be overridden by using the **-f** option.

You can use the **mv** command to move files within the same file system or between file systems. Whether you are working in one file system or across file systems, the **mv** command copies the file to the target and deletes the original file. The **mv** command preserves in the new file the time of the most recent data modification, the time of the most recent access, the user ID, the group ID, the file mode, the extended attributes, and ACLs of the original file. For symbolic links, the **mv** command preserves only the owner and group of the link itself.

If it is unable to preserve the owner and group ID, the **mv** command clears S_ISUID and S_ISGID bits in the target. The **mv** command prints a diagnostic message to stderr if it is unable to clear these bits, though the exit code is not affected.

The **mv** command modifies either the source file or the destination path if the command is prematurely terminated.

Note: The **mv** command supports the **--** (dash, dash) parameter as a delimiter that indicates the end of the flags.

The **mv** command will not move an object if the object is exported as an NFS version 4 referral. The referral object is marked as busy and remains so until it is unexported.

Note: The I/O buffer size for the read and write system calls generated by this command can be configured by using the **AIX_STDBUFSZ** environment variable.

Flags

Attention: The **mv** command can overwrite many existing files unless you specify the **-i** flag. The **-i** flag prompts you to confirm before it overwrites a file. If both the **-f** and **-i** flags are specified in combination, the last flag specified takes precedence.

It	Description
-----------	--------------------

m	
----------	--

-d	The source file is stored in decrypted (clear-text) format on target.
-----------	---

-e	The source file is stored in encrypted form, if the target file system is an Encrypted File System (EFS).
-----------	---

Item Description

-E The **-E** option requires one of the following arguments. If you omit the **-E** option, **warn** is the default behavior.

force

Fails the **mv** 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 extent size of the file cannot be preserved.

-f Does not prompt you before overwriting an existing file.

-i Prompts you before moving a file or directory to an existing path name by displaying the name of the file followed by a question mark. If you answer with a line starting with y or the locale's equivalent of a y, the move continues. Any other reply prevents the move from occurring.

-I Suppresses the warning message during ACL conversion.

The following table shows the encryption or decryption status of the target file under different conditions:

Explicit flag for the mv command	Source file	Target file system	Result
-e (encrypted)	Non-EFS	Non-EFS	Error
-e	Non-EFS	EFS	Encrypted file
-e	EFS	EFS	Encrypted file
-e	EFS	Non-EFS	Error
-d (decrypted)	Non-EFS	Non-EFS	Clear-text file
-d	Non-EFS	EFS	Clear-text file
-d	EFS	Non-EFS	Clear-text file
-d	EFS	EFS	Clear-text file
No explicit flag	Non-EFS	Non-EFS	Clear-text file
No explicit flag	Non-EFS	EFS	If the target file system is EFS activated, the target file is an encrypted file. Else, the target file is a clear-text file.
No explicit flag	EFS	EFS	Encrypted file
No explicit flag	EFS	Non-EFS	Error

Note: It is not permitted to overwrite an encrypted file with a plain-text file and vice versa unless you specify the **-f** flag. The encryption status of the target depends on the **-e** or **-d** flag, the encryption inheritance if you do not specify the **-e** or **-d** flag with the **-f** flag, and the encryption status of the source file if the encryption inheritance is not active.

Examples

1. To rename a file, enter:

```
mv appendix apndx.a
```

This command renames `appendix` to `apndx.a`. If a file named `apndx.a` already exists, its old contents are replaced with those of `appendix`.

2. To move a directory, enter:

```
mv book manual
```

This command moves all files and directories under `book` to the directory named `manual`, if `manual` exists. Otherwise, the directory `book` is renamed `manual`.

3. To move a file to another directory and give it a new name, enter:

```
mv intro manual/chap1
```

This command moves `intro` to `manual/chap1`. The name `intro` is removed from the current directory, and the same file appears as `chap1` in the directory `manual`.

4. To move a file to another directory, keeping the same name, enter:

```
mv chap3 manual
```

This command moves `chap3` to `manual/chap3`

Note: Examples 1 and 3 name two files, example 2 names two existing directories, and example 4 names a file and a directory.

5. To move several files into another directory, enter:

```
mv chap4 jim/chap5 /home/manual
```

This command moves the `chap4` file to the `/home/manual/chap4` file directory and the `jim/chap5` file to the `/home/manual/chap5` file.

6. To use the **mv** command with pattern-matching characters, enter:

```
mv manual/* .
```

This command moves all files in the `manual` directory into the current directory `.` (period), retaining the names they had in `manual`. This move also empties `manual`. You must type a space between the asterisk and the period.

Note: Pattern-matching characters expand names of existing files only. For example, the command `mv intro man*/chap1` does not work if the file `manual/chap1` does not exist.

Exit Status

Itc Description

m

0 All input files were moved successfully.

>0 An error occurred.

Files

Item

Description

`/usr/bin/mv`

Contains the **mv** command.

mvdір Command

Purpose

Moves (renames) a directory.

Syntax

mvdір *Directory1* *Directory2*

Description

The **mvdір** command renames directories within a file system. To use the **mvdір** command, you must have write permission to *Directory1* and *Directory2* as well as in the parent directories.

The *Directory1* parameter must name an existing directory. If *Directory2* does not exist, *Directory1* is moved to *Directory2*. If *Directory2* exists, *Directory1* becomes a subdirectory of *Directory2*. Neither directory can be a subset of the other.

The **mvdір** Command can also be used to move or rename files. If the *Directory1* parameter is an existing file name and the *Directory2* parameter is an existing directory name, the file specified by *Directory1* is moved to the directory specified by *Directory2*. If the *Directory1* parameter is an existing file name and the *Directory2* parameter does not yet exist, *Directory2* replaces the file name *Directory1*. If both are existing file names, the file specified by *Directory1* is renamed *Directory2*, and the existing *Directory2* is removed.

The **mv** command provides the same functionality as the **mvdір** command.

The **mvdір** command will not rename a directory if the directory is exported for use by NFS version 4, or if the directory leads to a directory exported for use by NFS version 4. NFS version 4-exported directories and directories leading to NFS version 4-exported directories are marked as busy and remain so until unexported.

Example

To rename or move a directory to another location, enter:

```
mvdір appendixes manual
```

If `manual` does not exist, this renames the `appendixes` directory to `manual`.

If a directory named `manual` already exists, this moves `appendixes` and its contents to `manual/appendixes`. In other words, `appendixes` becomes a subdirectory of `manual`.

Files

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

mvfilt Command

Purpose

Moves a filter rule.

Syntax

```
mvfilt -v 4|6 -p p_fid -n n_fid
```

Description

Use the **mvfilt** command to change the position of a filter rule in the filter rule table. IPsec filter rules for this command can be configured using the **genfilt** command or IPsec **smit** (IP version 4 or IP version 6).

Flags

Item	Description
-v	IP version of the filter rule. The value 4 specifies IP version 4 and the value 6 specifies IP version 6.
-p	Filter rule ID. It specifies the previous position of the rule in the filter rule table. For IP version 4, the value of 1 is invalid since the first filter rule is unmoveable.
-n	Filter rule ID. It specifies the new position of the rule in the filter rule table after the move. For IP version 4, the value of 1 is invalid since the first filter rule is reserved and thus is unmoveable.

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.

mvmt Command

Purpose

Typesets English-language view graphs and slides.

Syntax

```
mvmt [-a] [-c] [-e] [-g] [-p] [-t] [-z] [-TName | -DDestination] [File ... | -]
```

Description

The **mvmt** command typesets its input with the **mv** macro package for view graphs and slides in a manner similar to the **mmt** command. The **mvmt** command has flags to specify preprocessing by the **tbl**, **eqn**, **pic**, **cw**, and **grap** commands. The flags you select determine which pipelines, flags, and parameters are generated for the **troff** command and the macro package.

The **mvmt** command, unlike the **troff** command, automatically pipes its output to a postprocessor, unless specifically requested not to do so. The user should not specify a postprocessor when using the **mvmt** command. The path that the **mvmt** command takes is as follows:

1. The **-z** flag (no postprocessor is used).
2. The **-TName** flag.
3. The **TYPESETTER** environment variable is read.
4. The default is set to **ibm3816**.

File specifies the file that the **mvt** command formats.

Flags

Flags can occur in any order, but they must be displayed before the *File* parameter. If no file is specified, the **mvt** command prints a list of its flags.

Item	Description
-a	Displays readable troff output to the terminal.
-c	Calls the cw command.
-e	Calls the eqn command; also causes the eqn command to read the /usr/share/lib/pub/eqnchar file.
-g	Calls the grap command, which in turn calls the pic command.
-p	Calls the pic command.
-t	Calls the tbl command.
-z	Calls no output filter (or postprocessor) to process or redirect the output of the troff command.
-D<i>Destination</i>	Directs the output to the specified device destination. Supported value for the <i>Destination</i> variable is 4014 , which is the Tektronix 4014 terminal by way of the tc command.
-T<i>Name</i>	Creates output for the troff device as specified by the <i>Name</i> variable. The output is sent through the appropriate postprocessor. The default is ibm3816 .
-	Forces input to be read from standard input.

Any other parameters or flags that you give the **mvt** command (such as the **-a** flag) are passed to the **troff** command.

The **mvt** command reads standard input when you specify the **-** (minus) flag instead of the *File* parameter.

Use the **-o*List*** flag of the **troff** command to specify ranges of pages to be output.

Note: If you call the **mvt** command with one or more of the **-e**, **-c**, **-t**, **-p**, **-g**, or **-** flags, together with the **-o*List*** flag of the **troff** command, you may receive a broken pipe message. This occurs if you do not specify the last page of the document in the *List* variable. This broken pipe message is not an indication of any problem and can be ignored.

Environment Variables

Item	Description
TYPESETTER	Contains information about a particular printing device.

Files

Item	Description
/usr/share/lib/pub/eqnchar	Contains special character definitions.

mwm Command

Purpose

Runs the AIXwindows Window Manager (MWM).

Syntax

mwm -display *Host:Display:ScreenID* **-xrm** *ResourceString* **-multiscreen** **-name** *Name* **-screens** *Name*
[*Name ...*]

Description

The **mwm** command runs the AIXwindows Window Manager (MWM) and is often started by a display or session manager. The AIXwindows Window Manager (MWM) is an X Window System client that provides window management functionality and some session management functionality. It provides functions that facilitate control (by the user and the programmer) of elements of window states such as placement, size, icon or normal display, and input-focus ownership. It also provides session management functions such as stopping a client.

The appearance and behavior of the window manager can be altered by changing the configuration of specific resources. Resources are defined under [X Defaults](#) .

By default, the **mwm** command manages only the single screen specified by the **-display** option or the **DISPLAY** environment variable (by default, screen 0). If the **-multiscreen** option is specified or if the **multiScreen** resource is True, the **mwm** command tries to manage all the screens on the display.

When the **mwm** command is managing multiple screens, the **-screens** option can be used to give each screen a unique resource name. The names are separated by blanks, for example, **-screens mwm0 mwm1**. If there are more screens than names, resources for the remaining screens are retrieved using the first name. By default, the screen number is used for the screen name.

For information on windows, icons, resources, events, button and key bindings, menus, and variables, see the following sections:

- [Windows](#)
- [Icons](#)
- [Icon Box](#)
- [Component Appearance Resources](#)
- [General Appearance and Behavior Resources](#)
- [Client-Specific Resources](#)
- [Window Manager Event Specification](#)
- [Button Bindings](#)
- [Key Bindings](#)
- [Menu Panes](#)
- [Environment](#)

Flags

Flag Description

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.
-xrm <i>ResourceString</i>	Enables the named resources when starting the mwm command.
-multiscreen	Causes the mwm command to manage all screens on the display. The default is to manage only a single screen.
-name <i>Name</i>	Causes the mwm command to retrieve its resources using the specified name, as in <i>Name*Resource</i> .
-screens <i>Name [Name [...]]</i>	Specifies the resource names to use for the screens managed by MWM. If MWM is managing a single screen, only the first name in the list is used. If multiple screens are being managed, the names are assigned to the screens in order, starting with screen 0. For example, screen 0 gets the first name and screen 1 gets the second name.

Windows

Default window manager window frames have the following distinct components with associated functions:

Windows Description

Item	Description
title area	In addition to displaying the client's title, the title area is used to move the window. To move the window, place the pointer over the title area, press button 1 and drag the window to a new location. A wire frame is moved during the drag to indicate the new location. When the button is released, the window is moved to the new location.
title bar	The title bar includes the title area, the Minimize button, the Maximize button, and the Window Menu button. In shaped windows, such as round windows, the title bar floats above the window.
Minimize button	To turn the window into an icon, click button 1 on the Minimize button (the frame box with a small square in it).
Maximize button	To make the window fill the screen (or enlarge to the largest size allowed by the configuration files), click button 1 on the Maximize button (the frame box with a large square in it).

Item	Description
Window Menu button	The Window Menu button is the frame box with a horizontal bar in it. To pull down the window menu, press button 1. While pressing the button, drag the pointer on the menu to your selection and release the button when your selection is highlighted. Pressing button 3 in the title bar or resize border handles also posts the window menu. Alternately, you can click button 1 to pull down the menu and keep it posted; then position the pointer and select. You can also post the window menu by pressing the Shift+Esc or Alt+Space key sequence. Double-clicking button 1 with the pointer on the Window Menu button closes the window. The following table lists the contents of the window menu:

Default Window Menu

Selection	Accelerator	Description
Restore	Alt+F5	Restores the window to its size before minimizing or maximizing.
Move	Alt+F7	Allows the window to be moved with keys or mouse.
Size	Alt+F8	Allows the window to be resized.
Minimize	Alt+F9	Turns the window into an icon.
Maximize	Alt+F10	Makes the window fill the screen.
Lower	Alt+F3	Moves window to bottom of window stack.
Close	Alt+F4	Causes client to stop.

resize border handles

To change the size of a window, move the pointer over a resize border handle (the cursor changes), press button 1, and drag the window to a new size. When the button is released, the window is resized. While dragging is being done, a rubber-band outline is displayed to indicate the new window size.

matte

An optional matte decoration can be added between the client area and the window frame. A matte is not actually part of the window frame. There is no functionality associated with a matte.

Icons

Icons are small graphic representations of windows. A window can be iconified (minimized) using the **Minimize** button on the window frame. Icons provide a way to reduce clutter on the screen.

Pressing the left mouse button when the pointer is over an icon causes the icon's window menu to open. Releasing the button (press + release without moving mouse = click) causes the menu to stay posted. The menu contains the following selections:

Icon Window Menu		
Selection	Accelerator	Description
Restore	Alt+F5	Opens the associated window.
Move	Alt+F7	Allows the icon to be moved with keys.
Size	Alt+F8	Inactive (not an option for icons).
Minimize	Alt+F9	Inactive (not an option for icons).
Maximize	Alt+F10	Opens the associated window and makes it fill the screen.
Lower	Alt+F3	Moves icon to bottom of icon stack.
Close	Alt+F4	Removes client from window manager management.

Pressing button 3 over an icon also causes the icon's window menu to open. To make a menu selection, drag the pointer over the menu and release button 3 when the desired item is highlighted.

Double-clicking button 1 on an icon calls the **f.restore_and_raise** function and restores the icon's associated window to its previous state. For example, if a maximized window is iconified, double-clicking button 1 restores it to its maximized state. Double-clicking button 1 on the icon box's icon opens the icon box and allow access to the contained icons. (Double-clicking a mouse button is a quick way to perform a function.) Pressing the Shift+Esc key sequence or the pop-up Menu key causes the icon window menu of the currently selected icon to open.

Icon Box

When icons begin to clutter the screen, they can be packed into an icon box. (To use an icon box, the window manager must be started with the icon box configuration already set.) The icon box is a window manager window that holds client icons. It includes one or more scroll bars when there are more window icons than the icon box can show at the same time.

Icons in the icon box can be manipulated with the mouse. The following button action descriptions summarize the behavior of this interface. Button actions apply whenever the pointer is on any part of the icon. Double-clicking an icon in the icon box calls the **f.restore_and_raise** function.

Icon Box

Button Action	Description
Button 1 click	Selects the icon.
Button 1 double-click	Normalizes (opens) the associated window.
Button 1 double-click	Raises an already <i>open</i> window to the top of the stack.
Button 1 drag	Moves the icon.
Button 3 press	Causes the menu for that icon to open.
Button 3 drag	Highlights items as the pointer moves across the menu.

Pressing mouse button 3 when the pointer is over an icon causes the menu for that icon to open.

Icon Menu for Icon Box		
Selection	Accelerator	Description
Restore	Alt+F5	Opens the associated window (if not already open).
Move	Alt+F7	Allows the icon to be moved with keys.
Size	Alt+F8	Inactive.
Minimize	Alt+F9	Inactive.
Maximize	Alt+F10	Opens the associated window (if not already open) and maximizes its size.
Lower	Alt+F3	Inactive.
Close	Alt+F4	Removes client from window manager management.

To pull down the window menu for the icon box itself, press button 1 with the pointer over the menu button for the icon box. The window menu of the icon box differs from the window menu of a client window: The **Close** selection is replaced with the **PackIcons** (Shift+Alt+F7) selection. When selected, the **PackIcons** option packs the icons in the box to achieve neat rows with no empty slots.

You can also post the window menu by pressing the Shift+Esc or Alt+Space key sequence. Pressing the pop-up Menu key causes the icon window menu of the currently selected icon to open.

Input Focus

The **mwm** command supports (by default) a keyboard input focus policy of *explicit selection*. This means when a window is selected to get keyboard input, it continues to get keyboard input until the window is withdrawn from window management, another window is explicitly selected to get keyboard input, or the window is iconified. Several resources control the input focus. The client window with the keyboard input focus has the active window appearance with a visually distinct window frame.

The following table and key action descriptions summarize the keyboard input focus selection behavior:

Input focus		
Button Action	Object	Function Description
Button 1 press	Window or window frame	Keyboard focus selection
Button 1 press	Icon	Keyboard focus selection

Function Description

Key Action	Function Description
Alt+Tab	Moves the input focus to next window in the window stack.
Alt+Shift+Tab	Moves the input focus to the previous window in the window stack (available only in explicit focus mode).

Window Stacking

There are two types of window stacks: global window stacks and an application's local family window stack.

The global stacking order of windows can be changed as a result of setting the keyboard input focus, iconifying a window, or performing a window manager window stacking function. When keyboard focus policy is explicit the default value of the **focusAutoRaise** resource is True. This causes a window to be raised to the top of the stack when it receives input focus, for example, by pressing button 1 on the title bar. The key actions defined in the preceding list raises the window receiving focus to the top of the stack.

In pointer mode, the default value of the **focusAutoRaise** resource is False; that is, the window stacking order is not changed when a window receives keyboard input focus. The following key actions can be used to cycle through the global window stack:

Windows Stacking function description

Key Action	Function Description
Alt+Esc	Places top window on bottom of stack.
Alt+Shift+Esc	Places bottom window on top of stack.

By default, a window's icon is placed on the bottom of the stack when the window is iconified; however, the default can be changed by the **lowerOnIconify** resource.

Transient windows (secondary windows such as dialog boxes) stay above their parent windows by default. However, an application's local family stacking order can be changed to allow a transient window to be placed below its parent top-level window. The following parameter values show the modification of the stacking order for the **f.lower** function:

Item	Description
f.lower	Lowers the transient window within the family (staying above the parent) and lowers the family in the global window stack.
f.lower [within]	Lowers the transient window within the family (staying above the parent) but does not lower the family in the global window stack.
f.lower [freeFamily]	Lowers the window separate from its family stack (below the parent), but does not lower the family in the global window stack.

The **within** and **freeFamily** parameter values can also be used with the **f.raise** and **f.raise_lower** functions.

X Defaults

The **mwm** command is configured from its resource database. This database is built from the following sources. They are listed in order of precedence.

1. **mwm** command line options
2. **XENVIRONMENT** variable or **\$HOME/.Xdefaults-host**
3. **RESOURCE_MANAGER** root window property or **\$HOME/.Xdefaults**
4. **\$HOME/Mwm**
5. **/usr/lib/X11/app-defaults/Mwm**.

The **/usr/lib/X11/app-defaults/Mwm** and **\$HOME/Mwm** file names represent customary locations for these files. The actual location of the systemwide class resource file might depend on the **XFILESEARCHPATH** environment variable and the current language environment. The actual location of the user-specific class resource file might depend on the **XUSERFILESEARCHPATH** and **XAPPLRESDIR** environment variables and the current language environment.

Entries in the resource database can refer to other resource files for specific types of resources. These include files that contain bitmaps, fonts, and **mwm**-specific resources such as menus and behavior specifications (for example, button and key bindings).

Mwm is the resource class name of the **mwm** command and **mwm** is the resource name used by the **mwm** command to look up resources. (For looking up resources of multiple screens, the **-screens** command-line option specifies resource names such as **mwm_b+w** and **mwm_color**.) In the following discussion of resource specification, "Mwm" and "mwm" (and the aliased **mwm** resource names) can be used interchangeably, but "mwm" takes precedence over "Mwm". The **mwm** command uses the following types of resources:

X Defaults description

Item	Description
<u>component appearance resource set</u>	These resources specify appearance attributes of window manager user-interface components. They can be applied to the appearance of window manager menus, feedback windows (for example, the window reconfiguration feedback window), client window frames, and icons.
<u>frame and icon component resource set</u>	This subset of component appearance resources specifies attributes that are specific to frame and icon components.
<u>general appearance and behavior resource set</u>	These resources specify the mwm command appearance and behavior (for example, window management policies). They are not set separately for different mwm command user-interface components.

Item	Description
<u>client-specific resource set</u>	These mwm resources can be set for a particular client window or class of client windows. They specify client-specific icon and client window frame appearance and behavior.

Resource identifiers can be either a resource name (for example, **foreground**) or a resource class (for example, **Foreground**). If the value of a resource is a file name and if the file name is prefixed by the ~/ (tilde followed by a slash) characters, it is relative to the path contained in the **HOME** environment variable (generally the user's home directory).

Component Appearance Resources

The syntax for specifying component appearance resources that apply to window manager icons, menus, and client window frames is as follows:

Mwm*ResourceID

For example, **Mwm*foreground** is used to specify the foreground color for the **mwm** command menus, icons, client window frames, and feedback dialogs.

The syntax for specifying component appearance resources that apply to a particular **mwm** component is as follows:

Mwm*[Menu|Icon|Client|Feedback]*ResourceID

If *Menu* is specified, the resource is applied only to **Mwm** menus; if *Icon* is specified, the resource is applied to icons; and if *Client* is specified, the resource is applied to client window frames. For example, **Mwm*Icon*foreground** is used to specify the foreground color for the **mwm** command icons, **Mwm*Menu*foreground** specifies the foreground color for the **mwm** command menus, and **Mwm*Client*foreground** is used to specify the foreground color for the **mwm** command client window frames.

The appearance of the title area of a client window frame (including window management buttons) can be separately configured. The syntax for configuring the title area of a client window frame is as follows:

Mwm*Client*Title*ResourceID

For example, **Mwm*Client*Title*foreground** specifies the foreground color for the title area. Defaults for title area resources are based on the values of the corresponding client window frame resources.

The appearance of menus can be configured based on the name of the menu. The syntax for specifying menu appearance by name is as follows:

Mwm*Menu*MenuName*ResourceID

For example, **Mwm*Menu*MyMenu*foreground** specifies the foreground color for the menu named **MyMenu**.

The user can also specify resources for window manager menu components (the gadgets that comprise the menu). These may include, for example, a menu title, a title separator, one or more buttons, and separators. If a menu contains more than one instance of a class, such as multiple **PushButtonGadget** gadgets, the name of the first instance is **PushButtonGadget1**, the second is **PushButtonGadget2**, and so on. The following list identifies the naming conventions used for window manager menu components:

Component Appearance Resources

Item	Description
TitleName	Menu title LabelGadget
TitleSeparator	Menu title SeparatorGadget

Component Appearance Resources (*continued*)

Item	Description
CascadeButtonGadget <i>n</i>	CascadeButtonGadget
PushButtonGadget <i>n</i>	PushButtonGadget
SeparatorGadget <i>n</i>	SeparatorGadget

The following component appearance resources that apply to all window manager parts can be specified.

Component Appearance Resource Set

Component Appearance Resource Set	
Name	Properties
background	Class Background Value type color Default varies ¹
backgroundPixmap	Class BackgroundPixmap Value type string ² Default varies ¹
bottomShadowColor	Class Foreground Value type color Default varies ¹
bottomShadowPixmap	Class BottomShadowPixmap Value type string ² Default varies ¹
fontList	Class FontList Value type string ³ Default "fixed"

Component Appearance Resource Set (continued)

Name	Properties
foreground	<p>Class Foreground</p> <p>Value type color</p> <p>Default varies¹</p>
saveUnder	<p>Class SaveUnder</p> <p>Value type True or False</p> <p>Default False</p>
topShadowColor	<p>Class Background</p> <p>Value type color</p> <p>Default varies¹</p>
topShadowPixmap	<p>Class TopShadowPixmap</p> <p>Value type string²</p> <p>Default varies¹</p>
background	<p>Class Background</p> <p>Value type color</p> <p>Default varies¹</p>
backgroundPixmap	<p>Class BackgroundPixmap</p> <p>Value type string²</p> <p>Default varies¹</p>
bottomShadowColor	<p>Class Foreground</p> <p>Value type color</p> <p>Default varies¹</p>

Component Appearance Resource Set (<i>continued</i>)	
Name	Properties
bottomShadowPixmap	Class BottomShadowPixmap Value type string ² Default varies ¹
fontList	Class FontList Value type string ³ Default "fixed"
foreground	Class Foreground Value type color Default varies ¹
saveUnder	Class SaveUnder Value type True or False Default False
topShadowColor	Class Background Value type color Default varies ¹
topShadowPixmap	Class TopShadowPixmap Value type string ² Default varies ¹

Note:

1. The default is chosen based on the visual type of the screen.
2. Image name.
3. X Version 11 Release 4 (X11R4) font description.

Item	Description
background (class Background)	Specifies the background color. Any legal X color can be specified. The default value is chosen based on the visual type of the screen.
backgroundPixmap (class BackgroundPixmap)	Specifies the background pixmap of the mwm decoration when the window is inactive (does not have the keyboard focus). The default value is chosen based on the visual type of the screen.
bottomShadowColor (class Foreground)	Specifies the bottom shadow color. This color is used for the lower and right bevels of the window manager decoration. Any legal X color can be specified. The default value is chosen based on the visual type of the screen.
bottomShadowPixmap (class BottomShadowPixmap)	Specifies the bottom shadow pixmap. This pixmap is used for the lower and right bevels of the window manager decoration. The default is chosen based on the visual type of the screen.
fontList (class FontList)	Specifies the font used in the window manager decoration. The character encoding of the font needs to match the character encoding of the strings that are used. The default is the fixed value.
foreground (class Foreground)	Specifies the foreground color. The default is chosen based on the visual type of the screen.
saveUnder (class SaveUnder)	Controls the repainting of windows that are uncovered after being obscured. This resource indicates whether <i>save unders</i> are used for mwm components. For this to have any effect, save unders must be implemented by the X server. If save unders are implemented, the X server saves the contents of windows obscured by windows that have the save under attribute set. If the saveUnder resource is True, the mwm command sets the save under attribute on the window manager frame of any client that has it set. If the saveUnder resource is False, save unders is not used on any window manager frames. The default value is False.
topShadowColor (class Background)	Specifies the top shadow color. This color is used for the upper and left bevels of the window manager decoration. The default is chosen based on the visual type of the screen.
topShadowPixmap (class TopShadowPixmap)	Specifies the top shadow pixmap. This pixmap is used for the upper and left bevels of the window manager decoration. The default is chosen based on the visual type of the screen.

Frame and Icon Component Resource Set

Note: Hyphens in the following table are for readability purposes only. Do not include hyphens within names in programs.

Frame and Icon Component Resource Set	
Name	Properties
activeBackground	Class Background Value type color Default varies ¹
activeBackground-Pixmap	Class BackgroundPixmap Value type string ² Default varies ¹
activeBottomShadow-Color	Class Foreground Value type color Default varies ¹
activeBottomShadow-Pixmap	Class BottomShadow-Pixmap Value type string ² Default varies ¹
activeForeground	Class Foreground Value type color Default varies ¹
activeTopShadowColor	Class Background Value type color Default varies ¹
activeTopShadowPixmap	Class TopShadowPixmap Value type string ² Default varies ¹

Frame and Icon Component Resource Set (continued)

Name	Properties
activeBackground	<p>Class Background</p> <p>Value type color</p> <p>Default varies¹</p>
activeBackgroundPixmap	<p>Class BackgroundPixmap</p> <p>Value type string²</p> <p>Default varies¹</p>
activeBottomShadowColor	<p>Class Foreground</p> <p>Value type color</p> <p>Default varies¹</p>
activeBottomShadowPixmap	<p>Class BottomShadowPixmap</p> <p>Value type string²</p> <p>Default varies¹</p>
activeForeground	<p>Class Foreground</p> <p>Value type color</p> <p>Default varies¹</p>
activeTopShadowColor	<p>Class Background</p> <p>Value type color</p> <p>Default varies¹</p>
activeTopShadowPixmap	<p>Class TopShadowPixmap</p> <p>Value type string²</p> <p>Default varies¹</p>

Note:

1. The default is chosen based on the visual type of the screen.
2. Image name.

Background

Item	Description
activeBackground (class Background)	Specifies the background color of the mwm decoration when the window is active (has the keyboard focus). The default is chosen based on the visual type of the screen.
activeBackgroundPixmap (class BackgroundPixmap)	Specifies the background pixmap of the mwm decoration when the window is active (has the keyboard focus). The default is chosen based on the visual type of the screen.
activeBottomShadowColor (class Foreground)	Specifies the bottom shadow color of the mwm decoration when the window is active (has the keyboard focus). The default is chosen based on the visual type of the screen.
activeBottomShadowPixmap (class BottomShadowPixmap)	Specifies the bottom shadow pixmap of the mwm decoration when the window is active (has the keyboard focus). The default is chosen based on the visual type of the screen.
activeForeground (class Foreground)	Specifies the foreground color of the mwm decoration when the window is active (has the keyboard focus). The default is chosen based on the visual type of the screen.
activeTopShadowColor (class Background)	Specifies the top shadow color of the mwm decoration when the window is active (has the keyboard focus). The default is chosen based on the visual type of the screen.
activeTopShadowPixmap (class TopShadowPixmap)	Specifies the top shadow pixmap of the mwm decoration when the window is active (has the keyboard focus). The default is chosen based on the visual type of the screen.

General Appearance and Behavior Resources

The syntax for specifying general appearance and behavior resources is as follows:

Mwm*ResourceID

For example, **Mwm*keyboardFocusPolicy** specifies the window manager policy for setting the keyboard focus to a particular client window.

General Appearance and Behavior Resource Set

Note: Hyphens in the following table are for readability purposes only. Do not include hyphens within names in programs.

General Appearance and Behavior Resource Set

Name	Properties
autoKeyFocus	<p>Class AutoKeyFocus</p> <p>Value type True or False</p> <p>Default True</p>
autoRaiseDelay	<p>Class AutoRaiseDelay</p> <p>Value type millisec.</p> <p>Default 500</p>
bitmapDirectory	<p>Class BitmapDirectory</p> <p>Value type directory</p> <p>Default /usr/include/X11/bitmaps</p>
buttonBindings	<p>Class ButtonBindings</p> <p>Value type string</p> <p>Default "DefaultButton-Bindings"</p>
cleanText	<p>Class CleanText</p> <p>Value type True or False</p> <p>Default True</p>
clientAutoPlace	<p>Class ClientAutoPlace</p> <p>Value type True or False</p> <p>Default True</p>
colormapFocusPolicy	<p>Class ColormapFocus-Policy</p> <p>Value type string</p> <p>Default keyboard</p>

General Appearance and Behavior Resource Set *(continued)*

Name	Properties
configFile	<p>Class ConfigFile</p> <p>Value type file</p> <p>Default .mwmrc</p>
defaultSetBehavior	<p>Class defaultSetBehavior</p> <p>Value type True or False</p> <p>Default True</p>
deiconifyKeyFocus	<p>Class DeiconifyKeyFocus</p> <p>Value type True or False</p> <p>Default True</p>
doubleClickTime	<p>Class DoubleClickTime</p> <p>Value type milliseconds</p> <p>Default multiclick time</p>
enableWarp	<p>Class enableWarp</p> <p>Value type True or False</p> <p>Default True</p>
enforceKeyFocus	<p>Class EnforceKeyFocus</p> <p>Value type True or False</p> <p>Default True</p>
fadeNormalIcon	<p>Class FadeNormalIcon</p> <p>Value type True or False</p> <p>Default False</p>

General Appearance and Behavior Resource Set *(continued)*

Name	Properties
feedbackGeometry	<p>Class FeedbackGeometry</p> <p>Value type string</p> <p>Default center on screen</p>
frameBorderWidth	<p>Class FrameBorderWidth</p> <p>Value type pixels</p> <p>Default varies</p>
iconAutoPlace	<p>Class IconAutoPlace</p> <p>Value type True or False</p> <p>Default True</p>
iconBoxGeometry	<p>Class IconBoxGeometry</p> <p>Value type string</p> <p>Default 6x1+0-0</p>
iconBoxName	<p>Class IconBoxName</p> <p>Value type string</p> <p>Default iconbox</p>
iconBoxSBDisplayPolicy	<p>Class IconBoxSBDisplayPolicy</p> <p>Value type string</p> <p>Default all</p>
iconBoxTitle	<p>Class IconBoxTitle</p> <p>Value type XmString</p> <p>Default Icons</p>

General Appearance and Behavior Resource Set *(continued)*

Name	Properties
iconClick	<p>Class IconClick</p> <p>Value type True or False</p> <p>Default True</p>
iconDecoration	<p>Class IconDecoration</p> <p>Value type string</p> <p>Default varies</p>
iconImage-Maximum	<p>Class IconImage-Maximum</p> <p>Value type width x height</p> <p>Default 50x50</p>
iconImageMinimum	<p>Class IconImageMinimum</p> <p>Value type width x height</p> <p>Default 16x16</p>
iconPlacement	<p>Class IconPlacement</p> <p>Value type string</p> <p>Default left bottom</p>
iconPlacementMargin	<p>Class IconPlacementMargin</p> <p>Value type pixels</p> <p>Default varies</p>
interactivePlacement	<p>Class InteractivePlacement</p> <p>Value type True or False</p> <p>Default False</p>

General Appearance and Behavior Resource Set *(continued)*

Name	Properties
keyBindings	<p>Class KeyBindings</p> <p>Value type string</p> <p>Default "DefaultKeyBindings"</p>
keyboardFocusPolicy	<p>Class KeyboardFocusPolicy</p> <p>Value type string</p> <p>Default explicit</p>
limitResize	<p>Class LimitResize</p> <p>Value type True or False</p> <p>Default True</p>
lowerOnIconify	<p>Class LowerOnIconify</p> <p>Value type True or False</p> <p>Default True</p>

Maximum size

Item	Description
maximumMaximuSize	<p>Class MaximumMaximuSize</p> <p>Value type width x height (pixels)</p> <p>Default 2X screen width & height</p>
moveOpaque	<p>Class MoveOpaque</p> <p>Value type True or False</p> <p>Default False</p>

Maximum size <i>(continued)</i>	
Item	Description
moveThreshold	Class MoveThreshold Value type pixels Default 4
multiScreen	Class MultiScreen Value type True or False Default False
passButtons	Class PassButtons Value type True or False Default False
PassSelectButton	Class PassSelectButton Value type True or False Default True
positionIsFrame	Class PositionIsFrame Value type True or False Default True
positionOnScreen	Class PositionOnScreen Value type True or False Default True
quitTimeout	Class QuitTimeout Value type milliseconds Default 1000

Maximum size (continued)

Item	Description
raiseKeyFocus	<p>Class RaiseKeyFocus</p> <p>Value type True or False</p> <p>Default False</p>
resizeBorderWidth	<p>Class ResizeBorderWidth</p> <p>Value type pixels</p> <p>Default varies</p>
resizeCursors	<p>Class ResizeCursors</p> <p>Value type True or False</p> <p>Default True</p>
screens	<p>Class Screens</p> <p>Value type string</p> <p>Default varies</p>
showFeedback	<p>Class ShowFeedback</p> <p>Value type string</p> <p>Default all</p>
startupKeyFocus	<p>Class StartupKeyFocus</p> <p>Value type True or False</p> <p>Default True</p>
transientDecoration	<p>Class TransientDecoration</p> <p>Value type string</p> <p>Default menu title</p>

Maximum size (continued)	
Item	Description
transientFunctions	<p>Class TransientFunctions</p> <p>Value type string</p> <p>Default -minimize -maximize</p>
useIconBox	<p>Class UseIconBox</p> <p>Value type True or False</p> <p>Default False</p>
wMenuButtonClick	<p>Class WMenuButtonClick</p> <p>Value type True or False</p> <p>Default True</p>
wMenuButtonClick2	<p>Class WMenuButtonClick2</p> <p>Value type True or False</p> <p>Default True</p>

AutoKeyFocus

Item	Description
autoKeyFocus (class AutoKeyFocus)	<p>Controls whether the focus is set to the previous window that had the focus. If the autoKeyFocus resource is given a value of True and a window with the keyboard input focus is withdrawn from window management or is iconified, the focus is set to the previous window that had the focus. If the value given is False, there is no automatic setting of the keyboard input focus. It is recommended that both the autoKeyFocus resource and the startupKeyFocus resource be set to a value of True to work with tear-off menus. The default value is True. This resource is available only when the keyboard input focus policy is set to the explicit value.</p>
autoRaiseDelay (class AutoRaiseDelay)	<p>Specifies the amount of time in milliseconds (ms) that the mwm command waits before raising a window after it gets the keyboard focus. The default value of this resource is 500 (milliseconds). This resource is available only when the focusAutoRaise resource is True and the keyboard focus policy is the pointer value.</p>

Item	Description
bitmapDirectory (class BitmapDirectory)	Identifies a directory to be searched for bitmaps referenced by mwm resources. This directory is searched if a bitmap is specified without an absolute path name. The default value for this resource is /usr/include/X11/bitmaps . The /usr/include/X11/bitmaps directory represents the customary locations for this directory. The actual location of this directory may vary on some systems. If the bitmap is not found in the specified directory, the XBMLANGPATH environment variable is searched.
buttonBindings (class ButtonBindings)	Identifies the set of button bindings for window management functions. The named set of button bindings is specified in the mwm resource description file. These button bindings are merged with the built-in default bindings. The default value for this resource is DefaultButtonBindings .
cleanText (class CleanText)	Controls the display of window manager text in the client title and feedback windows. If the default value of True is used, the text is drawn with a clear (no stipple) background. This makes text easier to read on monochrome systems where a backgroundPixmap is specified. Only the stippling in the area immediately around the text is cleared. If False, the text is drawn directly on top of the existing background.
clientAutoPlace (class ClientAutoPlace)	Determines the position of a window when the window does not have a user-specified position. With a value of True, windows are positioned with the top left corners of the frames offset horizontally and vertically. A value of False causes the currently configured position of the window to be used. In either case, the mwm command attempts to place the windows totally on-screen. The default value is True.
colormapFocusPolicy (class ColormapFocusPolicy)	Indicates the colormap focus policy that is to be used. If the resource value is explicit, a colormap selection action is done on a client window to set the colormap focus to that window. If the value is pointer, the client window containing the pointer has the colormap focus. If the value is keyboard, the client window that has the keyboard input focus has the colormap focus. The default value for this resource is keyboard.

Item	Description
configFile (class ConfigFile)	<p>Contains the path name for an mwm resource description file.</p> <p>If the path name begins with the ~/ characters, the mwm command considers it to be relative to the user's home directory (as specified by the HOME environment variable). If the LANG environment variable is set, the mwm command looks for \$HOME/\$LANG/configFile. If that file does not exist or if LANG is not set, mwm looks for \$HOME/configFile.</p> <p>If the configFile path name does not begin with the ~/ characters, mwm considers it to be relative to the current working directory.</p> <p>If the configFile resource is not specified or if that file does not exist, the mwm command uses several default paths to find a configuration file. If the LANG environment variable is set, the mwm command looks for the configuration file first in the \$HOME/\$LANG/.mwmrc file. If that file does not exist or if the LANG environment variable is not set, the mwm command looks for the \$HOME/.mwmrc file. If the \$HOME/.mwmrc file does not exist and if the LANG environment variable is set, the mwm command next looks for the /usr/lib/X11/\$LANG/system.mwmrc file. If the /usr/lib/X11/\$LANG/system.mwmrc file does not exist or if the LANG environment variable is not set, the mwm command looks for /usr/lib/X11/system.mwmrc.</p>
defaultSetBehavior (class DefaultSetBehavior)	<p>Determines whether the mwm command automatically add key bindings to the f.set_behavior function (see .mwmrc file).</p> <p>If the value for the defaultSetBehavior resource is True (or On), regardless of the key bindings defined in the .mwmrc configuration file, Alt Ctrl<key>1 and Alt Shift Ctrl<Key>exclam is bound to the f.set_behavior function.</p> <p>If the value for the defaultSetBehavior resource is False (or Off), the f.set_behavior function is bound to a key specified in the .mwmrc configuration file. If no key bindings are specified in the configuration file, the mwm command uses the default key binding.</p> <p>The default value for the defaultSetBehavior resource is True (or On).</p>
deiconifyKeyFocus (class DeiconifyKeyFocus)	<p>Determines whether a window receives the keyboard input focus when it is deiconified (normalized). The default value is True. This resource applies only when the keyboard input focus policy is set to the explicit value.</p>
doubleClickTime (class DoubleClickTime)	<p>Sets the maximum time (in ms) between the clicks (button presses) that make up a double-click. The default value of this resource is the display's multiclick time.</p>

Item	Description
enableWarp (class EnableWarp)	Causes the mwm command to <i>warp</i> the pointer to the center of the selected window during keyboard-controlled resize and move operations. Setting the value to False causes the mwm command to leave the pointer at its original place on the screen unless the user explicitly moves it with the cursor keys or pointing device. The default value of this resource is True .
enforceKeyFocus (class EnforceKeyFocus)	Determines whether the keyboard input focus is always explicitly set to selected windows even if there is an indication that they are <i>globally active</i> input windows. (An example of a globally active window is a scroll bar that can be operated without setting the focus to that client.) If the resource is False , the keyboard input focus is not explicitly set to globally active windows. The default value is True .
fadeNormalIcon (class FadeNormalIcon)	Determines whether an icon is unavailable whenever it is normalized (its window is opened). The default value is False .
feedbackGeometry (class FeedbackGeometry)	Sets the position of the move and resize feedback window. If this resource is not specified, the default is to place the feedback window at the center of the screen. The value of the resource is a standard window geometry string with the following syntax: [=][{+-}XOffset{+-}YOffset]
frameBorderWidth (class FrameBorderWidth)	Specifies the width in pixels of a client window frame border without resize handles. The border width includes the three-dimensional (3-D) shadows. The default value is based on the size and resolution of the screen.
iconAutoPlace (class IconAutoPlace)	Indicates whether the window manager arranges icons in a particular area of the screen or places each icon where the window was when it was iconified. The True value indicates that icons are arranged in a particular area of the screen determined by the iconPlacement resource. The False value indicates that an icon is placed at the location of the window when it is iconified. The default is True .
iconBoxGeometry (class IconBoxGeometry)	Indicates the initial position and size of the icon box. The value of the resource is a standard window geometry string with the following syntax: [=][WidthxHeight][{+-}XOffset{+-}YOffset]
	If the offsets are not provided, the iconPlacement policy is used to determine the initial placement. The units for width and height are columns and rows.
	The actual screen size of the icon box window depends on the iconImageMaximum (size) and iconDecoration resources. The default value for size is (6 times iconWidth + padding) wide by (1 times iconHeight + padding) high. The default value of the location is +0 -0.
iconBoxName (class IconBoxName)	Specifies the name that is used to look up icon box resources. The default name is iconbox .

Item	Description
iconBoxSBDisplayPolicy (class IconBoxSBDisplayPolicy)	Specifies the scroll bar display policy of the window manager in the icon box. The resource has the following three possible values: all, vertical, and horizontal. The default value, all, causes both vertical and horizontal scroll bars always to be displayed. The vertical value causes a single vertical scroll bar to be displayed in the icon box and sets the orientation of the icon box to horizontal (regardless of the iconBoxGeometry specification). The horizontal value causes a single horizontal scroll bar to be displayed in the icon box and sets the orientation of the icon box to vertical (regardless of the iconBoxGeometry specification).
iconBoxTitle (class IconBoxTitle)	Specifies the name that is used in the title area of the icon box frame. The default value is Icons.
iconClick (class IconClick)	Specifies whether the system menu is posted and remains posted when an icon is clicked. The default value is True.
iconDecoration (class IconDecoration)	Specifies the general icon decoration. The resource value is label (only the label part is displayed) or image (only the image part is displayed) or label image (both the label and image parts are displayed). A value of activelabel can also be specified to get a label (not truncated to the width of the icon) when the icon is selected. The default icon decoration for icon box icons is that each icon has a label part and an image part (label image). The default icon decoration for standalone icons is that each icon has an active label part, a label part, and an image part (activelabel, label, and image).
iconImageMaximum (class IconImageMaximum)	Specifies the maximum size of the icon image. The resource value is <i>Width x Height</i> (for example, 64x64). The maximum supported size is 128x128. The default value of this resource is 50x50.
iconImageMinimum (class IconImageMinimum)	Specifies the minimum size of the icon image. The resource value is <i>Width x Height</i> (for example, 32x50). The minimum supported size is 16x16. The default value of this resource is 16x16.
iconPlacement (class IconPlacement)	Specifies the icon placement scheme to be used. The resource value has the following syntax: <i>PrimaryLayout SecondaryLayout [Tight]</i> The layout values are described as one of the following: top Lays out the icons from top to bottom. bottom Lays out the icons from bottom to top. left Lays out the icons from left to right. right Lays out the icons from right to left.

Item	Description
	<p>A horizontal (vertical) layout value must not be used for both the <i>PrimaryLayout</i> and the <i>SecondaryLayout</i> (for example, do not use top for the <i>PrimaryLayout</i> and bottom for the <i>SecondaryLayout</i>). The <i>PrimaryLayout</i> indicates at the time an icon placement is done whether the icon is placed in a row or a column and the direction of placement. The <i>SecondaryLayout</i> indicates where to place new rows or columns.</p>
	<p>For example, the top right value indicates that icons should be placed top to bottom on the screen and that columns should be added from right to left on the screen. The default placement is the left bottom value (icons are placed from left to right on the screen, with the first row on the bottom of the screen, and new rows added from the bottom of the screen to the top of the screen). A tight value places icons with zero spacing between icons. This value is useful for aesthetic reasons, as well as for terminals with small screens.</p>
	<p>The following is a list of options for iconPlacement values:</p>
	<p>Icon Placement Appropriate Scheme</p>
	<p>From left to right across the top of the screen, new rows below Left top</p>
	<p>From right to left across the top of the screen, new rows below Right top</p>
	<p>From left to right across the bottom of the screen, new rows above Left bottom</p>
	<p>From right to left across the bottom of the screen, new rows above Right bottom</p>
	<p>From bottom to top along the left of the screen, new columns to right Bottom left</p>
	<p>From bottom to top along the right of the screen, new columns to left Bottom right</p>
	<p>From top to bottom along the left of the screen, new columns to right Top left</p>
	<p>From top to bottom along the right of the screen, new columns to left Top right</p>

Item	Description
iconPlacementMargin (class IconPlacementMargin)	Sets the distance between the edge of the screen and the icons that are placed along the edge of the screen. The value should be greater than or equal to 0. A default value is used if the value specified is invalid. The default value for this resource is equal to the space between icons as they are placed on the screen (this space is based on maximizing the number of icons in each row and column).
interactivePlacement (class InteractivePlacement)	Controls the initial placement of new windows on the screen. If the value is True, the pointer shape changes before a new window is placed on the screen to indicate to the user that a position needs to be selected for the upper-left corner of the window. If the value is False, windows are placed according to the initial window configuration attributes. The default value of this resource is False.

Key Bindings

Item	Description
keyBindings (class KeyBindings)	Identifies the set of key bindings for window management functions. If specified, these key bindings replace the built-in default bindings. The named set of key bindings is specified in mwm resource description file. The default value for this resource is DefaultKeyBindings.
keyboardFocusPolicy (class KeyboardFocusPolicy)	Determines the keyboard focus policy. If set to the pointer value, the keyboard focus policy has the keyboard focus set to the client window that contains the pointer (the pointer could also be in the client window decoration that the mwm command adds). If set to the explicit value, the policy is to have the keyboard focus set to a client window when the user presses the left mouse button with the pointer on the client window or any part of the associated mwm decoration. The default value for this resource is explicit.
limitResize (class LimitResize)	Determines whether the user is allowed to resize a window to greater than the maximum size. If this resource is True, the user is not allowed to resize a window to greater than the maximum size. The default value for this resource is True.
lowerOnIconify (class LowerOnIconify)	Determines whether a window icon is displayed on the bottom of the window stack when the window is iconified (minimized). A value of False places the icon in the stacking order at the same place as its associated window. The default value of this resource is True.
maximumMaximumSize (class MaximumMaximumSize)	Limits the maximum size of a client window as set by the user or client. The resource value is <i>Width x Height</i> (for example, 1024x1024) where the width and height are in pixels. The default value of this resource is twice the screen width and height.
moveOpaque (class MoveOpaque)	Controls whether the actual window is moved or a rectangular outline of the window is moved. A default value of False displays a rectangular outline on move operations.

Item	Description
moveThreshold (class MoveThreshold)	Controls the sensitivity of dragging operations that move windows and icons. The value of this resource is the number of pixels that the locator is moved with a button down before the move operation is initiated. This is used to prevent window and icon movement when you click or double-click and there is unintentional pointer movement with the button down. The default value of this resource is 4 (pixels).
multiScreen (class MultiScreen)	Determines whether the mwm command manages all the screens on the display. If False , the mwm command manages only a single screen. The default value is False .
passButtons (class PassButtons)	Indicates whether button press events are passed to clients after they are used to do a window manager function in the client context. If the resource value is False , the button press is not passed to the client. If the value is True , the button press is passed to the client window. The window manager function is done in either case. The default value for this resource is False .
passSelectButton (class PassSelectButton)	Indicates whether to pass the select button press events to clients after they are used to do a window manager function in the client context. If the resource value is False , the button press is not passed to the client. If the value is True , the button press is passed to the client window. The window manager function is done in either case. The default value for this resource is True .
positionIsFrame (class PositionIsFrame)	Indicates how client window position information (from the WM_NORMAL_HINTS property and from configuration requests) is to be interpreted. If the resource value is True , the information is interpreted as the position of the MWM client window frame. If the value is False , it is interpreted as being the position of the client area of the window. The default value of this resource is True .
positionOnScreen (class PositionOnScreen)	Indicates that windows should initially be placed (if possible) so that they are not clipped by the edge of the screen (if the resource value is True). If a window is larger than the size of the screen, at least the upper-left corner of the window is on-screen. If the resource value is False , windows are placed in the requested position even if totally off-screen. The default value of this resource is True .
quitTimeout (class QuitTimeout)	Specifies the amount of time in milliseconds that the mwm command waits for a client to update the WM_COMMAND property after the mwm command has sent the WM_SAVE_YOURSELF message. This protocol is used only for those clients that have a WM_SAVE_YOURSELF atom and no WM_DELETE_WINDOW atom in the WM_PROTOCOLS client window property. The default value of this resource is 1000 (milliseconds). See the f.kill function for additional information.

Item	Description
raiseKeyFocus (class RaiseKeyFocus)	Specifies whether a window raised by means of the f.normalize_and_raise function also receives the input focus. The default value of this resource is False. This resource is available only when the keyboard input focus policy is set to the explicit value.
resizeBorderWidth (class ResizeBorderWidth)	Specifies the width (in pixels) of a client window frame border with resize handles. The specified border width includes the 3-D shadows. The default value is based on the size and resolution of the screen.
resizeCursors (class ResizeCursors)	Indicates whether the resize cursors are always displayed when the pointer is in the window size border. If True, the cursors are shown; otherwise, the window manager cursor is shown. The default value is True.
screens (class Screens)	Specifies the resource names to use for the screens managed by the mwm command. If the mwm command is managing a single screen, only the first name in the list is used. If the mwm command is managing multiple screens, the names are assigned to the screens in order, starting with screen 0. For example, screen 0 gets the first name and screen 1 gets the second name. Examples of default screen names are 0 and 1.

Item	Description
showFeedback (class ShowFeedback)	<p data-bbox="735 233 1468 359">Controls when feedback information is displayed. It controls both window position and size feedback during move or resize operations and initial client placement. It also controls window manager message and dialog boxes.</p> <p data-bbox="735 380 1468 695">The value for this resource is a list of names of the feedback options to be enabled or disabled; the names must be separated by a space. If an option is preceded by a minus sign, that option is excluded from the list. The sign of the first item in the list determines the initial set of options. If the sign of the first option is - (a minus sign), the mwm command assumes all options are present and starts subtracting from that set. If the sign of the first decoration is + (a plus sign) or is not specified, the mwm command starts with no options and builds up a list from the resource.</p> <p data-bbox="735 716 1333 743">The names of the feedback options are as follows:</p> <p data-bbox="735 764 1195 821">all Shows all feedback (default value).</p> <p data-bbox="735 842 1133 898">behavior Confirms the behavior switch.</p> <p data-bbox="735 919 1235 976">kill Confirms on receipt of the KILL signal.</p> <p data-bbox="735 997 1167 1054">move Shows position during the move.</p> <p data-bbox="735 1075 1019 1131">none Shows no feedback.</p> <p data-bbox="735 1152 1360 1209">placement Shows position and size during initial placement.</p> <p data-bbox="735 1230 1065 1287">quit Confirms quitting MWM.</p> <p data-bbox="735 1308 1078 1365">resize Shows size during resize.</p> <p data-bbox="735 1386 1053 1442">restart Confirms MWM restart.</p> <p data-bbox="735 1463 1411 1520">The following command line illustrates the syntax for the showFeedback resource:</p> <p data-bbox="735 1541 1446 1568">Mwm*showFeedback: placement resize behavior restart</p> <p data-bbox="735 1589 1468 1694">This resource specification provides feedback for initial client placement and resize, and it enables the dialog boxes to confirm the restart and set behavior functions. It disables feedback for the move function.</p> <p data-bbox="735 1715 1325 1743">The default value for this resource is the all value.</p>

Item	Description
startupKeyFocus (class StartupKeyFocus)	Determines whether a window gets the keyboard input focus when the window is mapped (that is, initially managed by the window manager). It is recommended that both the autoKeyFocus resource and the startupKeyFocus resource be set to a value of True to work with tear-off menus. The default value is True. This resource is available only when the keyboard input focus policy is set to the explicit value.
transientDecoration (class TransientDecoration)	<p>Controls the amount of decoration that Mwm puts on transient windows. The decoration specification is exactly the same as for the clientDecoration (client-specific) resource. Transient windows are identified by the WM_TRANSIENT_FOR property, which is added by the client to indicate a relatively temporary window. The default value for this resource is the menu title value (that is, transient windows have resize borders and a title bar with a window menu button).</p> <p>An application can also specify which decorations the window manager should apply to its windows. If it does so, the window manager applies only those decorations indicated by both the application and the transientDecoration resource. Otherwise, the window manager applies only the decorations indicated by the transientDecoration resource.</p>
transientFunctions (class TransientFunctions)	<p>Indicates which window management functions are applicable (or not applicable) to transient windows. The function specification is exactly the same as for the clientFunctions (client-specific) resource. The default value for this resource is -minimize -maximize.</p> <p>An application can also specify which functions the window manager should apply to its windows. If it does so, the window manager applies only those functions indicated by both the application and the transientFunctions resource. Otherwise, the window manager applies only the functions indicated by the transientFunctions resource.</p>
useIconBox (class UseIconBox)	Determines whether icons are placed in an icon box. If this resource is given a value of True, icons are placed in an icon box. When an icon box is not used, the icons are placed on the root window (default value).
wMenuButtonClick (class WMenuButtonClick)	Indicates whether the window menu is posted and remains posted after a click of the mouse when the pointer is over the Window Menu button. If the value given this resource is True, the menu remains posted. True is the default value for this resource.
wMenuButtonClick2 (class WMenuButtonClick2)	Indicates whether a double-click action on the Window Menu button performs an f.kill function. When this resource is given the default value of True, a double-click action on the Window Menu button performs an f.kill function.

Client-Specific Resources

The syntax for specifying client-specific resources is as follows:

Mwm*ClientNameOrClass*ResourceID

For example, **Mwm*mterm>windowMenu** is used to specify the window menu to be used with **mterm** clients.

The syntax for specifying client-specific resources for all classes of clients is as follows:

Mwm*ResourceID

Specific client specifications take precedence over the specifications for all clients. For example, **Mwm>windowMenu** is used to specify the window menu to be used for all classes of clients that do not have a window menu specified.

The syntax for specifying resource values for windows that have an unknown name and class (that is, windows that do not have a **WM_CLASS** property associated with them) is as follows:

Mwm*defaults*ResourceID

For example, **Mwm*defaults*iconImage** is used to specify the icon image to be used for windows that have an unknown name and class.

Client-Specific Resource Set

Note: Hyphens in the following table are for readability purposes only. Do not include hyphens within names in programs.

Client-Specific Resource Set	
Name	Properties
clientDecoration	Class ClientDecoration Value type all Default
clientFunctions	Class ClientFunctions Value type string Default all
focusAutoRaise	Class FocusAutoRaise Value type True or False Default varies

Client-Specific Resource Set (continued)

Name	Properties
iconImage	<p>Class IconImage</p> <p>Value type pathname</p> <p>Default (image)</p>
iconImageBackground	<p>Class Background</p> <p>Value type color</p> <p>Default icon background</p>
iconImageBottomShadowColor	<p>Class Foreground</p> <p>Value type color</p> <p>Default icon bottom shadow</p>
iconImageBottomShadowPixmap	<p>Class BottomShadowPixmap</p> <p>Value type color</p> <p>Default icon bottom shadow pixmap</p>
iconImageForeground	<p>Class Foreground</p> <p>Value type color</p> <p>Default varies</p>
iconImageTopShadowColor	<p>Class Background</p> <p>Value type color</p> <p>Default icon top shadow color</p>
iconImageTopShadowPixmap	<p>Class TopShadowPixmap</p> <p>Value type color</p> <p>Default icon top shadow pixmap</p>

Client-Specific Resource Set (*continued*)

Name	Properties
matteBackground	<p>Class Background</p> <p>Value type color</p> <p>Default background</p>
matteBottomShadowColor	<p>Class Foreground</p> <p>Value type color</p> <p>Default bottom shadow color</p>
matteBottomShadowPixmap	<p>Class BottomShadowPixmap</p> <p>Value type color</p> <p>Default bottom shadow pixmap</p>
matteForeground	<p>Class Foreground</p> <p>Value type color</p> <p>Default foreground</p>
matteTopShadowColor	<p>Class Background</p> <p>Value type color</p> <p>Default top shadow color</p>
matteTopShadowPixmap	<p>Class TopShadowPixmap</p> <p>Value type color</p> <p>Default top shadow pixmap</p>
matteWidth	<p>Class MatteWidth</p> <p>Value type pixels</p> <p>Default 0</p>

Client-Specific Resource Set (continued)

Name	Properties
maximumClientSize	<p>Class MaximumClientSize</p> <p>Value type width x height, vertical, horizontal</p> <p>Default fill the screen</p>
useClientIcon	<p>Class UseClientIcon</p> <p>Value type True or False</p> <p>Default F</p>
usePPosition	<p>Class UsePPosition</p> <p>Value type string</p> <p>Default nonzero</p>
windowMenu	<p>Class WindowMenu</p> <p>Value type string</p> <p>Default DefaultWindowMenu</p>

Item	Description
clientDecoration (class ClientDecoration)	<p>Controls the amount of window frame decoration. The resource is specified as a list of decorations to specify their inclusion in the frame. If a decoration is preceded by - (a minus sign), that decoration is excluded from the frame. The sign of the first item in the list determines the initial amount of decoration. If the sign of the first decoration is a minus sign, the mwm command assumes all decorations are present and starts subtracting from that set. If the sign of the first decoration is plus (or not specified), the mwm command starts with no decoration and builds up a list from the resource.</p> <p>An application can also specify which decorations the mwm command should apply to its windows. If it does so, the mwm command applies only those decorations indicated by both the application and the clientDecoration resource. Otherwise, the mwm command applies the decorations indicated by the clientDecoration resource. Following is a list of window frame decorations:</p> <p>all Specifies to include all decorations (default value).</p> <p>border Specifies the window border.</p> <p>maximize Specifies the Maximize button (includes title bar).</p> <p>minimize Specifies the Minimize button (includes title bar).</p> <p>none Specifies no decorations.</p> <p>resizeh Specifies the border resize handles (includes border).</p> <p>menu Specifies the Window Menu button (includes title bar).</p> <p>title Specifies the title bar (includes border).</p> <p>Following are examples of window frame decoration commands:</p> <pre>Mwm*XClock.clientDecoration: -resizeh -maximize</pre> <p>This removes the resize handles and Maximize button from XClock windows.</p> <pre>Mwm*XClock.clientDecoration: menu minimize border</pre> <p>This removes the resize handles and Maximize button from XClock windows. Note that either menu or minimize implies title.</p>

Item	Description
clientFunctions (class ClientFunctions)	<p>Indicates which mwm functions are applicable (or not applicable) to the client window. The value for the resource is a list of functions. If the first function in the list has - (a minus sign) in front of it, the mwm command starts with all functions and subtracts from that set. If the first function in the list has a + (plus sign) in front of it, the mwm command starts with no functions and builds up a list. Each function in the list must be preceded by the appropriate + (plus) or - (minus) sign and separated from the next function by a space.</p> <p>An application can also specify which functions the mwm command should apply to its windows. If it does so, the mwm command applies only those functions indicated by both the application and the clientFunctions resource. Otherwise, the mwm command applies the functions indicated by the clientFunctions resource.</p> <p>Following is a list of functions available for this resource:</p> <p>all Specifies to include all functions (default value).</p> <p>none Specifies no functions.</p> <p>resize Specifies f.resize.</p> <p>move Specifies f.move.</p> <p>minimize Specifies f.minimize.</p> <p>maximize Specifies f.maximize.</p> <p>close Specifies f.kill.</p>
focusAutoRaise (class FocusAutoRaise)	<p>Determines whether clients are raised when they get the keyboard input focus. If the value is False, the stacking of windows on the display is not changed when a window gets the keyboard input focus. The default value is True when the keyboardFocusPolicy is the explicit value and False when the keyboardFocusPolicy is the pointer value.</p>
iconImage (class IconImage)	<p>Specifies an icon image for a client (for example, Mwm*myclock*iconImage). The resource value is a path name for a bitmap file. The value of the (client-specific) useClientIcon resource is used to determine whether user-supplied icon images are used instead of client-supplied icon images. The default value is to display a built-in window manager icon image.</p>
iconImageBackground (class Background)	<p>Specifies the background color of the icon image that is displayed in the image part of an icon. The default value of this resource is the icon background color (that is, specified by Mwm*background or Mwm*icon*background).</p>
iconImageBottomShadowColor (class Foreground)	<p>Specifies the bottom shadow color of the icon image that is displayed in the image part of an icon. The default value of this resource is the icon bottom shadow color (that is, specified by Mwm*icon*bottomShadowColor).</p>
iconImageBottomShadowPixmap (class BottomShadowPixmap)	<p>Specifies the bottom shadow pixmap of the icon image that is displayed in the image part of an icon. The default value of this resource is the icon bottom shadow pixmap (that is, specified by Mwm*icon*bottomShadowPixmap).</p>
iconImageForeground (class Foreground)	<p>Specifies the foreground color of the icon image that is displayed in the image part of an icon. The default value of this resource varies depending on the icon background.</p>
iconImageTopShadowColor (class Background)	<p>Specifies the top shadow color of the icon image that is displayed in the image part of an icon. The default value of this resource is the icon top shadow color (that is, specified by Mwm*icon*topShadowColor).</p>
iconImageTopShadowPixmap (class TopShadowPixmap)	<p>Specifies the top shadow pixmap of the icon image that is displayed in the image part of an icon. The default value of this resource is the icon top shadow pixmap (that is, specified by Mwm*icon*topShadowPixmap).</p>
matteBackground (class Background)	<p>Specifies the background color of the matte when the matteWidth resource is a positive value. The default value of this resource is the client background color (that is, specified by Mwm*background or Mwm*client*background).</p>

Item	Description
matteBottomShadowColor (class Foreground)	Specifies the bottom shadow color of the matte when the matteWidth resource is a positive value. The default value of this resource is the client bottom shadow color (that is, specified by Mwm*bottomShadowColor or Mwm*client*bottomShadowColor).
matteBottomShadowPixmap (class BottomShadowPixmap)	Specifies the bottom shadow pixmap of the matte when the matteWidth resource is a positive value. The default value of this resource is the client bottom shadow pixmap (that is, specified by Mwm*bottomShadowPixmap or Mwm*client*bottomShadowPixmap).
matteForeground (class Foreground)	Specifies the foreground color of the matte when the matteWidth resource is a positive value. The default value of this resource is the client foreground color (that is, specified by Mwm*foreground or Mwm*client*foreground).
matteTopShadowColor (class Background)	Specifies the top shadow color of the matte when the matteWidth resource is a positive value. The default value of this resource is the client top shadow color (that is, specified by Mwm*topShadowColor or Mwm*client*topShadowColor).
matteTopShadowPixmap (class TopShadowPixmap)	Specifies the top shadow pixmap of the matte when the matteWidth resource is a positive value. The default value of this resource is the client top shadow pixmap (that is, specified by Mwm*topShadowPixmap or Mwm*client*topShadowPixmap).
matteWidth (class MatteWidth)	Specifies the width of the optional matte. The default value is 0, which effectively disables the matte.
maximumClientSize (class MaximumClientSize)	Indicates the client size to be used when an application is maximized. The resource value is specified <i>WidthxHeight</i> . The width and height are interpreted in the units that the client uses (for example, this is generally characters for terminal emulators). Alternately, the vertical or horizontal value can be specified to indicate the direction in which the client maximizes. If this resource is not specified, the maximum size from the WM_NORMAL_HINTS property is used if set. Otherwise, the default value is the size where the client window with window management borders fills the screen. When the maximum client size is not determined by the maximumClientSize resource, the maximumMaximumSize resource value is used as a constraint on the maximum size.
useClientIcon (class UseClientIcon)	Determines whether a client-supplied icon image takes precedence over a user-supplied icon image. The default value is False, giving the user-supplied icon image higher precedence than the client-supplied icon image.
usePPosition (class UsePPosition)	Specifies whether the window manager honors the program-specified position PPosition specified in the WM_NORMAL_HINTS property in the absence of a user-specified position. Setting this resource to On causes the mwm command to always honor the program-specified position. Setting this resource to Off causes the mwm command to always ignore the program-specified position. Setting this resource to the default value of nonzero causes the mwm command to honor program-specified positions other than (0,0).
windowMenu (class WindowMenu)	Indicates the name of the menu pane that is posted when the window menu is opened (usually by pressing button 1 on the Window Menu button on the client window frame). Menu panes are specified in the mwm resource description file. Window menus can be customized on a client class basis by specifying resources of the form Mwm*ClientNameOrClass*windowMenu (See mwm Resource Description File Syntax for more information.) The default value of this resource is DefaultWindowMenu.

Resource Description File

The **mwm** resource description file is a supplementary resource file that contains resource descriptions that are referred to by entries in the defaults files (**.Xdefaults**, **app-defaults/Mwm**). It contains descriptions of resources that are to be used by the **mwm** command and that cannot be easily encoded in the defaults files (a bitmap file is an analogous type of resource description file). A particular **mwm** resource description file can be selected using the **configFile** resource.

The following types of resources can be described in the **mwm** resource description file:

Resource Description

Item	Description
buttons	Window manager functions can be bound (associated) with button events.
keys	Window manager functions can be bound (associated) with key press events.
menus	Menu panes can be used for the window menu and other menus posted with key bindings and button bindings.

mwm Resource Description File Syntax

The **mwm** resource description file is a standard text file that contains items of information separated by blanks, tabs, and new-line characters. Blank lines are ignored. Items or characters can be quoted to avoid special interpretation (for example, the # (comment character) can be quoted to prevent it from being interpreted as the comment character). A quoted item can be contained in "" (double quotation marks). Single characters can be quoted by preceding them with the \ (backslash). All text from an unquoted # (comment character) to the end of the line is regarded as a comment and is not interpreted as part of a resource description. If an ! (exclamation mark) is the first character in a line, the line is regarded as a comment. If a line ends in a \ (backslash), the next line is considered a continuation of that line.

Window manager functions can be accessed with button and key bindings and with window manager menus. Functions are indicated as part of the specifications for button and key binding sets and for menu panes. The function specification has the following syntax:

```
Function = FunctionName [FunctionArguments]
FunctionName = Window Manager Function
FunctionArguments = {QuotedItem | UnquotedItem}
```

The following functions are supported. If a function is specified that is not one of the supported functions, it is interpreted by the **mwm** command as the **f.nop** function.

Resource Description File Syntax

Item	Description
f.beep	Causes a beep.
f.circle_down [<i>Icon</i> <i>Window</i>]	Causes the window or icon that is on the top of the window stack to be put on the bottom of the window stack (so that it no longer obscures any other window or icon). This function affects only those windows and icons that obscure other windows and icons or that are obscured by other windows and icons. Secondary windows (that is, transient windows) are restacked with their associated primary window. Secondary windows always stay on top of the associated primary window and there can be no other primary windows between the secondary windows and their primary window. If an <i>Icon</i> function argument is specified, the function applies only to icons. If a <i>Window</i> function argument is specified, the function applies only to windows.

Item	Description
f.circle_up [<i>Icon</i> <i>Window</i>]	Raises the window or icon on the bottom of the window stack (so that it is not obscured by any other windows). This function affects only those windows and icons that obscure other windows and icons or that are obscured by other windows and icons. Secondary windows (that is, transient windows) are restacked with their associated primary window. If an <i>Icon</i> function argument is specified, the function applies only to icons. If a <i>Window</i> function argument is specified, the function applies only to windows.
f.exec or !	Causes the command to be run (using the value of the MWMSHELL environment variable if it is set; otherwise, the value of the SHELL environment variable if it is set; otherwise, /usr/bin/sh is used). The ! notation can be used in place of the f.exec function name.
f.focus_color	Sets the colormap focus to a client window. If this function is done in a root context, the default colormap (set up by the X Window System client for the screen where MWM is running) is installed and there is no specific client window colormap focus. This function is treated as f.nop if colormapFocusPolicy is not set to the explicit value.
f.focus_key	Sets the keyboard input focus to a client window or icon. This function is treated as f.nop if keyboardFocusPolicy is not set to the explicit value or the function is run in a root context.
f.kill	Stops a client. If the WM_DELETE_WINDOW protocol is set up, the client is sent a client message event indicating that the client window needs to be deleted. If the WM_SAVE_YOURSELF protocol is set up and the WM_DELETE_WINDOW protocol is not set up, the client is sent a client message event indicating that the client needs to prepare to be stopped. If the client does not have the WM_DELETE_WINDOW or WM_SAVE_YOURSELF protocol set up, this function causes a client's X connection to be stopped (usually resulting in the end of the client).
	See the description of the quitTimeout resource.
f.lower [<i>-Client</i> within freeFamily]	Lowers a client window to the bottom of the window stack (where it obscures no other window). Secondary windows (that is, transient windows) are restacked with their associated primary window. The <i>Client</i> argument indicates the name or class of a client to lower. If the <i>Client</i> argument is not specified, the context, in which the function was started, indicates the window or icon to lower.

Item	Description
f.maximize	Causes a client window to be displayed with its maximum size.
f.menu	Associates a cascading (pull-right) menu with a menu pane entry or a menu with a button or key binding. The menu_name function argument identifies the menu to be used.
f.minimize	Causes a client window to be iconified (minimized). When a window is minimized and no icon box is used, its icon is placed on the bottom of the window stack (so that it obscures no other windows). If an icon box is used, the client's icon changes to its iconified form inside the icon box. Secondary windows (that is, transient windows) are minimized with their associated primary window. There is only one icon for a primary window and all its secondary windows.
f.move	Causes a client window to be interactively moved.
f.next_cmap	Installs the next colormap in the list of colormaps for the window with the colormap focus.
f.next_key [<i>Icon</i> <i>Window</i> <i>Transient</i>]	Sets the keyboard input focus to the next window or icon in the set of windows and icons managed by the window manager (the ordering of this set is based on the stacking of windows on the screen). This function is treated as f.nop if keyboardFocusPolicy is not the explicit value. The keyboard input focus is moved only to windows that do not have an associated secondary window that is application-modal. If the <i>Transient</i> argument is specified, transient (secondary) windows are crossed (otherwise, if only the <i>Window</i> argument is specified, traversal is done only to the last focused window in a transient group). If an <i>Icon</i> function argument is specified, the function applies only to icons. If a <i>Window</i> function argument is specified, the function applies only to windows.
f.nop	Does nothing. If a function is specified in a type of resource where it is not supported or is started in a context that does not apply, the function is treated as f.nop .
f.normalize	Causes a client window to be displayed with its normal size. Secondary windows (that is, transient windows) are placed in their normal state along with their associated primary window.
f.normalize_and_raise	Causes the corresponding client window to be displayed with its normal size and raised to the top of the window stack. Secondary windows (that is, transient windows) are placed in their normal state along with their associated primary window.

Item	Description
f.pack_icons	Causes icons to be packed into the icon grid. This function is used to relay out icons (based on the layout policy being used) on the root window or in the icon box.
f.pass_keys	Enables or disables (toggles) processing of key bindings for window manager functions. When it disables key binding processing, all keys are passed on to the window with the keyboard input focus and no window manager functions are started. If the f.pass_keys function is started with a key binding to disable key-binding processing, the same key binding can be used to enable key-binding processing.
f.post_wmenu	Posts the window menu. If a key is used to post the window menu and the Window Menu button is present, the window menu is automatically placed with its top-left corner at the bottom-left corner of the Window Menu button for the client window. If no Window Menu button is present, the window menu is placed at the top-left corner of the client window.
f.prev_cmap	Installs the previous colormap in the list of colormaps for the window with the colormap focus.
f.prev_key [<i>Icon</i> <i>Window</i> <i>Transient</i>]	Sets the keyboard input focus to the previous window or icon in the set of windows and icons managed by the window manager (the ordering of this set is based on the stacking of windows on the screen). This function is treated as f.nop if keyboardFocusPolicy is not the explicit value. The keyboard input focus is moved only to windows that do not have an associated secondary window that is application-modal. If the <i>Transient</i> argument is specified, transient (secondary) windows are crossed (otherwise, if only window is specified, traversal is done only to the last focused window in a transient group). If an <i>Icon</i> function argument is specified, the function applies only to icons. If a <i>Window</i> function argument is specified, the function applies only to windows.
f.quit_mwm	Stops the mwm command (but <i>not</i> the X Window System client).

Item	Description
f.raise [- <i>Client</i> within freeFamily]	Raises a client window to the top of the window stack (where it is obscured by no other window). Raises the secondary window (transient window or dialog box) within the client family. The arguments to this function are mutually exclusive. The <i>Client</i> argument indicates the name or class of a client to raise. If the <i>Client</i> argument is not specified, the context in which the function was started indicates the window or icon to raise. Specifying within raises the secondary window within the family but does not raise the client family in the global window stack. Specifying freeFamily raises the window to the top of its local family stack and raises the family to the top of the global window stack.
f.raise_lower [within freeFamily]	Raises a primary window to the top of the window stack if it is partially obscured by another window; otherwise, it lowers the window to the bottom of the window stack. The arguments to this function are mutually exclusive. Specifying within raises a secondary window within the family (staying above the parent window), if it is partially obscured by another window in the application's family; otherwise, it lowers the window to the bottom of the family stack. It has no effect on the global stacking order. Specifying freeFamily raises the window to the top of its local family stack, if obscured by another window, and raises the family to the top of the global window stack; otherwise, it lowers the window to the bottom of its local family stack and lowers the family to the bottom of the global window stack.
f.refresh	Causes all windows to be redrawn.
f.refresh_win	Causes a client window to be redrawn.
f.resize	Causes a client window to be interactively resized.
f.restart	Causes the mwm command to be restarted (effectively stopped and restarted).
f.restore	Restores the previous state of an icon's associated window. If a maximized window is iconified, the f.restore function restores it to its maximized state. If a normalized window is iconified, the f.restore function restores it to its normalized state.

Item	Description
f.restore_and_raise	Restores the previous state of an icon's associated window and raises the window to the top of the window stack. If a maximized window is iconified, the f.restore_and_raise function restores it to its maximized state and raises it to the top of the window stack. If a normalized window is iconified, the f.restore_and_raise function restores it to its normalized state and raises it to the top of the window stack.
f.screen [next prev back <i>ScreenNumber</i>]	Causes the pointer to warp to a specific screen number or to the next, previous, or last visited screen. The arguments to this function are mutually exclusive. The following arguments are available: ScreenNumber Indicates the screen number to which the pointer is warped. Screens are numbered starting from screen 0. next Warps the pointer to the next managed screen (skipping over any unmanaged screens). prev Warps the pointer to the previous managed screen (skipping over any unmanaged screens). back Warps the pointer to the last visited screen.
f.send_msg <i>MessageNumber</i>	Sends a client message of the _MOTIF_WM_MESSAGES type with the <i>MessageType</i> indicated by the <i>MessageNumber</i> function argument. The client message is sent only if <i>MessageNumber</i> is included in the client's _MOTIF_WM_MESSAGES property. A menu item label is unavailable if the menu item is used to perform the f.send_msg function of a message that is not included in the client's _MOTIF_WM_MESSAGES property.
f.separator	Causes a menu separator to be put in the menu pane at the specified location (the label is ignored).

Item	Description
f.set_behavior	<p>Causes the window manager to restart with the default behavior (if a custom behavior is configured) or revert to the custom behavior. By default this is bound to the Shift+Ctrl+Meta+! key sequence.</p> <p>The Meta+Shift+Ctrl+! key sequence switches (that is, toggles) between the default and custom behaviors. When the user switches to the default MWM behavior, a number of mwm resources assume their default values and the mwm command restarts. When the user switches back to the custom behavior, the resource values that were changed to default values are reset with the custom values and the mwm command restarts.</p> <p>When an f.set_behavior function is performed, the following user interaction occurs:</p> <ol style="list-style-type: none"> 1. A system-modal dialog box is displayed prompting the user for confirmation of the f.set_behavior action. 2. The user can cancel the action at this point. 3. The window manager restarts. 4. The window manager applies the new (custom or default) configuration values. 5. Window manager components are mapped. <p>When the default MWM behavior is being set, default resource values are applied and, if specified, client properties that control window manager behavior are applied. This includes the _MOTIF_WM_HINTS and _MOTIF_WM_MENU properties. These properties might alter default MWM behavior, but it is done in a way that is consistent for all users.</p>
f.title	<p>Inserts a title in the menu pane at the specified location.</p>

Function Contexts

Each function may be constrained as to which resource types can specify the function (for example, menu pane) and also what context the function can be used in (for example, the function is done to the selected client window). The following are the function contexts:

Function contexts

Item	Description
root	No client window or icon is selected as an object for the function.
window	A client window is selected as an object for the function. This includes the window's title bar and frame. Some functions are applied only when the window is in its normalized state (for example, f.maximize) or its maximized state (for example, f.normalize).
icon	An icon is selected as an object for the function.

If a function's context is specified as **icon|window** and the function is started in an icon box, the function applies to the icon box, not to the icons inside.

If a function is specified in a type of resource where it is not supported or is started in a context that does not apply, the function is treated as **f.nop**. The following table indicates the resource types and function contexts in which window manager functions apply:

Function contexts		
Function	Contexts	Resources
f.beep	root, icon, window	button, key, menu
f.circle_down	root, icon, window	button, key, menu
f.circle_up	root, icon, window	button, key, menu
f.exec	root, icon, window	button, key, menu
f.focus_color	root, icon, window	button, key, menu
f.focus_key	root, icon, window	button, key, menu
f.kill	icon, window	button, key, menu
f.lower	icon, window	button, key, menu
f.maximize	icon, window (normal)	button, key, menu
f.menu	root, icon, window	button, key, menu
f.minimize	window	button, key, menu
f.move	icon, window	button, key, menu
f.next_cmap	root, icon, window	button, key, menu
f.next_key	root, icon, window	button, key, menu
f.nop	root, icon, window	button, key, menu

Normalize item description		
Item	Description	
f.normalize	icon, window (maximized)	button, key, menu
f.normalize_and_raise	icon, window	button, key, menu
f.pack_icons	root, icon, window	button, key, menu
f.pass_keys	root, icon, window	button, key, menu
f.post_wmenu	root, icon, window	button, key
f.prev_cmap	root, icon, window	button, key, menu
f.prev_key	root, icon, window	button, key, menu
f.quit_mwm	root, icon, window	button, key, menu (root only)
f.raise	icon, window	button, key, menu
f.raise_lower	icon, window	button, key, menu
f.refresh	root, icon, window	button, key, menu
f.refresh_win	window	button, key, menu
f.resize	window	button, key, menu

Normalize item description (<i>continued</i>)		
Item	Description	
f.restart	root, icon, window	button, key, menu
f.restore	icon, window	button, key, menu
f.restore_and_raise	icon, window	button, key, menu
f.screen	root, icon, window	button, key, menu
f.send_msg	icon, window	button, key, menu
f.separator	root, icon, window	menu
f.set_behavior	root, icon, window	button, key, menu
f.title	root, icon, window	menu

Window Manager Event Specification

Events are indicated as part of the specifications for button and key-binding sets and for menu panes.

Button events have the following syntax:

```
Button = [ModifierList]<ButtonEventName>
ModifierList = Modifier Name {ModifierName}
```

All modifiers specified are interpreted as being exclusive (this means that only the specified modifiers can be present when the button event occurs). Following is a list that indicates the values that can be used for the *ModifierName* parameter. The Alt key is frequently labeled Extend or Meta. Alt and Meta can be used interchangeably in event specification.

Window Manager Event Specification

Item	Description
Modifier	Description
Ctrl	Control key
Shift	Shift key
Alt	Alt or Meta key
Meta	Meta or Alt key
Lock	Lock key
Mod1	Modifier1
Mod2	Modifier2
Mod3	Modifier3
Mod4	Modifier4
Mod5	Modifier5

Following is a list that indicates the values that can be used for the *ButtonEventName* parameter.

Button Event Name descriptions

Button	Description
Btn1Down	Button 1 press

Button Event Name descriptions (*continued*)

Button	Description
Btn1Up	Button 1 release
Btn1Click	Button 1 press and release
Btn1Click2	Button 1 double click
Btn2Down	Button 2 press
Btn2Up	Button 2 release
Btn2Click	Button 2 press and release
Btn2Click2	Button 2 double click
Btn3Down	Button 3 press
Btn3Up	Button 3 release
Btn3Click	Button 3 press and release
Btn3Click2	Button 3 double click
Btn4Down	Button 4 press
Btn4Up	Button 4 release
Btn4Click	Button 4 press and release
Btn4Click2	Button 4 double click
Btn5Down	Button 5 press
Btn5Up	Button 5 release
Btn5Click	Button 5 press and release
Btn5Click2	Button 5 double click.

Key events that are used by the window manager for menu mnemonics and for binding to window manager functions are single key presses; key releases are ignored. Key events have the following syntax:

```
Key = [ModifierList] <Key> KeyName  
ModifierList = ModifierName {ModifierName}
```

All modifiers specified are interpreted as being exclusive (this means that only the specified modifiers can be present when the key event occurs). Modifiers for keys are the same as those that apply to buttons. The *KeyName* parameter is an X11 keysym name. Key symbol names can be found in the **keysymdef.h** file (remove the *XK_* prefix).

The key symbol names will be resolved to a single specific key code by the Window Manager during startup and will not change unless the Window Manager is restarted.

Button Bindings

The **buttonBindings** resource value is the name of a set of button bindings that are used to configure window manager behavior. A window manager function can be used when a button press occurs with the pointer over a framed client window, an icon, or the root window. The context for indicating where the button press applies is also the context for starting the window manager function when the button press is done (significant for functions that are context-sensitive).

Following is the button binding syntax:

```
Buttons BindingsSetName  
{
```

```

Button Context Function
Button Context Function
.
.
Button Context Function
}

```

Following is the syntax for the context specification:

```

Context = Object[|Context]
Object = root | icon | window | title | frame | border | app

```

The *Context* specification indicates where the pointer must be for the button binding to be effective. For example, a context of **window** indicates that the pointer must be over a client window or window management frame for the button binding to be effective. The **frame** context is for the window management frame around a client window (including the border and title bar), the **border** context is for the border part of the window management frame (not including the title bar), the **title** context is for the title area of the window management frame, and the **app** context is for the application window (not including the window management frame).

If an **f.nop** function is specified for a button binding, the button binding is not done.

Key Bindings

The **keyBindings** resource value is the name of a set of key bindings that are used to configure window manager behavior. A window manager function can be done when a particular key is pressed. The context in which the key binding applies is indicated in the key binding specification. The valid contexts are the same as those that apply to button bindings.

Following is the key binding syntax:

```

Keys BindingsSetName
{
Key Context Function
Key Context Function
.
.
Key Context Function
}

```

If an **f.nop** function is specified for a key binding, the key binding is not done. If an **f.post_wmenu** or **f.menu** function is bound to a key, the **mwm** command automatically uses the same key for removing the menu from the screen after it is open.

The *Context* specification syntax is the same as for button bindings. For key bindings, the **frame**, **title**, **border**, and **app** contexts are equivalent to the **window** context. The context for a key event is the window or icon that has the keyboard input focus (**root** if no window or icon has the keyboard input focus).

Menu Panes

Menus can be opened using the **f.post_wmenu** and **f.menu** window manager functions. The context for window manager functions that are done from a menu is **root**, **icon**, or **window**, depending on how the menu is opened. In the case of the window menu or menus opened with a key binding, the location of the keyboard input focus indicates the context. For menus opened using a button binding, the context of the button binding is the context of the menu.

Following is the menu pane specification syntax:

```

Menu MenuName
{
Label [Mnemonic] [Accelerator] Function
Label [Mnemonic] [Accelerator] Function
}

```

```

Label [Mnemonic] [Accelerator] Function
}

```

Each line in the *Menu* specification identifies the label for a menu item and the function to be completed if the menu item is selected. Optionally, a menu button mnemonic and a menu button keyboard accelerator can be specified. Mnemonics are functional only when the menu is posted and keyboard traversal applies.

The label can be a string or a bitmap file. The *Label* specification has the following syntax:

```

Label = Text | BitmapFile
BitmapFile = @FileName
Text = QuotedItem | UnquotedItem

```

The string encoding for labels must be compatible with the menu font that is used. Labels are not available for menu items that use the **f.nop** function, an invalid function, or a function that does not apply in the current context.

A *Mnemonic* specification has the following syntax:

```

Mnemonic = _Character

```

The first matching *Character* in the label is underlined. If there is no matching *Character* in the label, no mnemonic is registered with the window manager for that label. Although the *Character* must exactly match a character in the label, the mnemonic does not perform if any modifier (such as the Shift key) is pressed with the character key.

The *Accelerator* specification is a key event specification with the same syntax that is used for key bindings to window manager functions.

Environment

The **mwm** command does the following:

- Uses the **HOME** environment variable to specify the user's home directory.
- Uses the **LANG** environment variable to specify the user's choice of language for the **mwm** message catalog and the **mwm** resource description file.
- Uses the **XFILESEARCHPATH**, **XUSERFILESEARCHPATH**, **XAPPLRESDIR**, **XENVIRONMENT**, **LANG**, and **HOME** environment variables to determine search paths for resource default files. The **mwm** command can also use the **XBMLANGPATH** environment variable to search for bitmap files.
- Reads the **\$HOME/.motifbind** file, if it exists, to install a virtual key bindings property on the root window.
- Uses the **MWMShell** environment variable (or **SHELL** if **MWMShell** is not set) to specify the shell to use when running commands through the **f.exec** function.

Exit Status

This command returns the following exit values:

Exit Status

It	Description
m	

0	Indicates successful completion.
----------	----------------------------------

>1	Indicates an error occurred.
--------------	------------------------------

Files

/usr/lib/X11/\$LANG/system.mwmrc

/usr/lib/X11/system.mwmrc
/usr/lib/X11/app-defaults/Mwm
\$HOME/Mwm
\$HOME/.Xdefaults
\$HOME/\$LANG/.mwmrc
\$HOME/.mwmrc
\$HOME/.motifbind

n

The following AIX commands begin with the letter *n*.

named Daemon – BIND Version 9.16

Purpose

Provides the server function for the Domain Name Protocol.

Syntax

Refer to the syntax of the **named9** daemon.

Description

AIX 7.3 supports only BIND version 9.16.

Files

Item	Description
<code>/usr/sbin/named</code>	Contains the soft link to the named9 daemon.
<code>/usr/sbin/bind_9_16/named9</code>	Contains the named9 daemon.
<code>/etc/resolv.conf</code>	Specifies the use of domain name services.
<code>/etc/services</code>	Defines socket service assignments.

named-checkconf Command

Purpose

Checks the syntax of a DNS configuration file.

Syntax

```
named-checkconf [-chjlvz] [-p [-x]] [-t directory] {filename}
```

Description

The **named-checkconf** command checks the syntax, but not the semantics, of a `named.conf` configuration file. The `named.conf` configuration file and all files that it includes are parsed and checked for syntax errors. If no file is specified, the `/etc/named.conf` file is read by default.

Note: The `named` daemon reads some files (such as `rndc.key` and `bind.keys`) in separate parser contexts. These files are not automatically checked by the **named-checkconf** command. Configuration errors in these files might cause the **named** daemon to fail, even if the **named-checkconf** command was successful. However, you can run the **named-checkconf** command on these files explicitly.

Flags

Item	Description
<code>-j</code>	Reads the journal if it exists when loading a zonefile.

Item	Description
-l	Lists all the configured zones. Each line of output contains the zone name, class (for example, IN), view, and type (for example, primary or secondary).
-c	Indicates to check only the core configuration. This flag suppresses the loading of plugin modules, and thus, all parameters of the plugin statements are ignored.
-i	Ignores warning messages related to deprecated options.
-p	Prints the named.conf configuration file and the included files in canonical form if no errors were detected. Also, see the -x option.
-t <i>directory</i>	Changes the present directory to the directory specified so that included directives in the configuration file are processed.
-v	Prints the version of the named-checkconf program and exits.
-x	Obscures shared secrets by replacing them with strings of question marks (?) when you print the configuration files in canonical form. This flag allows the contents of the named.conf configuration file and related files to be shared. For example, when submitting bug reports that do not contain private data. This option cannot be used without the -p flag.
-z	Performs a check and loads the master zone files found in the named.conf configuration file.
<i>filename</i>	Specifies the name of the configuration file to be checked. If not specified, the default value is /etc/named.conf.

Exit Status

Item	Description
0	Indicates a successful completion.
1	Indicates errors.

named-checkzone, named-compilezone Commands

Purpose

Zone file validity checking or converting tool of a named configuration file.

Syntax

```
named-checkzone [ -d ] [ -h ] [ -j ] [ -q ] [ -v ] [ -c class ] [ -f format ] [ -F format ] [ -J filename ] [ -i mode ]
[ -k mode ] [ -l ttl ] [ -L serial ] [ -m mode ] [ -M mode ] [ -n mode ] [ -o filename ] [ -r mode ] [ -s style ]
[ -S mode ] [ -t directory ] [ -T mode ] [ -w directory ] [ -D ] [ -W mode ] zonename filename
```

```
named-compilezone [ -d ] [ -h ] [ -j ] [ -q ] [ -v ] [ -c class ] [ -f format ] [ -F format ] [ -J filename ] [ -i
mode ] [ -k mode ] [ -l ttl ] [ -L serial ] [ -m mode ] [ -n mode ] [ -o filename ] [ -s style ] [ -t directory ] [ -T
mode ] [ -w directory ] [ -D ] [ -W mode ] zonename filename
```

Description

The **named-checkzone** command checks the syntax and integrity of a zone file. It performs the same checks as the **named** daemon does when loading a zone. This makes the **named-checkzone** command useful for checking zone files before configuring them into a name server.

The **named-compilezone** command is similar to the **named-checkzone** command, but it always dumps the zone contents to a specified file in a specified format. Additionally, it applies stricter check levels by default, since the dump output will be used as an actual zone file loaded by the named daemon. When manually specified otherwise, the check levels must at least be as strict as those specified in the named configuration file.

Flags

Item	Description
-c <i>class</i>	Specifies the class of the zone. If not specified, the class is set to IN by default.
-d	Enables debugging.
-h	Prints usage summary.
-D	Dumps zone file in canonical format. This is always enabled for the named-compilezone command.
-i <i>mode</i>	Performs post load zone integrity checks. The <i>mode</i> parameter can be one of the following values: full Checks if MX records, SRV records, and delegation NS records refer to A or AAAA record (both in-zone and out-of-zone host names). It also checks if glue addresses records in the zone match those advertised by the child. full-sibling Disables sibling glue checks but is otherwise the same as mode full . local Only checks if MX records, SRV records, and delegation NS records refer to in-zone host names or if some required glue exists, that is when the name server is in a child zone. local-sibling Disables sibling glue checks but is otherwise the same as mode local . none Disables the checks.
-j	Reads the journal if it exists when loading the zone file.
-f <i>format</i>	Specifies the format of the zone file. Possible formats are <code>text</code> (default) and <code>raw</code> .
-F <i>format</i>	Specifies the format of the output file specified. Possible formats are <code>text</code> (default), <code>map</code> , <code>ra</code> and <code>raw=N</code> . This flag does not cause any effects unless it dumps the zone contents.
-J <i>filename</i>	Specifies the file from which the journal file is read when a zone file is loaded. The -J flag implies the -j flag.
-k <i>mode</i>	Performs check-names checks with the specified failure mode. Possible modes are <code>fail</code> , <code>warn</code> (default) and <code>ignore</code> .
-l <i>ttl</i>	Sets a maximum permissible time-to-live (TTL) for the input file. Any record with a TTL higher than this value causes the zone to be rejected. This flag usage is similar to using the <code>max-zone-ttl</code> option in the <code>named.conf</code> file.

Item	Description
-L <i>serial</i>	Sets the source serial value in the header to the specified serial number when a zone file is compiled to the raw or map format. You must use this flag for testing purposes only.
-m <i>mode</i>	Specifies whether MX records must be checked to see if they are addresses. Possible modes are <i>fail</i> , <i>warn</i> (default) and <i>ignore</i> .
-M <i>mode</i>	Checks if a MX record refers to a CNAME. Possible modes are <i>fail</i> , <i>warn</i> (default) and <i>ignore</i> .
-n <i>mode</i>	Specifies whether NS records must be checked to see if they are addresses. Possible modes are <i>fail</i> , <i>warn</i> (default) and <i>ignore</i> .
-o <i>filename</i>	Writes zone output to the file specified by the <i>filename</i> value.
-q	Indicates quiet mode (exits code only).
-r <i>mode</i>	Checks for records that are considered as different by DNSSEC but are semantically equal in DNS. Valid values for the mode parameter are <i>fail</i> , <i>warn</i> (default), and <i>ignore</i> .
-s <i>style</i>	Specifies the style of the dumped zone file. Possible styles are <i>full</i> (default) and <i>relative</i> . The <i>full</i> format is most suitable for processing automatically by a separate script. On the other hand, the <i>relative</i> format is more human-readable and is thus suitable for editing by hand. This flag does not cause any effects unless it dumps the zone file contents. It also does not have any meaning if the output format is not text.
-S <i>mode</i>	Checks if a SRV record refers to a CNAME. Possible modes are <i>fail</i> , <i>warn</i> (default) and <i>ignore</i> .
-t <i>directory</i>	Changes the directory to the <i>directory</i> so that included directives in the configuration file are processed.
-v	Prints the version of the named-checkzone command and exits.
-w <i>directory</i>	Changes the current directory to the <i>directory</i> so that relative file names in master file \$INCLUDE directives work. This is similar to the directory clause in the named.conf file.
-W <i>mode</i>	Specifies whether to check for non-terminal wildcards. Non-terminal wildcards are almost always the result of a failure to understand the wildcard matching algorithm (RFC 1034). Possible modes are <i>warn</i> (default) and <i>ignore</i> .
<i>zonename</i>	Specifies the domain name of the zone being checked.
<i>filename</i>	Specifies the name of the zone file.

Exit Status

Item	Description
0	Indicates a successful completion.
1	Indicates errors.

named9 Daemon

Purpose

Manages the Internet domain name system (DNS) server operations.

Syntax


```
named9 [ -4 ] [ -6 ] [ -c config-file ] [ -d debug-level ] [ -D string ] [ -E engine-name ] [ -f ] [ -g ] [ -L logfile ]  
[ -M option ] [ -m flag ] [ -n #cpus ] [ -p port ] [ -s ] [ -S #max-socks ] [ -t directory ] [ -U #listeners ] [ -u  
user ] [ -v ] [ -V ] [ -X lock-file ] [ -x cache-file ]
```

Description

named9 is a Domain Name System (DNS) server, part of the BIND 9 distribution from Internet Systems Consortium (ISC). For more information on the DNS, see [RFC 1033](#), [RFC 1034](#), and [RFC 1035](#). When invoked without arguments, the **named9** daemon reads the default configuration file `/etc/named.conf`, reads any initial data, and listens for queries.

Flags

Item	Description
-4	Uses only the IPv4 protocol even if the host machine is capable of IPv6. The -4 and -6 options are mutually exclusive.
-6	Uses IPv6 only even if the host machine is capable of IPv4. The -4 and -6 options are mutually exclusive.
-c <i>config-file</i>	Uses <i>config-file</i> as the configuration file instead of the default, <code>/etc/named.conf</code> . To ensure that reloading the configuration file continues to work after the server has changed its working directory due to a possible directory option in the configuration file, the <i>config-file</i> value must be an absolute path name.
-d <i>debug-level</i>	Sets the daemon's debug level to the <i>debug-level</i> value. Debugging traces from the named9 daemon become more verbose as the debug level increases.
-D <i>string</i>	Specifies a string that is used to identify an instance of named9 daemon in a process listing. The contents of <i>string</i> are not examined.
-E <i>engine-name</i>	Specifies the hardware to be used for cryptographic operations, whenever applicable, such as digital signing by a secure key store method. When BIND 9 is built with OpenSSL, you must specify this flag with an OpenSSL engine identifier that drives the cryptographic accelerator or hardware service module, such as <code>pkcs11</code> . When BIND is built with native Public Key Cryptography Standards (PKCS)#11 cryptography (<code>--enable-native-pkcs11</code>), this flag is set by default to the path of the PKCS#11 provider library that is specified by using <code>--with-pkcs11</code> .
-f	Runs the server in the foreground.
-g	Runs the server in the foreground and forces all logging to the standard error stderr .
-L <i>logfile</i>	Sets the logging operation to the <i>logfile</i> file, instead of the default system log.

Item	Description
-M <i>option</i>	<p>Sets the default memory context options. Valid values are as follows:</p> <p>external The internal memory manager is bypassed for system-provided memory allocation functions.</p> <p>fill Blocks of memory are filled with tag values when memory is allocated or freed, to assist debugging of memory problems.</p> <p>Nofill Disables this option and uses the default value unless named9 daemon is compiled with developer options.</p>
-m <i>flag</i>	Turns on memory usage debugging flags. Possible flags are <i>usage</i> , <i>trace</i> , <i>record</i> , <i>size</i> , and <i>mctx</i> . These values correspond to the <code>ISC_MEM_DEBUGXXXX</code> flags that are described in <code><isc/mem.h></code> .
-n <i>#cpus</i>	Creates <i>#cpus</i> worker threads to take advantage of multiple CPUs. If not specified, the named9 daemon tries to determine the number of CPUs present and creates one thread per CPU. If it is unable to determine the number of CPUs, the named9 daemon creates a single worker thread.
-p <i>port</i>	Listens for queries on port <i>port</i> . If not specified, the default is port 53.
-s	Writes memory usage statistics to the standard output stdout on exit.
-S <i>#max-socks</i>	<p>Allows the named9 daemon to use up to <i>#max-socks</i> sockets. The default value is 21000.</p> <p> Warning: Set this option only when the default configuration causes exhaustion of file descriptors and the operation environment is known to support the specified number of sockets. Setting this option unnecessarily can yield harmful results because the specified value might exceed the limit of the underlying system API. In addition, the maximum number is normally slightly fewer than the specified value, because named9 reserves some file descriptors for its internal use.</p>
-t <i>directory</i>	<p>Changes the current directory to the specified directory after processing the command line arguments, but before reading the configuration file.</p> <p>Warning: You must use this option in conjunction with the -u option, because changing the current directory of a process that is running as root does not enhance security on most systems.</p>
-u <i>user</i>	Sets the process user ID to the specified user after completing privileged operations, such as creating sockets that listen on privileged ports.
-U <i>#listeners</i>	Specifies the number of worker threads that the named9 daemon must listen on, for incoming UDP packets on each address. If you don't specify this flag, the named9 daemon calculates a default value based on the number of detected CPUs: 1 for 1 CPU, and the number of detected CPUs minus one for machines with more than 1 CPU. This value cannot be increased to a value greater than the number of CPUs. If the -n flag is set to a greater value than the number of detected CPUs, then the -U flag might be increased to that value, but not to a greater value.
-v	Reports the version number and exit.
-x <i>cache-file</i>	Loads data from the specified <i>cache-file</i> into the cache of the default configuration file.

Item	Description
-X <i>lock-file</i>	Acquires a lock on the specified file at runtime. This lock prevents duplicate named9 instances from running simultaneously. Use of this option overrides the <code>lock-file</code> option in the <code>named.conf</code> file. If set to <code>none</code> , the lock file check is disabled.

Signals

In routine operation, you cannot use signals to control the name server; you must use the **rndc** command.

Item	Description
SIGHUP	Forces a reload of the server.
SIGINT, SIGTERM	Shuts down the server.

The result of sending any other signals to the server is undefined.

Configuration

For a complete description of the **named9** configuration file, refer to the BIND 9 Administrator Reference Manual.

Files

Item	Description
/usr/sbin/bind_9_16/named9	Contains the named9 daemon.
/etc/named.conf	The default configuration file.
/etc/named.pid	The default process-id file.

namerslv Command

Purpose

Directly manipulates domain name server entries for local resolver routines in the system configuration database.

Syntax

To Add a Name Server Entry

```
namerslv -a { -i IPAddress | -D DomainName | -S SearchList }
```

To Delete a Name Server Entry

```
namerslv -d { -i IPAddress | -n | -l }
```

To Delete All Name Server Entries

```
namerslv -X [ -I ]
```

To Change a Name Server Entry

```
namerslv -c DomainName
```

To Display a Name Server Entry

```
namerslv -s [ -I | -n | -l ] [ -Z ]
```

To Create the Configuration Database File

namerslv **-b** [**-i** *IPAddress* [**-D** *DomainName*] [**-S** *SearchList*]]

To Rename the Configuration Database File

namerslv **-E** *FileName*

To Move the Configuration Database File to Prevent Name Server Use

namerslv **-e**

To Import a File into the Configuration Database File

namerslv **-B***FileName*

To Change a Search List Entry

namerslv **-C***Search List*

Description

The **namerslv** low-level command adds or deletes domain name server entries for local resolver routines in the system configuration database. By default, the system configuration database is contained in the **/etc/resolv.conf** file is moved to the file specified by the *FileName* variable.

Item	Description
-a	Adds an entry to the system configuration database. The -a flag must be used with either the -i or -D flag.
-B <i>FileName</i>	Restores the /etc/resolv.conf file from the file specified by the FileName variable.
-b	Creates the system configuration database, using the /etc/resolv.conf.sv file. If the /etc/resolv.conf.sv file does not exist, an error is returned. Note: The /etc/resolv.conf.sv file is not shipped with the system. You have to create the file before the -b flag will work.
-C	Changes the search list in the /etc/resolv.conf file.
-c <i>DomainName</i>	Changes the domain name in the system configuration database.
-D	Indicates that the command deals with the domain name entry.
-d	Deletes an entry in the system configuration database. It must be used with the -i IPAddress flag or the -n flag. The -i flag deletes a name server entry. The -n flag deletes the domain name entry.
-E <i>FileName</i>	Renames the system configuration database file, so you can stop using a name server. The /etc/resolv.conf file is moved to the file specified by the <i>FileName</i> variable.
-e	Moves the /etc/resolv.conf file to the /etc/resolv.conf.sv file, preventing use of a name server.
-I	(Uppercase i) Specifies that the -s flag or -X flag should print all name server entries.

Item	Description
-i <i>IPAddress</i>	Indicates that the command deals with a name server entry. Use dotted decimal format for the given IP address.
-l	(Lowercase L) Specifies that the operation is on the search list. Use this flag with the -d and -s flag.
-n	Specifies that the operation is on the domain name. Use this flag with the -d flag and the -s flag.
-S <i>SearchList</i>	Changes the search list in the system configuration database.
-s	Shows all domain and name server entries in the configuration system database. If you use the -i flag, the namerslv command shows all name server entries. If you use the -n flag, the namerslv command shows the domain name entry found in the database.
-X	Deletes all entries in the database. Use the -I flag with this flag to delete all name server entries.
-Z	Generates the output of the query in colon format. This flag is used when the namerslv command is called from the SMIT usability interface.

Examples

1. To add a domain entry with a domain name of `abc.aus.century.com`, type:

```
namerslv -a -D abc.aus.century.com
```

2. To change the `abc.aus.century.com` domain entry to the domain name `xyz.aus.century.com`, type:

```
namerslv xyz.aus.century.com
```

3. To add a name server entry with IP address `192.9.201.1`, type:

```
namerslv -a -i 192.9.201.1
```

4. To show all system configuration database entries related to domain name server information used by local resolver routines, type:

```
namerslv -s
```

The output is given in the following format:

```
domain xyz.aus.century.com
name server 192.9.201.1
```

5. To rename the `/etc/resolv.conf` file to stop using the name server and specify the new file name, `/etc/resolv.back`, type:

```
namerslv -E /etc/resolv.back
```

Files

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

Item	Description
/etc/html	

ncheck Command

Purpose

Generates path names from i-node numbers.

Syntax

```
ncheck [[ [ -a ] [ -i InNumber ... ] ] [ -s ] ] [-o Options] [ FileSystem ]
```

Description

The **ncheck** command displays the i-node number and path names for filesystem files. It uses question marks (??) displayed in the path to indicate a component that could not be found. Path names displayed with ... (ellipses) at the beginning indicate either a loop or a path name of greater than 10 entries. The **ncheck** command uses a simple hashing algorithm to reconstruct the path names that it displays. Because of this, it is restricted to filesystems with less than 50,000 directory entries.

Flags

Item	Description
-a	Lists the . (dot) and .. (dot dot) file names.
-i InNumber	Lists only the file or files specified by the <i>InNumber</i> parameter.
-o Options	Specifies a comma-separated list of implementation-specific options for a virtual file system. The following options are specific to the enhanced journaled file system (JFS2) -o snapshot=snapName): Specifies the name of the internal snapshot subject to the ncheck command. The file system owning the snapshot must be mounted.
-s	Lists only special files and files with set-user-ID 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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To list the i-node number and path name of each file in the default file systems, enter:

```
ncheck
```

2. To list all the files in a specified file system, enter:

```
ncheck -a /
```

This lists the i-node number and path name of each file in the / (root) file system, including the .(dot) and .. (dot dot) entries in each directory.

3. To list the name of a file when you know its i-node number, enter:

```
ncheck -i 690 357 280 /tmp
```

This lists the i-node number and path name for every file in the /tmp file system with i-node numbers of 690, 357, or 280. If a file has more than one link, all of its path names are listed.

4. To list special and set-user-ID files, enter:

```
ncheck -s /
```

This lists the i-node and path name for every file in the / (root) file system that is a special file (also called a device file) or that has set-user-ID mode enabled.

nddctl Command

Purpose

Issues commands to network device drivers (NDDs).

Syntax

```
nddctl { -r } Device
```

Description

The **nddctl** command allows the user to control an NDD device at runtime (that is, without having to reconfigure the device driver, which usually entails disruption to the network connection).

Flags

Item	Description
-r	Forces the NDD device to renegotiate its link attributes (speed and duplexity) at runtime. Note: Forcing link renegotiation entails resetting the device; this might cause a loss of network connectivity, lasting a few seconds, while the device re-initializes itself.

Parameters

Item	Description
<i>Device</i>	Specifies the NDD device on which to perform the specified command.

Exit Status

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

Examples

1. To force the device ent0 to renegotiate its link attributes at runtime, type:

```
nddctl -r ent0
```

Location

/usr/sbin

ndp Command

Purpose

IPv6 neighbor discovery display and control.

Syntax

ndp [**-n**] *hostname*

ndp [**-n**] **-a**

ndp [**-d**] *hostname* | *IpAddress*

ndp [**-i** *interface_index*] **-s** *hostname addr* [**temp**]

Description

The **ndp** program displays and modifies the IPv6-to-Ethernet, IPv6-to-TokenRing, or IPv6-to-InfiniBand address translation tables used by the IPv6 neighbor discovery protocol.

With no flags, the program displays the current **ndp** entry for *hostname*. The host may be specified by name or by number, using IPv6 textual notation.

Flags

Item	Description
-a	Displays all of the current ndp entries.
-d	Lets a super-user delete an entry for the host called <i>hostname</i> with the -d flag.
-i <i>interface_index</i>	Specifies the index of the interface to use when an ndp entry is added with the -s flag (useful with the local-link interface).
-n	Shows network addresses as numbers (normally ndp attempts to display addresses symbolically).
-s <i>hostname addr</i>	Creates an ndp entry for <i>hostname</i> with the Hardware address <i>addr</i> . The Hardware address is given as six hex bytes separated by colons. The entry is permanent unless the temp is specified in the command.

Examples

This is an example output from the **-a** flag:

```
# ndp -a
e-crankv6 (::903:9182) at link#2 0:20:af:db:b8:cf
e-crankv6-11 (fe80:0:100::20:afdb:b8cf) at link#2 0:20:af:db:b8:cf
e-crankv6-11 (fe80::2:c903:1:1e85) at link#5 SQP:0xe SLID0x49 DQP:0x48 DLID:0xf
0:48:fe80::2:c903:1:1e85 [InfiniBand]
# ndp -d e-crankv6-11
e-crankv6-11 (fe80:0:100::20:afdb:b8cf) deleted
# ndp -d fe80::2:c903:1:1e85
```

ndpd-host Daemon

Purpose

Neighbor Discovery Protocol (NDP) daemon for a host.

Syntax

```
ndpd-host [ -d] [ -v] [ -t] [ -c conffile][-T [ValidLifetime PreferredLifetime]] [ -g]
```

Description

The **ndpd-host** command manages the Neighbor Discovery Protocol (NDP) for nonkernel activities, such as Router Discovery, Prefix Discovery, Parameter Discovery, and Redirects. The **ndpd-host** command handles the default route, which includes the default router, the default interface, and the default interface address. However, the **ndpd-host** command does not overwrite the static default routes that are set on the host. When the daemon is stopped, the daemon cleans up the prefix addresses and the routes that are created during its lifetime.

Interfaces

The **ndpd-host** command knows about IEEE and CTI point to point interfaces. The **ndpd-host** command exchanges packets on all the known interfaces UP with a Link-Local Address. Any change of status of an interface is detected. If an interface goes down or loses its Link-Local address, the NDP processing is stopped on this interface. If an interface goes up, the NDP processing is started.

The IEEE interfaces are configured by using the **autoconf6** command. The PPP interfaces are configured by using the **pppd** daemon. The token negotiation defines the Link-Local addresses. To send the Router Advertisements over a CTI configured tunnel, it must have local and distant Link-Local addresses.

ndpd-host can generate Temporary Addresses as per RFC 4941. You can enable or disable temporary address generation for a particular prefix or interface by configuring the daemon in the `tempaddr.conf` file format. You can set the default preferred and valid lifetimes of Temporary Addresses by using the `-T` option.

Note: For all the up point to point interfaces, **ndpd-host** sets a local route through the `1o0` for local addresses.

Flags

Item	Description
<code>-c</code> <i>conf</i> file	Specifies the SEND configuration file. By default, the configuration file is the <code>/etc/ndpd/ndpdh.conf</code> file. To enable the SEND option, you must install the <code>clib.rte</code> fileset and OpenSSL.
<code>-d</code>	Enables debugging (exceptional conditions and dump).
<code>-g</code>	Allows the ndpd-host command to retain all the static global IPv6 address during initialization.
<code>-T</code> [<i>ValidLifetime PreferredLifetime</i>]	Enables Temporary Address generation. Along with <code>-T</code> flag, user can optionally specify default valid and preferred lifetimes for Temporary Addresses generated. By default, Temporary addresses are not generated, if this flag is not given.
<code>-t</code>	Adds a time stamp in each log.
<code>-v</code>	Logs all interesting events (<code>daemon.info</code> and console).

Signals

Item	Description
SIGUSR1	Turns on verbose.
SIGUSR2	Turns off verbose.
SIGINT	Dumps the current state of ndpd-host to syslog or stdout .
SIGTERM	Cleans up ndpd-host and exits.

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
/etc/ndpd/ndpdh.cnf	Specifies the SEND file locations.
/etc/ndpd/ cgaparams.sec	Specifies the configuration for each interface by using the SEND option.
/etc/ndpd/sendh_anchor	Specifies the trusted anchor values necessary for the SEND option.
/etc/ndpd/ tempaddr.conf	Specifies whether the generation of the Temporary Address for the router prefixes must be denied or allowed. The contents of the file are read only when ndpd-host is started with the -r flag.

ndpd-router Daemon

Purpose

NDP and RIPng daemon for a router.

Syntax

```
ndpd-router [ -r ] [ -p ] [ -M ] [ -O ] [ -s ] [ -q ] [ -g ] [ -n ] [ -R ] [ -S ] [ -d ] [ -t ] [ -v ] [ -H ] [ -m ] [ -u port ] [ -D  
max[min[/i>life]]] [ -P [invlife]/[deplife]] [ -T [reachtim]/[retrans]/[hlim]] [ -e [ off | compatible | only ] ]
```

Description

The **ndpd-router** daemon manages the Neighbor Discovery Protocol (NDP) for non-kernel activities. It receives Router Solicitations and sends Router Advertisements. It can also exchange routing information using the RIPng protocol.

The **/etc/gateway6** file provides options for **ndpd-router**. This file can be modified while the program is running. The changes are checked before any emission or reception of message, or on reception of the HUP signal. The file contains directives, one by line (with # as comment). All the IPv6 addresses and prefixes in the file must be in numeric form. No symbolic name is allowed. Except for the gateway directive, each line begins with a keyword and is made of options of the form *key=argument*.

Interfaces

The **ndpd-router** daemon knows about IEEE and CTI point to point interfaces. The **ndpd-router** daemon exchanges packets on all the known interfaces UP with a Link-Local Address. Any change of status of

an interface is detected. If an interface goes down or loses its Link-Local address, the NDP and RIPng processing is stopped on this interface. If an interface goes up, the NDP and RIPng processing is started.

To send Router Advertisements or RIPng packets or both, local *and* remote Link-Local addresses must be configured.

Flags

Item	Description
-e [off compatible only]	<p>Specifies the SEND mode:</p> <p>off</p> <p>Implies that the SEND option is not enabled. For example, the router behaves as is prior to RFC 3971/3972.</p> <p>compatible</p> <p>Implies that the router complies to RFC 3971/3972 but does not require the options specified in the RFC. The environment can be one where certain nodes are SEND capable while others are not. However, if the SEND options are embedded in the incoming packets, they must be correct.</p> <p>only</p> <p>Implies that all message must conform to RFC 3971/3972, or the message will be rejected.</p> <p>In order to enable the SEND option, you must install the clic.rte fileset and OpenSSL.</p>
-H	Enables the system to process NDP features needed to function as a mobile IPv6 home agent
-m	Enables the system to aid movement detection for mobile IPv6 mobile nodes.
-D <i>max</i> [<i>min</i> / <i>life</i>]	Sends Unsolicited Router Advertisements at intervals from <i>min</i> to <i>max</i> seconds. Default <i>max</i> value is 600 seconds, valid range is 4 to 1800 seconds. Default <i>min</i> equals to <i>max</i> / 3, valid range is from 1 to 0.75 * <i>max</i> . The router lifetime is set with <i>life</i> , default value is 10 * <i>max</i> . Valid range is 0 to 65535 seconds.
-T [<i>reachtim</i>] / [<i>retrans</i>] / [<i>hlim</i>]	Sets the BaseReachableTime field to <i>reachim</i> seconds, if <i>reachim</i> is not zero. If <i>retrans</i> is not zero, sets the RetransTime field to <i>retrans</i> seconds. If <i>hlim</i> is not zero, sets the hop limit field in Router Advertisements to <i>hlim</i> .
-M	Sets the M flag (stateful configuration) in advertisements.
-O	Sets the O flag (other stateful information) in advertisements
-p	Does not offer prefixes (learned from interface configuration).
-P [<i>invlife</i>] / [<i>deplife</i>]	Sets the invalid life value and the deprecated life value for announced prefixes (in seconds). The default value is 0xffffffff (infinite).
-r	Does not offer to be the default router in Router Advertisements.
-s	Enables the RIPng protocol (the default is: RIPng disabled).
-q	Enables the RIPng protocol, but does not send RIPng packets.
-g	Broadcast a default route in RIPng.
-n	Does not install routes received by RIPng.
-u <i>port</i>	Uses UDP port <i>port</i> for RIPng. The default is 521.
-R	Uses split horizon without corrupting reverse for RIPng.
-S	Does not use any split horizon for RIPng.
-d	Enables debugging (exceptional conditions and dump).

Item	Description
-v	Logs all interesting events (daemon.info and console).
-t	Adds time stamps in logged messages.

Available directives

The main directives for the `/etc/gateway6` file are:

option [*option-directive ...*]

Sets per-interface/default options.

prefix [*prefix-directive ...*]

Sets per-interface/default prefix processing options.

filter [*filter-directive ...*]

Sets per-interface/default filters.

gateway directives

Sets routes in RIPng packets or in the kernel.

Each of these directives is explained in more detail below.

The option directive

Sets different per-interface options.

Any value settings for the **option** directive which follow the **if** option must appear in a comma-separated list.

Note: At least one option (other than the **if** option) must be specified following the **option** directive. If the **if** option is specified, it must be the first option following the **option** directive. There must be a space between the **if** option and any comma-separated list of options which follow.

Syntax:

option [**if**=*n1,n2*] **ripin**=(*y|n*),**ripout**=(*y|n|S|R*),**rtadv**=(*y|n|min[/max]*),**flag**=[**M|O**],**life**=*Seconds*,**reach**=*Seconds*,**retrans**=*Seconds*

Item	Description
if = <i>list</i> interface = <i>list</i>	If there is no keyword, the option directive is a default option. If there is an interface field, the option parameters apply only to the listed interfaces. The list is comma-separated. You can use <code>1e*</code> to match all the <code>1eX</code> interfaces. The default option must be the first line in the <code>/etc/gateway6</code> file.
mtu [= <i>mtuval</i>]	Advertises a MTU value of <i>mtuval</i> in router advertisements. If there is no <i>mtuval</i> argument, the advertised MTU is the MTU of the interface. If <i>mtuval</i> is 0, suppress the advertisement of MTU.
ripin =(<i>n y</i>)	Does not listen (listen) to incoming RIPng packets. Does not send (send) RIPng packets. With the -S flag, do not use split horizon. With the -R flag, use split horizon without poisoning reverse.
rtadv =(<i>n y min[/max]</i>)	Does not send (send) router advertisements. With <i>min[/max]</i> option, set the interval (in seconds) between router advertisements.

Item	Description
flag ={ M O }	Sets the stateful mode flags in router advertisements. M Uses stateful configuration O Uses stateful configuration, but not for addresses
life = <i>Seconds</i>	Sets the router life field in router advertisements (in seconds).
reach = <i>Seconds</i>	Sets the reachable field in router advertisements (in seconds).
retrans = <i>Seconds</i>	Sets the retransmit interval field in router advertisements (in seconds).

The prefix directive

Defines the prefixes announced in Router advertisement directives. If there is no prefix-directive for an interface, the router advertisement contains the list of prefixes deduced from the address list of the interface. If there are prefix-directives, the router advertisement contains the list of prefixes defined by the different prefix directives (in order). No prefix is installed in the kernel. If there is one directive of the form `prefix prefix=none`, no prefix list is advertised.

Syntax:

prefix if=*n* **prefix**=(**none**|*xxx::/PrefixLength*) **flag**=[**L**][**A**] **valid**=*Seconds* **deprec**=*Seconds*

Item	Description
if = <i>Interface</i> or interface = <i>Interface</i>	Specifies the interface on which the directive applies. The if keyword is mandatory for the prefix directive. It is not an option.
prefix = <i>xxx::/PrefixLength</i>	The advertised prefix.
flag =[L][A]	Set the L and/or A flag for the prefix (the default is LA).
deprec = <i>Seconds</i>	Set the deprecated time (in seconds) for the prefix.
valid = <i>Seconds</i>	Set the validity time (in seconds) for the prefix.

The filter directive

Define a filter pattern for incoming (**filter=in**) or outgoing (**filter=out**) RIPng packets. There is one incoming and one outgoing filter per interface, and one default incoming and one default outgoing filter for interfaces without explicit filter.

Any received RIPng information is tested against the input filter of the interface, or, if there is none, against the default input filter. The static interface routes are seen as input information coming from the interface and from a gateway with the link local address of the interface. The routes set by a gateway directive with a **gateway** keyword are seen as input information coming from the specified interface and gateway. The default route (**-g** flag) and the routes set by a gateway directive without a **gateway** keyword are seen as input information coming from gateway :: and no interface (the default input filter applies).

Any sent RIPng information is tested against the output filter of the interface, or, if there is none, against the default output filter.

Each filter is a sequence of matching patterns. The patterns are tested in order. Each pattern can test the prefix length, the source gateway (for input filters and that the prefix (padded with zeroes) matches a

fixed prefix. If a pattern contains more than one test description, the match is the conjunction of all the tests. The first matching pattern defines the action to perform. If no pattern matches, the default action is accept. The possible actions are accept, reject and truncate/*NumberOfBits*. The truncate/*NumberOfBits* action means: if the pattern matches and if prefix length is greater or equal to *NumberOfBits*, accept the prefix with new length *NumberOfBits*. The accepted prefix is immediately accepted, that is, not checked again against the filters.

For example, the following directive inhibits sending host routes on any interface without an explicit outgoing filter:

```
filter=out length==128 action=reject
```

Syntax:

filter=(in|out) [if=*n1,n2*] **prefix**=*xxx::/NumberOfBits* **gateway**=*xxx* **length**=(|=|>|=|<|>)*NumberOfBits*
action=(accept|reject|truncate/*xx*)

Item	Description
if = <i>list</i> or interface = <i>list</i>	If there is no interface keyword, the filter directive is a default option. If there is an interface field, the filter pattern is added at the end of the filters of all specified interfaces. The list is comma-separated. For example, you can specify interface =le* to specify all the leX interfaces.
prefix = <i>xxx::/NumberOfBits</i>	The pattern matches only if <i>xxx::/NumberOfBits</i> is a prefix of the prefix in the RIPng packet.
gateway = <i>xxx</i>	The pattern matches only if the RIPng message comes from source address <i>xxx</i> , only in incoming filters.
length =(= > = < >) <i>NumberOfBits</i>	The pattern match only if the prefix length in the RIPng message is equal to (or greater than, less than, etc., depending on the operator specified) to <i>NumberOfBits</i> .
action =(accept reject truncate/ <i>NumberOfBits</i>)	Specify the action to perform if the pattern matches: accept the message, reject the message, accept but truncate the prefix to <i>NumberOfBits</i> bits.

Gateway directives

The gateway directives allow the user to set up routes in RIPng packets and/or in the kernel. These directives must appear at the end of the **/etc/gateway6** file, after the other directives.

Syntax:

```
xxx::/NumberOfBits metric Value
```

```
xxx::/NumberOfBits metric Value gateway IPv6Address ifname
```

The second syntax is used to add the route to the kernel.

Examples

The following examples are of the **/etc/gateway6** file.

On a site where all addresses are of the form 5f06:2200:c001:0200:xxxx, the following example means that only one route, describing all the site, is exported on all the Configured Tunnel Interface (CTI) **ctiX** interfaces. The keyword abbreviations shown are valid.

```
filter=out if=cti* pref=5f06:2200:c001:0200::/64 len=>=64 act=trunc/64
```

Setting a default outgoing route:

```
::/0 metric 2 gateway 5f06:2200:c102:0200::1 cti0
```

Declare that any CTI interface active with RIPng defines a default route:

```
filter=in if=cti* act=trunc/0
```

The following example defines a site with an exterior connection `cti0`, which aggregates other sites connected through `ctiX`, and which uses split horizon without poisoned reverse. The order of the lines is important, as all filter descriptions apply to `cti0`.

```
option if=cti* ripout=R
filter=out if=cti0 prefix=5f06:2200::/24 len=>=24 act=trunc/24
filter=out if=cti* pref=5f06:2200:c001:0200::/64 len=>=64 act=trunc/64
filter=in if=cti0 act=trunc/0
filter=in if=cti* prefix=5f06:2200::/24 len=>=24 act=trunc/64
filter=in if=cti* act=reject
```

Diagnostics

All errors are logged at the **daemon.err** level, unless the debug option is set. This includes all the syntax errors in the **/etc/gateway6** file and configuration mismatches between different routers.

Signals

ndpd-router responds to the following signals:

Item	Description
SIGINT	Dumps its current state to syslog, if syslog is defined. Otherwise, dumped to stdout.
SIGHUP	The /etc/gateway6 file is read again.
SIGUSR1	Verbosity is incremented.
SIGUSR2	Verbosity is reset.
SIGTERM	Resets to a reasonable state and stops.
SIGQUIT	Resets to a reasonable state and stops.

Files

Item	Description
/etc/gateway6	
/etc/ndpd/sendr_anchor	The SEND router anchor file for the certificate chain.

ndx Command

Purpose

Creates a subject-page index for a document.

Syntax

```
ndx [ SubjectFile ] " FormatterCommandLine "
```

Description

The **ndx** command, given a list of subjects (*SubjectFile*), searches a specified English-language document and writes a subject-page index to standard output.

The document must include formatting directives for the **mm**, **mmt**, **nroff**, or **troff** commands. The formatter command line informs the **ndx** command whether the **troff** command, **nroff** command, **mm** command, or **mmt** command can be used to produce the final version of the document. These commands do the following:

Item	Description
troff or mmt	Specifies the troff command as the formatting program.
nroff or mm	Specifies the nroff command as the formatting program.

Parameters

Item	Description
<i>SubjectFile</i>	<p>Specifies the list of subjects that are included in the index. Each subject must begin on a new line and have the following format:</p> <pre>word1[word2...][,wordk...]</pre> <p>For example:</p> <pre>printed circuit boards arrays arrays, dynamic storage Smith, W.P. printed circuit boards, channel-oriented multi-layer Aranoff University of Illinois PL/1</pre>
<i>FormatterCommandLine</i>	<p>The subject must start in column one.</p> <p>Creates the final form of the document. The syntax for this parameter is as follows:</p> <pre>Formatter [Flag...] File...</pre> <pre>mm -Tlp File(s) nroff -mm -Tlp -rW60 File(s) troff -rB2 -Tibm3816 -r01.5i File(s)</pre> <p>For more information on the formatter command line, see the mmt command, nroff command, and html</p>

neqn Command

Purpose

Formats mathematical text for the **nroff** command.

Syntax

```
neqn [ -d Delimiter1Delimiter2 ] [ -f Font ] [ -p Number ] [ -s Size ] [ - ] [ File ... | - ]
```

Description

The **neqn** command is an **nroff** preprocessor for formatting mathematical text on typewriter-like terminals. Pipe the output of the **neqn** command into the **nroff** command as follows:

```
neqn [Flag...] File... | nroff [Flag...] | [Printer]
```

The **neqn** command reads one or more files. If no files are specified for the *File* parameter or the - (minus sign) flag is specified as the last parameter, standard input is read by default. A line beginning with the **.EN** macro. These lines are not altered by the **nroff** command, so they can be defined in macro packages to provide additional formatting functions such as centering and numbering.

The — (double dash) delimiter indicates the end of flags.

Depending on the target output devices, **neqn** command output formatted by the **nroff** command may need to be post-processed by the **eqn** command gives more information about the input format and keywords used.

Flags

Item	Description
-d <i>Delimiter1Delimiter2</i>	Sets two ASCII characters, <i>Delimiter1</i> and <i>Delimiter2</i> , as delimiters of the text to be processed by the neqn command, in addition to input enclosed by the .EQ and .EN macros. The text between these delimiters is treated as input to the neqn command. Within a file, you can also set delimiters for neqn text using the delim <i>Delimiter1Delimiter2</i> request. These delimiters are turned off by the delim off request. All text that is not between delimiters or the .EN macro is passed through unprocessed.
-f <i>Font</i>	Changes font in all the neqn command-processed text to the value specified by the <i>Font</i> variable. The <i>Font</i> value (a font name or position) must be one or two ASCII characters.
-p <i>Number</i>	Reduces subscripts and superscripts to the specified number of points in size. The default is 3 points.
-s <i>Size</i>	Changes point size in all the neqn command-processed text to the value specified by the <i>Size</i> variable.
-	Reads from standard input.
—	(double dash) Marks the end of the flags.

Files

Item	Description
/usr/share/lib/pub/eqnchar	Contains special character definitions.

netcd Daemon

Purpose

Launches the network caching (netcd) daemon.

Syntax

```
netcd [ -l file ] [ -c file ] [ -d level ] [ -h ]
```

Description

The **netcd** daemon reduces the time taken by the local, DNS, NIS, and user loadable module services to respond to a query by caching the response retrieved from resolvers.

When the **netcd** daemon is running and configured for a resolver (for example, DNS) and a map (for example, hosts), the resolution is first made using the cached answers. If it fails, the resolver is called and the response is cached by the **netcd** daemon.

The type of the maps that are supported for the local, NIS, and user loadable modules resolutions are hosts, services, networks, protocols and netgroup. For DNS, hosts is the only type of map that you can use.

In addition, for the specific case of Yellow Pages, the following maps have been added:

- passwd.byname
- passwd.byuid
- group.byname
- group.bygid
- netid.byname
- passwd.adjunct.byname

You can use a configuration file to specify the resolvers and maps that you want to configure. You can also set other **netcd** parameters using this file. By default, the configuration file used is the **/etc/netcd.conf** file. You can change the path of this configuration file using the **-c** argument of the **netcd** daemon. If the **/etc/netcd.conf** file does not exist, the **netcd** daemon uses the default parameters. You can find a sample of this file under the **/usr/samples/tcpip** file. Do not use this file as a configuration file because it will be overwritten by a new installation of the package containing the file.

You can specify the level of debugging using the **-d** argument. The debugging levels are similar to the one used by the **syslogd** daemon. Log messages are written to the **/var/tmp/netcd.log** file. You can override the default using the netcd configuration file. As with the **syslogd** daemon, you can specify rotation for the netcd log file.

netcd Parameters

When an entry is inserted in a netcd cache, a time-to-live (TTL) is associated to it. You can configure this TTL using the netcd configuration file (cache declarations). For DNS, this TTL is the one contains the response from the DNS.

To clean the caches of outdated entries, you must run two tasks periodically, one to clean local caches and the other to clean the other caches. You can set the frequency of these tasks using the *local_scan_frequency* and *net_scan_frequency* parameters in the netcd configuration file.

Caches are hashed tables. The size of the hash tables can be controlled using the netcd configuration file and the **netcdctrl** command.

To communicate between the applications, the **netcd** daemon uses a socket (**/dev/netcd**). You can configure the size of the message queue using the netcd configuration file.

netcd supports the System Resource Controller

The **netcd** daemon is part of the netcd System Resource Controller (SRC) group. The following are the SRC commands you can use to manage the **netcd** daemon:

- You can start the **netcd** daemon using the **startsrc** command, or stop the **netcd** daemon using the **stopsrc** command.
- The **lssrc** command provides a short status output that includes the Process ID (PID) and the status of the **netcd** daemon.
- The **lssrc -l** command provides a long status output that includes the PID, the status of the **netcd** daemon, the configuration file used when starting the **netcd** daemon, and the configured caches.

Note: You cannot use the **refresh** command with the **netcd** daemon.

Flags

Item	Description
-c <i>file</i>	Specifies a configuration file. The default file name is /etc/netcd.conf .
-d <i>level</i>	Specifies the logging level. The <i>level</i> value must be an integer between 0 and 7.
-h	Displays help information.
-l <i>file</i>	Loads caches from the specified binary file created by the netcdctrl command. The local files (for example, /etc/hosts , /etc/services) are loaded depending on the configuration file.

Examples

1. To launch the **netcd** daemon using the SRC, enter:

```
startsrc -s netcd
```

2. To display the status of the **netcd** daemon using the SRC, enter:

```
lssrc -s netcd
```

This command produces the following output:

Subsystem	Group	PID	Status
netcd	netcd	299064	active

3. To display the status of the **netcd** daemon in long form using the SRC, enter:

```
lssrc -l -s netcd
```

This command produces the following output:

Subsystem	Group	PID	Status
netcd	netcd	299064	active
Configuration File	/etc/netcd.conf		
Configured Cache	local services		
Configured Cache	local protocols		
Configured Cache	local hosts		
Configured Cache	local networks		
Configured Cache	local netgroup		

4. To launch the **netcd** daemon without using the SRC, enter:

```
netcd
```

netcdctrl Command

Purpose

Manages the network caching (netcd) daemon caches.

Syntax

```
netcdctrl [ -t type -e type [ -a file | -b file | -f | -s file ] [ -l level ] [ -h ]
```

Description

The **netcdctrl** command provides the following functions:

- Dumps specific caches in ASCII format: provides a readable output of the caches content.

- Dumps specific caches in binary format. The binary format can be used later to reload the caches when starting the **netcd** daemon. Dumping avoids reloading the caches from the beginning.
- Displays statistics on caches use. The caches are tables, and the access to these tables is controlled by a hash algorithm. This output helps you size the table for a given resolution and a given map using the **netcd** configuration file.
- Flushes specific caches. The content of the specified caches are erased, and local caches are then reloaded. Other caches are reloaded by resolver's responses.
- Changes the logging level dynamically.

Requirement: You must have the root authority to issue the **netcdctrl** command.

Flags

Item	Description
-a <i>file</i>	Specifies ASCII dumping of the specified caches.
-b <i>file</i>	Specifies binary dumping of the specified caches (local caches are not dumped).
-e <i>type</i>	Specifies the map. The <i>type</i> parameter can be one of the following values: <ul style="list-style-type: none"> • hosts • protocols • servers • networks • netgroup • a yellow pages map name (for example passwd.byname or group.bygid) • all <p>Use this flag only with the -b, -a, -f and -s flags.</p>
-f	Flushes the specified caches.
-h	Displays help information.
-l <i>level</i>	Changes the logging level of the netcd daemon. The <i>level</i> value must be an integer of 0 through 7.
-s <i>file</i>	Provides statistics on caches use.
-t <i>type</i>	Specifies the resolution. The <i>type</i> parameter can be one of the following values: <ul style="list-style-type: none"> • local • dns • nis • yp • ulm • a specific module name as provided in the netcd.conf file • all <p>Use his flag only with the -b, -a, -f and -s flags.</p>

Examples

1. To flush all the caches, enter:


```
netcdctl -t all -e all -f
```

2. To dump all the NIS caches in binary format, enter:

```
netcdctl -t nis -e all -b /tmp/netcd_nis_binary_dump
```

3. To dump the local cache for hosts in ASCII format, enter:

```
netcdctl -t local -e hosts -a /tmp/netcd_dns_hosts
```

4. To set the level of logging to obtain all possible traces, enter:

```
netcdctl -l 7
```

netpmon Command

Purpose

Monitors activity and reports statistics on network I/O and network-related CPU usage.

Syntax

```
netpmon [ -o File ] [ -d ] [ -T n ] [ -P ] [ -t ] [ -v ] [ -r PURR ] [ -O ReportType ... ] [ -i Trace_File -n Gensyms_File ] [ -@ [WparList | ALL] ]
```

Description

The **netpmon** command monitors a trace of system events, and reports on network activity and performance during the monitored interval. By default, the **netpmon** command runs in the background while one or more application programs or system commands are being executed and monitored. The **netpmon** command automatically starts and monitors a trace of network-related system events in real time. By default, the trace is started immediately; optionally, tracing may be deferred until the user issues a **trcon** command. When tracing is stopped by a **trcstop** command, the **netpmon** command generates all specified reports and exits.

The **netpmon** command can also work in offline mode, that is, on a previously generated trace file. In this mode, a file generated by the **gensyms** command is also required. The **gensyms** file should be generated immediately after the trace has been stopped, and on the same machine. When running in offline mode, the **netpmon** command cannot recognize protocols used by sockets, which limits the level of detail available in the socket reports.

The **netpmon** command reports on the following system activities:

CPU Usage

The **netpmon** command monitors CPU usage by all threads and interrupt handlers. It estimates how much of this usage is due to network-related activities.

Network Device-Driver I/O

The **netpmon** command monitors I/O operations through token-ring and Fiber-Distributed Data Interface (FDDI) network device drivers. In the case of transmission I/O, the command also monitors utilizations, queue lengths, and destination hosts. For receive ID, the command also monitors time in the demux layer.

Internet Socket Calls

The **netpmon** command monitors all **send**, **recv**, **sendto**, **recvfrom**, **read**, and **write** subroutines on Internet sockets. It reports statistics on a per-process basis, for each of the following protocol types:

- Internet Control Message Protocol (ICMP)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

NFS I/O

The **netpmon** command monitors **read** and **write** subroutines on client Network File System (NFS) files, client NFS remote procedure call (RPC) requests, and NFS server read or write requests. The command reports subroutine statistics on a per-process or optional per-thread basis and on a per-file basis for each server. The **netpmon** command reports client RPC statistics for each server, and server read and write statistics for each client.

Any combination of the preceding report types can be specified with the command line flags. By default, all the reports are produced.

Notes: The reports produced by the **netpmon** command can be quite long. Consequently, the **-o** flag should usually be used to write the report to an output file. The **netpmon** command obtains performance data using the system trace facility. The trace facility only supports one output stream. Consequently, only one **netpmon** or **trace** process can be active at a time. If another **netpmon** or **trace** process is already running, the **netpmon** command responds with the message:

```
/dev/systrace: Device busy
```

While monitoring very network-intensive applications, the **netpmon** command may not be able to consume trace events as fast as they are produced in real time. When that happens, the error message:

```
Trace kernel buffers overflowed, N missed entries
```

displays on standard error, indicating how many trace events were lost while the trace buffers were full. The **netpmon** command continues monitoring network activity, but the accuracy of the report diminishes by some unknown degree. One way to avoid overflow is to increase the trace buffer size using the **-T** flag, to accommodate larger bursts of trace events before overflow. Another way to avoid overflow problems all together is to run **netpmon** in offline mode.

When running in memory-constrained environments (where demand for memory exceeds supply), the **-P** flag can be used to pin the text and data pages of the real-time **netpmon** process in memory so the pages cannot be swapped out. If the **-P** flag is not used, allowing the **netpmon** process to be swapped out, the progress of the **netpmon** command may be delayed such that it cannot process trace events fast enough to prevent trace buffer overflow.

If the **/unix** file and the running kernel are not the same, the kernel addresses will be incorrect, causing the **netpmon** command to exit.

Flags

Item	Description
-d	Starts the netpmon command, but defers tracing until the trcon command has been executed by the user. By default, tracing is started immediately.
-i <i>Trace_File</i>	Reads trace records from the file <i>Trace_File</i> produced with the trace command instead of a live system. The trace file must be rewritten first in raw format using the trcpt -r command. This flag cannot be used without the -n flag.
-n <i>Gensyms_File</i>	Reads necessary mapping information from the file <i>Gensyms_File</i> produced by the gensyms command. This flag is mandatory when the -i flag is used.
-o <i>File</i>	Writes the reports to the specified <i>File</i> , instead of to standard output.

Item	Description
-O ReportType ...	<p>Produces the specified report types. Valid report type values are:</p> <p>cpu CPU usage</p> <p>dd Network device-driver I/O. This report is not available inside a workload partition (WPAR) in online mode or in the global WPAR with the '-@ <i>WparList</i>' flag.</p> <p>so Internet socket call I/O</p> <p>nfs NFS I/O (any version)</p> <p>nfs2 NFS Version 2 I/O</p> <p>nfs3 NFS Version 3 I/O</p> <p>nfs4 NFS Version 4 I/O</p> <p>all All reports are produced. This is the default value when the netpmon command is run in the global WPAR without the -@ flag.</p>
-P	<p>Pins monitor process in memory. This flag causes the netpmon text and data pages to be pinned in memory for the duration of the monitoring period. This flag can be used to ensure that the real-time netpmon process does not run out of memory space when running in a memory-constrained environment.</p>
-r PURR	<p>Uses PURR time instead of TimeBase in percent and CPU time calculation. Elapsed time calculations are unaffected.</p>
-t	<p>Prints CPU reports on a per-thread basis.</p>
-T n	<p>Sets the kernel's trace buffer size to <i>n</i> bytes. The default size is 64000 bytes. The buffer size can be increased to accommodate larger bursts of events, if any. (A typical event record size is on the order of 30 bytes.)</p> <p style="padding-left: 40px;">Note: The trace driver in the kernel uses double buffering, so actually two buffers of size <i>n</i> bytes will be allocated. These buffers are pinned in memory, so they are not subject to paging.</p>
-v	<p>Prints extra information in the report. All processes and all accessed remote files are included in the report instead of only the 20 most active processes and files.</p>
-@ [WparList ALL]	<p>Specifies that reports are limited to the list of WPARs that are passed as an argument.</p>

Reports

The reports generated by the **netpmon** command begin with a header, which identifies the date, the machine ID, and the length of the monitoring period in seconds. This is followed by a set of summary and detailed reports for all specified report types.

CPU Usage Reports

Process CPU Usage Statistics: Each row describes the CPU usage associated with a process. Unless the verbose option is specified, only the 20 most active processes are listed. At the bottom of the report, CPU usage for all processes is totaled, and CPU idle time is reported.

Process

Process name

PID

Process ID number

CPU Time

Total amount of CPU time used by this process

CPU %

CPU usage for this process as a percentage of total time

Network CPU %

Percentage of total time that this process spent executing network-related code

Thread CPU Usage Statistics

If the **-t** flag is used, each process row described above is immediately followed by rows describing the CPU usage of each thread owned by that process. The fields in these rows are identical to those for the process, except for the name field. (Threads are not named.)

First-Level Interrupt Handler Usage Statistics: Each row describes the CPU usage associated with a first-level interrupt handler (FLIH). At the bottom of the report, CPU usage for all FLIHs is totaled.

FLIH

First-level interrupt handler description

CPU Time

Total amount of CPU time used by this FLIH

CPU %

CPU usage for this interrupt handler as a percentage of total time

Network CPU %

Percentage of total time that this interrupt handler executed on behalf of network-related events

Second-Level Interrupt Handler Usage Statistics: Each row describes the CPU usage associated with a second-level interrupt handler (SLIH). At the bottom of the report, CPU usage for all SLIHs is totaled.

SLIH

Second-level interrupt handler description

CPU Time

Total amount of CPU time used by this SLIH

CPU %

CPU usage for this interrupt handler as a percentage of total time

Network CPU %

Percentage of total time that this interrupt handler executed on behalf of network-related events

Summary Network Device-Driver Reports

Network Device-Driver Statistics (by Device): Each row describes the statistics associated with a network device.

Device

Path name of special file associated with device

Xmit Pkts/s

Packets per second transmitted through this device

Xmit Bytes/s

Bytes per second transmitted through this device

Xmit Util

Busy time for this device, as a percent of total time

Xmit Qlen

Number of requests waiting to be transmitted through this device, averaged over time, including any transaction currently being transmitted

Recv Pkts/s

Packets per second received through this device

Recv Bytes/s

Bytes per second received through this device

Recv Demux

Time spent in demux layer as a fraction of total time

Network Device-Driver Transmit Statistics (by Destination Host): Each row describes the amount of transmit traffic associated with a particular destination host, at the device-driver level.

When hosts are on the same subnet, the destination host name is displayed. When hosts are in a different subnet, the destination host can be bridges, routers, or gateways as resolved by ARP protocol.

Host

Destination host name. An * (asterisk) is used for transmissions for which no host name can be determined.

Pkts/s

Packets per second transmitted to this host

Xmit Bytes/s

Bytes per second transmitted to this host

Summary Internet Socket Reports

- *On-line mode:* **Socket Call Statistics for Each Internet Protocol (by Process):** Each row describes the amount of **read/write** subroutine activity on sockets of this protocol type associated with a particular process. Unless the verbose option is specified, only the top 20 processes are listed. At the bottom of the report, all socket calls for this protocol are totaled.
- *Off-line mode:* **Socket Call Statistics for Each Process:** Each row describes the amount of **read/write** subroutine activity on sockets associated with a particular process. Unless the verbose option is specified, only the top 20 processes are listed. At the bottom of the report, all socket calls are totaled.

Process

Process name

PID

Process ID number

Read Calls/s or Read Ops/s

Number of **read**, **recv**, and **recvfrom** subroutines per second made by this process on sockets of this type

Read Bytes/s

Bytes per second requested by the above calls

Write Calls/s or Write Ops/s

Number of **write**, **send**, and **sendto** subroutines per second made by this process on sockets of this type

Write Bytes/s

Bytes per second written by this process to sockets of this protocol type

Summary NFS Reports

NFS Client Statistics for Each Server (by File): Each row describes the amount of **read/write** subroutine activity associated with a file mounted remotely from this server. Unless the verbose option is specified, only the top 20 files are listed. At the bottom of the report, calls for all files on this server are totaled.

File

Simple file name

Read Calls/s or Read Ops/s

Number of **read** subroutines per second on this file

Read Bytes/s

Bytes per second requested by the above calls

Write Calls/s or Write Ops/s

Number of **write** subroutines per second on this file

Write Bytes/s

Bytes per second written to this file

NFS Client RPC Statistics (by Server): Each row describes the number of NFS remote procedure calls being made by this client to a particular NFS server. At the bottom of the report, calls for all servers are totaled.

Server

Host name of server. An * (asterisk) is used for RPC calls for which no hostname could be determined.

Calls/s or Ops/s

Number of NFS RPC calls per second being made to this server.

NFS Client Statistics (by Process): Each row describes the amount of NFS **read/write** subroutine activity associated with a particular process. Unless the verbose option is specified, only the top 20 processes are listed. At the bottom of the report, calls for all processes are totaled.

Process

Process name

PID

Process ID number

Read Calls/s or Read Ops/s

Number of NFS **read** subroutines per second made by this process

Read Bytes/s

Bytes per second requested by the above calls

Write Calls/s or Write Ops/s

Number of NFS **write** subroutines per second made by this process

Write Bytes/s

Bytes per second written to NFS mounted files by this process

NFS Server Statistics (by Client): Each row describes the amount of NFS activity handled by this server on behalf of particular client. At the bottom of the report, calls for all clients are totaled.

Client

Host name of client

Read Calls/s or Read Ops/s

Number of remote read requests per second processed on behalf of this client

Read Bytes/s

Bytes per second requested by this client's read calls

Write Calls/s or Write Ops/s

Number of remote write requests per second processed on behalf of this client

Write Bytes/s

Bytes per second written by this client

Other Calls/s or Ops/s

Number of other remote requests per second processed on behalf of this client

Detailed Reports

Detailed reports are generated for any of the specified report types. For these report types, a detailed report is produced for most of the summary reports. The detailed reports contain an entry for each entry in the summary reports with statistics for each type of transaction associated with the entry.

Transaction statistics consist of a count of the number of transactions of that type, followed by response time and size distribution data (where applicable). The distribution data consists of average, minimum, and maximum values, as well as standard deviations. Roughly two-thirds of the values are between average - standard deviation and average + standard deviation. Sizes are reported in bytes. Response times are reported in milliseconds.

Detailed Second Level Interrupt Handler CPU Usage Statistics:

SLIH

Name of second-level interrupt handler

Count

Number of interrupts of this type

CPU Time (Msec)

CPU usage statistics for handling interrupts of this type

Detailed Network Device-Driver Statistics (by Device):

Device

Path name of special file associated with device

Recv Packets

Number of packets received through this device

Recv Sizes (Bytes)

Size statistics for received packets

Recv Times (msec)

Response time statistics for processing received packets

Xmit Packets

Number of packets transmitted to this host

Demux Times (msec)

Time statistics for processing received packets in the demux layer

Xmit Sizes (Bytes)

Size statistics for transmitted packets

Xmit Times (Msec)

Response time statistics for processing transmitted packets

Detailed Network Device-Driver Transmit Statistics (by Host):

Host

Destination host name

Xmit Packets

Number of packets transmitted through this device

Xmit Sizes (Bytes)

Size statistics for transmitted packets

Xmit Times (Msec)

Response time statistics for processing transmitted packets

Detailed Socket Call Statistics for Each Internet Protocol (by Process): (*on-line mode*) **Detailed Socket Call Statistics for Each Process:** (*off-line mode*)

Process

Process name

PID

Process ID number

Reads

Number of **read**, **recv**, **recvfrom**, and **recvmsg** subroutines made by this process on sockets of this type

Read Sizes (Bytes)

Size statistics for **read** calls

Read Times (Msec)

Response time statistics for **read** calls

Writes

Number of **write** , **send** , **sendto** , and **sendmsg** subroutines made by this process on sockets of this type

Write Sizes (Bytes)

Size statistics for **write** calls

Write Times (Msec)

Response time statistics for **write** calls

Detailed NFS Client Statistics for Each Server (by File):**File**

File path name

Reads

Number of NFS **read** subroutines for this file

Read Sizes (Bytes)

Size statistics for **read** calls

Read Times (Msec)

Response time statistics for **read** calls

Writes

Number of NFS **write** subroutines for this file

Write Sizes (Bytes)

Size statistics for **write** calls

Write Times (Msec)

Response time statistics for **write** calls

Detailed NFS Client RPC Statistics (by Server):**Server**

Server host name

Calls

Number of NFS client RPC calls made to this server

Call Times (Msec)

Response time statistics for RPC calls

Detailed NFS Client Statistics (by Process):**Process**

Process name

PID

Process ID number

Reads

Number of NFS **read** subroutines made by this process

Read Sizes (Bytes)

Size statistics for **read** calls

Read Times (Msec)

Response time statistics for **read** calls

Writes

Number of NFS **write** subroutines made by this process

Write Sizes (Bytes)

Size statistics for **write** calls

Write Times (Msec)

Response time statistics for **write** calls

Detailed NFS Server Statistics (by Client):

Client

Client host name

Reads

Number of NFS read requests received from this client

Read Sizes (Bytes)

Size statistics for read requests

Read Times (Msec)

Response time statistics for read requests

Writes

Number of NFS write requests received from this client

Write Sizes (Bytes)

Size statistics for write requests

Write Times (Msec)

Response time statistics for write requests

Other Calls

Number of other NFS requests received from this client

Other Times (Msec)

Response time statistics for other requests

Examples

1. To monitor network activity during the execution of certain application programs and generate all report types, type:

```
netpmon  
<run application programs and commands here>  
trcstop
```

The **netpmon** command automatically starts the system trace and puts itself in the background. Application programs and system commands can be run at this time. After the **trcstop** command is issued, all reports are displayed on standard output.

2. To generate CPU and NFS report types and write the reports to the `nmon.out` file, type:

```
netpmon -o nmon.out -O cpu,nfs  
<run application programs and commands here>  
trcstop
```

The **netpmon** command immediately starts the system trace. After the **trcstop** command is issued, the I/O activity report is written to the `nmon.out` file. Only the CPU and NFS reports will be generated.

3. To generate all report types and write verbose output to the `nmon.out` file, type:

```
netpmon -v -o nmon.out  
<run application programs and commands here>  
trcstop
```

With the verbose output, the **netpmon** command indicates the steps it is taking to start up the trace. The summary and detailed reports include all files and processes, instead of just the 20 most active files and processes.

4. To use the **netpmon** command in offline mode, type:

```
trace -a  
run application programs and commands here  
trcoff  
gensyms > gen.out  
trcstop  
netpmon -i tracefile -n gen.out -o netpmon.out
```

netrule Command

Purpose

Adds, removes, lists, or queries rules, flags and security labels for interfaces and hosts.

Syntax

netrule hl [**i** | **o** | **io**]

netrule hq { **i** | **o** } *src_host_rule_specification* *dst_host_rule_specification*

netrule h- [**i** | **o**][**u**] [*src_host_rule_specification* *dst_host_rule_specification*]

netrule h+ { **i** | **o** } [**u**] *src_host_rule_specification* *dst_host_rule_specification* [*flags*][*RIPSO/CIPSO options*] *security_label_information*

netrule il

netrule iq *interface*

netrule i- [**u**][*interface*]

netrule i+ [**u**] *interface* [*flags*][*RIPSO/CIPSO options*] *security_label_information*

netrule eq

netrule e { **on** | **off** }

Description

The **netrule** command lists, queries, adds and removes rule specifications for interfaces and hosts. The system default interface rules are set using the interface name. When an interface is removed using the **i-** flag, it will be given these default interface rules. The default interface rules are also set using the **tninit load** command.

Note: Because there must always be an interface rule for an interface, the remove operation sets the interface rule to its default state. All of the command line flags must follow the order as shown in the syntax statements.

Flags

Item	Description
e { on off }	Sets the policy for sending the ICMP error response to incoming packets that are not accepted by the system. This setting is off by default and must be set with this flag to be on. You cannot specify the e flag when you specify the h or i flag.
h	Specifies that the object of the netrule command is a host. You cannot specify the h flag when you specify the i or e flag.
i	Specifies that the object of the netrule command is an interface. You cannot specify the i flag when you specify the h or e flag.
l	Lists all rules for interfaces or hosts.
o	Specifies the host out rules (for host rule only).
q	Queries an interface, a host rule, or the status of the error response setting.
u	Specifies that the /etc/security/rules.host and /etc/security/rules.int files will be updated after the host or interface rule is successfully added or removed.
+	Adds an interface or a host rule.

Item	Description
-	Removes an interface or a host rule.
<i>interface</i>	Specifies an interface name.
<i>src_host_rule_specification</i>	<p>This parameter takes the following format:</p> <pre style="background-color: #f0f0f0; padding: 5px;"><code>src_host [/ mask][= proto [:start_port_range [:end_port_range]]]</code></pre> <p>Requirement: There is a space or tab in between each field.</p> <p>src_host A source IPv6 address, or an IPv4 address, or a host name.</p> <p>mask The subnet mask number indicates how many bits are set, starting from the most significant bit. For example, 24 means 255.255.255.0 for an IPv4 address.</p> <p>proto A protocol.</p> <p>start_port_range A particular port number or name to begin from.</p> <p>end_port_range A particular port number or name to end at.</p>
<i>dst_host_rule_specification</i>	<p>This parameter takes the following format:</p> <pre style="background-color: #f0f0f0; padding: 5px;"><code>dst_host [/ mask][= proto [:start_port_range [:end_port_range]]]</code></pre> <p>Requirement: There is a space or tab in between each field.</p> <p>dst_host A destination IPv6 address, or an IP v4 address, or a host name.</p> <p>mask The subnet mask number, which indicates how many bits are set, starting from the most significant bit. For example, 24 means 255.255.255.0 for an IPv4 address.</p> <p>proto A protocol.</p> <p>start_port_range A particular port number or name to begin in range from.</p> <p>end_port_range A particular port number or name to end at.</p>

Item*flags***Description**

This parameter takes the following format:

```
-d drop
```

drop

AIX Trusted Network can be configured to drop all packets. You can specify one of the following values:

r

Drops all packets

n

Does not drop all packets (interface default).

i

Uses interface default (host default, host only).

```
-f rflag:tflag
```

rflags

Security option requirement on incoming (received) packets. You can specify one of the following values:

r

Revised Interconnection Protocol Security Option (RIPSO) only.

c

Commercial Internet Protocol Security Option (CIPSO) only.

e

Either RIPSO or CIPSO.

n

Neither RIPSO or CIPSO (system default).

a

No restrictions.

i

Uses interface or system default (default).

tflag

Security option handling on outgoing (transmitted) packets. You can specify one of the following values:

r

Transmits RIPSO.

c

Transmits CIPSO.

n

Does not transmit any security options (interface default).

i

Uses interface default (host default, host only).

Item	Description
<i>RIPSO/CIPSO options</i>	<p>This parameter takes the following format:</p> <p>-rpafs=<i>PAF_field</i>[,<i>PAF_field</i>...] Specifies the PAF fields that are used to receive IPSO packets. This is a list of PAF fields that are accepted. There can be up to 256 fields.</p> <p><i>PAF_field</i>: NONE <i>PAF</i> [+<i>PAF</i>...] Specifies PAF fields, which are collections of PAFs. The following are the five PAFs that can be included in a single PAF field:</p> <ul style="list-style-type: none"> • GENSER • SIOP-ESI • SCI • NSA • DOE <p>A PAF field is a combination of these values separated by a plus sign (+). For example, a PAF field containing both GENSER and SCI is represented as GENSER+SCI. You can use the PAF field NONE to specify the PAF field without any specified PAFs.</p> <p>-epaf=<i>PAF_field</i> Specifies the PAF field that is attached to error responses for incoming IPSO packets that were not accepted by the system.</p> <p>-tpaf=<i>PAF_field</i> Specifies the PAF field that is included in the IPSO options of outgoing packets.</p> <p>-DOI = <i>doi</i> Specifies the domain of interpretation (DOI) for CIPSO packets. Incoming packets must have this DOI and outgoing packets will be given this DOI.</p> <p>-tags=<i>tag</i>[,<i>tag</i>...]</p> <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;"> <i>tag</i> = 1 2 5 </div> <p>Specifies the set of tags that are accepted and available to be transmitted by CIPSO options. This is a combination of 1, 2 and 5. For example 1,2 would enable tags 1 and 2.</p>
<i>security_label_information</i>	<p>This parameter takes the following format:</p> <p>+min +max +default -s <i>input_file</i> Specifies the standard output (SL) that will apply when adding a rule. You can also specify the -s flag and include the SLs in the file in the following order, specifying one per line:</p> <ul style="list-style-type: none"> • min SL • max SL • default SL <p>You cannot include any comments in the file. Use a backslash (\) at the end of the line if more than one line is needed. If you are not using a file, list the sensitivity labels delimited by a plus sign (+) for the minimum level, the maximum level, and the default or implicit level for unmarked packets.</p>

Security

A user must have the **aix.mls.network.config** and the **aix.mls.network.init** authorizations to run the **netrule** command.

Examples

1. To add in host rule, and update the local database after in host rule is successfully added to kernel, enter:

```
netrule h+iu 9.3.149.25 9.41.86.19 +impl_lo +ts all +pub
```

2. To add out host rule, enter:

```
netrule h+o 9.41.86.19 9.3.149.25 -s /tmp/rule
```

or:

```
impl_lo
ts all
pub
```

The following are the contents of the input **/tmp/rule** file:

```
impl_lo
ts \
all
pub
```

3. To drop all incoming UDP packets from a host, enter:

```
netrule h+i 192.0.0.5 =udp 9.41.86.19 =udp -dr +impl_lo +impl_lo +impl_lo
```

4. To remove all host rules and update the local, enter:

```
netrule h-u
```

5. To list all host rules, enter:

```
netrule hl
```

6. To list all interface rules, enter:

```
netrule il
```

7. To add an interface rule, enter:

```
netrule i+ en0 -dn -fa:n +public +ts +secret
```

8. To remove a particular host rule, enter:

```
netrule h-i 192.0.0.5 =udp 9.41.86.19 =udp
```

9. To add a particular host rule, enter:

```
netrule h+i 9.41.86.19 /24 =tcp :ftp :telnet 9.3.149.6 /28 +public +ts +secret
```

10. To set the default interface rule, enter:

```
netrule i+ default -dn -fa:n +impl_lo +ts all +impl_lo
```

11. To set the default interface rule to the system drop-all-packets default, enter:

```
netrule i- default
```

12. To set the interface to send and only receive CIPSO packets, enter:

```
netrule i+ en0 -fc:c +impl_lo +ts all +impl_lo
```

13. To set the interface to receive either CIPSO or RIPSO packets and send RIPSO packets with PAF values, a CIPSO DOI, and CIPSO flags, enter:

```
netrule i+ en0 -fe:r -rpafs=SCI,NSA+DOE -epaf=SCI -tpaf=NSA -DOI=0x010  
-tags=1,2 +impl_lo +ts all +impl_lo
```

14. To set the system-wide policy for sending ICMP responses on incoming packets that are not valid, enter:

```
netrule e on
```

netstat Command

Purpose

Shows network status.

Syntax

To display active sockets for each protocol or routing table information

```
/bin/netstat [ -n ] [ { -A -a } | { -r -C -i -I Interface } ] [ -f AddressFamily ] [ [ -p Protocol ] | [ -@ WparName ] ] [ Interval ]
```

To display the contents of a network data structure

```
/bin/netstat [ -m | -M | -s | -ss | -u | -v ] [ -f AddressFamily ] [ [ -p Protocol ] | [ -@ WparName ] ] [ Interval ]
```

To display the virtual interface table and multicast forwarding cache

```
/bin/netstat -g
```

To display the packet counts throughout the communications subsystem

```
/bin/netstat -D
```

To display the network buffer cache statistics

```
/bin/netstat -c
```

To display the data link provider interface statistics

```
/bin/netstat -P
```

To clear the associated statistics

```
/bin/netstat [ -Zc | -Zi | -Zm | -Zs ]
```

> | To display artifacts for a specific protocol | <

```
/bin/netstat -K protocol [ -F filename ] [ -w ] [ -b ]
```

Description

The **netstat** command symbolically displays the contents of various network-related data structures for active connections. The *Interval* parameter, which is specified in seconds, continuously displays information regarding packet traffic on the configured network interfaces. The *Interval* parameter takes no flags.

Flags

Item	Description
-A	Shows the address of any protocol control blocks associated with the sockets. This flag acts with the default display and is used for debugging purposes.
-a	Shows the state of all sockets. If this flag is not specified, sockets that are used by server processes that are not bound to an interface are not shown.
-b	Suppresses the printing of non-zero counter values when you use the -K flag to display the RAS artifacts for a specific protocol.
-c	Shows the statistics of the Network Buffer Cache.

The Network Buffer Cache is a list of network buffers that contain data objects that can be transmitted to networks. The Network Buffer Cache grows dynamically as data objects are added to or removed from it. The Network Buffer Cache is used by some network kernel interfaces for performance enhancement on the network I/O. The **netstat -c** command prints the following statistic:

```
Network Buffer Cache Statistics:
Current total cache buffer size: 0
Maximum total cache buffer size: 0
Current total cache data size: 0
Maximum total cache data size: 0
Current number of cache: 0
Maximum number of cache: 0
Number of cache with data: 0
Number of searches in cache: 0
Number of cache hit: 0
Number of cache miss: 0
Number of cache newly added: 0
Number of cache updated: 0
Number of cache removed: 0
Number of successful cache accesses: 0
Number of unsuccessful cache accesses: 0
Number of cache validation: 0
Current total cache data size in private segments: 0
Maximum total cache data size in private segments: 0
Current total number of private segments: 0
Maximum total number of private segments: 0
Current number of free private segments: 0
Current total NBC_NAMED_FILE entries: 0
Maximum total NBC_NAMED_FILE entries: 0
```


Item	Description
-C	<p>Shows the routing tables, including the user-configured and current costs of each route. The user-configured cost is set by using the -hopcount flag of the route command. The current cost can be different than the user-configured cost if Dead Gateway Detection has changed the cost of the route.</p> <p>In addition to the costs of the route, it also shows the weight and policy information associated with each route. These fields are applicable only when the Multipath Routing Feature is used. The policy information displays the routing policy that has been currently selected to choose between the multiple routes available. The policies available are:</p> <ul style="list-style-type: none"> • Default - Weighted Round Robin (WRR) • Hashed (HSH) • Random (RND) • Weighted Random (WRND) • Lowest Utilization (LUT) <p>If multiple routes are present for the same destination (multipath routes), one of these routes display the policy value of WRR, HSH, RND, WRND, or LUT. All the other routes in this set display the policy information as - " - . This means that all the routes in this set are using the same routing policy displayed by the first route.</p> <p>The weight field is a user-configured weight associated with the route that will be used for Weighted Round-Robin and Weighted Random Policies. For more information about these policies, see the no command.</p>
-D	<p>Shows the number of packets received, transmitted, and dropped in the communications subsystem.</p> <p>Note: In the statistics output, a N/A displayed in a field value indicates the count is not applicable. For the NFS/RPC statistics, the number of incoming packets that pass through RPC are the same packets that pass through NFS, so these numbers are not summed in the NFS/RPC <code>Total</code> field, thus the N/A. NFS has no outgoing packet or outgoing packet drop counters specific to NFS and RPC. Therefore, individual counts have a field value of N/A, and the cumulative count is stored in the NFS/RPC <code>Total</code> field.</p>
-F filename	<p>Writes all the reliability, availability, and serviceability (RAS) artifacts for the specified protocol to the specified file. You can specify this file only with the -K flag.</p>
-f AddressFamily	<p>Limits reports of statistics or address control blocks to those items specified by the <i>AddressFamily</i> variable. The following address families are recognized:</p> <p>inet Indicates the AF_INET address family.</p> <p>inet6 Indicates the AF_INET6 address family.</p> <p>unix Indicates the AF_UNIX address family.</p>

Item	Description
-g	Shows Virtual Interface Table and Multicast Forwarding Cache information. If used in conjunction with the -s flag, it will show the multicast routing information.
-i	Shows the state of all configured interfaces. See Interface Display Note: The collision count for Ethernet interfaces is not supported.
-I <i>Interface</i>	Shows the state of the configured interface specified by the <i>Interface</i> variable.
> > -K <i>protocol</i>	Displays all reliability, availability, and serviceability (RAS) artifacts for the specified protocol. If you specify the -F filename with the -K flag, all the artifact information for the specified protocol is written to a file in the specified path and with the specified file name. Note: The supported protocols for this flag are tcp, udp, unix, and ip.
	 <<<
-M	Shows network memory's mbuf cluster pool statistics.
-m	Shows statistics recorded by the memory management routines.
-n	Shows network addresses as numbers. When this flag is not specified, the netstat command interprets addresses where possible and displays them symbolically. This flag can be used with any of the display formats.
-o	Used in conjunction with the -a flag to display detailed data about a socket, such as socket options, flags, and buffer statistics.
-p <i>Protocol</i>	Shows statistics about the value specified for the <i>Protocol</i> variable, which is either a well-known name for a protocol or an alias for it. Some protocol names and aliases are listed in the /etc/networks file.
-P	Shows the statistics of the Data Link Provider Interface (DLPI). The netstat -P command prints the following statistic: <pre>DLPI statistics: Number of received packets = 0 Number of transmitted packets = 0 Number of received bytes = 0 Number of transmitted bytes = 0 Number of incoming pkts discard = 0 Number of outgoing pkts discard = 0 Number of times no buffers = 0 Number of successful binds = 0 Number of unknown message types = 0 Status of phys level promisc = 0 Status of sap level promisc = 0 Status of multi level promisc = 0 Number of enab_multi addresses = 0</pre>
	If DLPI is not loaded, it displays: <pre>can't find symbol: dl_stats</pre>
-r	Shows the routing tables. When used with the -s flag, the -r flag shows routing statistics. See Routing Table Display.
-s	Shows statistics for each protocol.
-ss	Displays all the non-zero protocol statistics and provides a concise display.
-u	Displays information about domain sockets.

Item	Description
-v	Shows statistics for CDLI-based communications adapters. This flag causes the netstat command to run the statistics commands for the netstat , tokstat , and fdistat commands. No flags are issued to these device driver commands. See the specific device driver statistics command to obtain descriptions of the statistical output.
-w	Starts the user interactive mode.
-Zc	Clear network buffer cache statistics.
-Zi	Clear interface statistics.
-Zm	Clear network memory allocator statistics.
-Zs	Clear protocol statistics. To clear statistics for a specific protocol, use -p protocol . For example, to clear TCP statistics, type netstat -Zs -p tcp .
-@ WparName	Displays the network statistics associated with workload partition (WparName). If no WparName is specified, then show the network statistics for all workload partitions.

Notes:

1. The **-C**, **-D**, **-c**, **-g**, **-m**, **-M**, **-P**, **-r**, **-v**, and **-Z** flags are not supported in the global environment when used in conjunction with the **-@ WparName** option.
2. The **-C**, **-D**, **-c**, **-g**, **-m**, **-M**, **-P**, **-r**, **-v**, and **-Z** flags are not supported in system workload partitions.

Default Display

The default display for active sockets shows the following items:

- Local and remote addresses
- Send and receive queue sizes (in bytes)
- Protocol
- Internal state of the protocol

Internet address formats are of the form `host . port` or `network . port` if a socket's address specifies a network but no specific host address. The host address is displayed symbolically if the address can be resolved to a symbolic host name, while network addresses are displayed symbolically according to the `/etc/networks` file.

If a symbolic name for a host is not known or if the **-n** flag is used, the address is printed numerically, according to the address family. Unspecified addresses and ports appear as an * (asterisk).

Interface Display (**netstat -i**)

The interface display format provides a table of cumulative statistics for the following items:

- Errors
- Collisions

Note: The collision count for Ethernet interfaces is not supported.

- Packets transferred

The interface display also provides the interface name, number, and address as well as the maximum transmission units (MTUs).

Routing Table Display (**netstat -r**)

The routing table display indicates the available routes and their statuses. Each route consists of a destination host or network and a gateway to use in forwarding packets.

A route is given in the format `A.B.C.D/XX`, which presents two pieces of information. `A.B.C.D` indicates the destination address and `XX` indicates the netmask associated with the route. The netmask is

represented by the number of bits set. For example, the route 9.3.252.192/26 has a netmask of 255.255.255.192, which has 26 bits set.

The routing table contains the following fields:

Item	Description
WPAR	Displays the name of the workload partition to which this route belongs. This field is only present when the -@ flag is used with the -r flag. For routes belonging to the global system, Global is displayed in this column.

Item	Description
Flags	<p>The flags field of the routing table shows the state of the route:</p> <p>A An Active Dead Gateway Detection is enabled on the route.</p> <p>U Up.</p> <p>H The route is to a host rather than to a network.</p> <p>G The route is to a gateway.</p> <p>D The route was created dynamically by a redirect.</p> <p>M The route has been modified by a redirect.</p> <p>L The link-level address is present in the route entry.</p> <p>c Access to this route creates a cloned route.</p> <p>W The route is a cloned route.</p> <p>1 Protocol specific routing flag #1.</p> <p>2 Protocol specific routing flag #2.</p> <p>3 Protocol specific routing flag #3.</p> <p>b The route represents a broadcast address.</p> <p>e Has a binding cache entry.</p> <p>l The route represents a local address.</p> <p>m The route represents a multicast address.</p> <p>P Pinned route.</p> <p>R Host or net unreachable.</p> <p>S Manually added.</p> <p>u Route usable.</p> <p>s The Group Routing stopsearch option is enabled on the route.</p> <p>Direct routes are created for each interface attached to the local host.</p>
Gateway	<p>The gateway field for these entries shows the address of the outgoing interface.</p>

Item	Description
Refs	Gives the current number of active uses for the route. Connection-oriented protocols hold on to a single route for the duration of a connection, while connectionless protocols obtain a route while sending to the same destination.
Use	Provides a count of the number of packets sent using that route.
PMTU	Gives the Path Maximum Transfer Unit (PMTU). AIX 5.3 does not display the PMTU column.
Interface	Indicates the network interfaces utilized for the route.
Exp	Displays the time (in minutes) remaining before the route expires.
Groups	Provides a list of group IDs associated with that route.
Netmasks	Lists the netmasks applied on the system.
Route Tree for Protocol Family	Specifies the active address families for existing routes. Supported values for this field are: <ul style="list-style-type: none"> 1 Specifies the UNIX address family. 2 Specifies the Internet address family (for example, TCP and UDP). For more information on other address families, refer to the /usr/include/sys/socket.h file.

When the **-@** flag is used with the **netstat -r** command and no *WparName* parameter is specified, all of the routes in the system's route table are displayed. If the *WparName* parameter is specified and the WPAR-specific routing is enabled for that WPAR, only the routes associated with that WPAR are displayed. If the *WparName* parameter is specified and the WPAR specific routing is disabled for that WPAR, the routes associated with the global system are displayed.

When a value is specified for the *Interval* parameter, the **netstat** command displays a running count of statistics related to network interfaces. This display contains two columns: a column for the primary interface (the first interface found during autoconfiguration) and a column summarizing information for all interfaces.

The primary interface may be replaced with another interface by using the **-I** flag. The first line of each screen of information contains a summary of statistics accumulated since the system was last restarted. The subsequent lines of output show values accumulated over intervals of the specified length.

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 display routing table information for an Internet interface, enter the following command:

```
netstat -r -f inet
```

This produces the following output:

```
Routing tables
Destination Gateway      Flags Refs Use  PMTU If  Exp Groups Netmasks:
(root node)
```

```

(0)0 ffff f000 0
(0)0 ffff f000 0
(0)0 8123 262f 0 0 0 0
(root node)

Route Tree for Protocol Family 2:
(root node)
default      129.35.38.47   UG   0 564 -   tr0 -
loopback     127.0.0.1        UH   1 202 -   lo0 -
129.35.32    129.35.41.172   U    4 30  -   tr0 -   +staff
129.35.32.117 129.35.41.172  UGHW 0 13 1492 tr0 30
192.100.61   192.100.61.11  U    1 195 -   en0 -
(root node)

Route Tree for Protocol Family 6:
(root node)
(root node)

```

The `-r -f inet` flags indicate a request for routing table information for all configured Internet interfaces. The network interfaces are listed in the `Interface` column; `en` designates a Standard Ethernet interface, while `tr` specifies a Token-Ring interface. Gateway addresses are in dotted decimal format.

Note: AIX 5.3 does not display the PMTU column.

- To display statistics for GRE Protocol, enter the following command:

```
netstat -s -p gre
```

This produces the following output:

```

GRE Interface gre0
  10 number of times gre_input got called
   8 number of times gre_output got called
  0 packets received with protocol not supported
  0 packets received with checksum on
  0 packets received with routing present
  0 packets received with key present
  0 packets received with sequence number present
  0 packets received with strict source route present
  0 packets received with recursion control present
  0 packets received where reserved0 non-zero
  0 packets received where version non-zero
  0 packets discarded
  0 packets dropped due to network down
  0 packets dropped due to protocol not supported
  0 packets dropped due to error in ip output routine
  0 packets got by NAT
  0 packets got by NAT but not TCP packet
  0 packets got by NAT but with IP options

```

- To display statistics for the IPv4 over IPv6 tunnel (GIF tunnel), enter the following command:

```
netstat -s -p gif
```

The command produces the following output:

```

GIF Interface gif0
44 total packets received
50 total packets sent
 0 packets received with protocol not supported
 0 packets received with checksum on
 0 packets received with routing present
 0 packets received with strict source route present
 0 packets received where version non-zero
 0 packets discarded
 0 packets dropped due to network down
 0 packets dropped due to protocol not supported
 0 packets dropped due to error in ipv6 output routine

```

- [>](#) To display statistics for the Virtual Extensible LAN (Vxlan) protocol, enter the following command:

```
netstat -s -p vxl
```

This command produces the following output:

```
VXLAN Interface vxl0
0 number of times vxl_input got called
0 number of times vxl_output got called
0 packets received with protocol not supported
0 packets received with checksum on
0 packets received with routing present
0 packets received with key present
0 packets received with sequence number present
0 packets received where version non-zero
0 packets discarded
0 packets dropped due to lack of memory
0 internal socket errors
0 packets dropped due to protocol not supported
0 packets dropped due to network down
0 packets received with size > MTU
0 packets received over layer 3 tunnel
0 control packets sent
0 control packets received
0 packets with flags discarded
```

⏪

5. To display statistics for the Generic Network Virtualization Encapsulation (Geneve) protocol, enter the following command:

```
netstat -s -p gne
```

This command produces the following output:

```
GENEVE Interface gne0
0 number of times gne_input got called
0 number of times gne_output got called
0 packets received with protocol not supported
0 packets received with checksum on
0 packets received with routing present
0 packets received with key present
0 packets received with sequence number present
0 packets received where version non-zero
0 packets discarded
0 packets dropped due to lack of memory
0 internal socket errors
0 packets dropped due to protocol not supported
0 packets dropped due to network down
0 packets received with size > MTU
0 packets received over layer 2 tunnel
0 packets received over layer 3 IPv4 tunnel
0 packets received over layer 3 IPv6 tunnel
0 control packets sent
0 control packets received
0 packets with options discarded
```

⏪

6. To display interface information for an Internet interface, enter the following command:

```
netstat -i -f inet
```

This produces the following output:

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
lo0	16896	Link#1		5161	0	5193	0	0
lo0	16896	127	localhost	5161	0	5193	0	0
lo0	16896	::1		5161	0	5193	0	0
en1	1500	Link#2	8.0.38.22.8.34	221240	0	100284	0	0
en1	1500	129.183.64	infoserv.frec.bul	221240	0	100284	0	0

The `-i -f inet` flags indicate a request for the status of all configured Internet interfaces. The network interfaces are listed in the Name column; `lo` designates a loopback interface, `en` designates a Standard Ethernet interface, while `tr` specifies a Token-Ring interface.

7. To display statistics for each protocol, enter the following command:


```
netstat -s -f inet
```

This produces the following output:

```
ip:
:
 44485 total packets received
 0 bad header checksums
 0 with size smaller than minimum
 0 with data size < data length
 0 with header length < data size
 0 with data length < header length
 0 with bad options
 0 with incorrect version number
 0 fragments received
 0 fragments dropped (dup or out of space)
 0 fragments dropped after timeout
 0 packets reassembled ok
44485 packets for this host
 0 packets for unknown/unsupported protocol
 0 packets forwarded
 0 packets not forwardable
 0 redirects sent
1506 packets sent from this host
 0 packets sent with fabricated ip header
 0 output packets dropped due to no bufs, etc.
 0 output packets discarded due to no route
 0 output datagrams fragmented
 0 fragments created
 0 datagrams that can't be fragmented
 0 IP Multicast packets dropped due to no receiver
 0 successful path MTU discovery cycles
 0 path MTU rediscovery cycles attempted
 0 path MTU discovery no-response estimates
 0 path MTU discovery response timeouts
 0 path MTU discovery decreases detected
 0 path MTU discovery packets sent
 0 path MTU discovery memory allocation failures
 0 ipintrq overflows

icmp:
 0 calls to icmp_error
 0 errors not generated 'cuz old message was icmp
Output histogram:
  echo reply: 6
 0 messages with bad code fields
 0 messages < minimum length
 0 bad checksums
 0 messages with bad length
Input histogram:
  echo: 19
 6 message responses generated

igmp:defect
 0 messages received
 0 messages received with too few bytes
 0 messages received with bad checksum
 0 membership queries received
 0 membership queries received with invalid field(s)
 0 membership reports received
 0 membership reports received with invalid field(s)
 0 membership reports received for groups to which we belong
 0 membership reports sent

tcp:
1393 packets sent
 857 data packets (135315 bytes)
 0 data packets (0 bytes) retransmitted
 367 URG only packets
 0 URG only packets
 0 window probe packets
 0 window update packets
170 control packets
1580 packets received
 790 acks (for 135491 bytes)
 60 duplicate acks
 0 acks for unsent data
 638 packets (2064 bytes) received in-sequence
 0 completely duplicate packets (0 bytes)
 0 packets with some dup. data (0 bytes duped)
```

```

117 out-of-order packets (0 bytes)
0 packets (0 bytes) of data after window
0 window probes
60 window update packets
0 packets received after close
0 discarded for bad checksums
0 discarded for bad header offset fields
0 connection request
58 connection requests
61 connection accepts
118 connections established (including accepts)
121 connections closed (including 0 drops)
0 embryonic connections dropped
845 segments updated rtt (of 847 attempts)
0 resends due to path MTU discovery
0 path MTU discovery terminations due to retransmits
0 retransmit timeouts
0 connections dropped by rexmit timeout
0 persist timeouts
0 keepalive timeouts
0 keepalive probes sent
0 connections dropped by keepalive

udp:
42886 datagrams received
:
0 incomplete headers
0 bad data length fields
0 bad checksums
0 dropped due to no socket
42860 broadcast/multicast datagrams dropped due to no

socket
0 socket buffer overflows
26 delivered
106 datagrams output

```

ip specifies the Internet Protocol; icmp specifies the Information Control Message Protocol; tcp specifies the Transmission Control Protocol; udp specifies the User Datagram Protocol.

Note: AIX 5.3 does not display the PMTU statistics for the IP protocol.

- To display device driver statistics, enter the following command:

```
netstat -v
```

The `netstat -v` command displays the statistics for each CDLI-based device driver that is up. To see sample output for this command, see the **tokstat** command, the **entstat** command, or the **fd distat** command.

- To display information regarding an interface for which multicast is enabled, and to see group membership, enter the following command:

```
netstat -a -I interface
```

For example, if an 802.3 interface was specified, the following output will be produced:

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
et0	1492	<Link>		0	0	2	0	0
et0	1492	9.4.37	hun-eth 224.0.0.1 02:60:8c:0a:02:e7 01:00:5e:00:00:01	0	0	2	0	0

If instead of **-I interface** the flag **-i** is given, then all configured interfaces will be listed. The network interfaces are listed in the Name column; **lo** designates a loopback interface, **et** designates an IEEE 802.3 interface, **tr** designates a Token-Ring interface, while **fi** specifies an FDDI interface.

The address column has the following meaning. A symbolic name for each interface is shown. Below this symbolic name, the group addresses of any multicast groups that have been joined on that interface are shown. Group address 224.0.0.1 is the special *all-hosts-group* to which all multicast interfaces belong. The MAC address of the interface (in colon notation) follows the group addresses,

plus a list of any other MAC level addresses that are enabled on behalf of IP Multicast for the particular interface.

- To display the packet counts in the communication subsystem, enter the following command:

```
netstat -D
```

The following output will be produced:

Source	Ipkts	Opkts	Idrops	Odrops
tok_dev0	720	542	0	0
ent_dev0	114	4	0	0
Devices Total	834	546	0	0
tok_dd0	720	542	0	0
ent_dd0	114	4	0	0
Drivers Total	834	546	0	0
tok_dmx0	720	N/A	0	N/A
ent_dmx0	114	N/A	0	N/A
Demuxer Total	834	N/A	0	N/A
IP	773	767	0	0
TCP	536	399	0	0
UDP	229	93	0	0
Protocols Total	1538	1259	0	0
lo_if0	69	69	0	0
en_if0	22	8	0	0
tr_if0	704	543	0	1
Net IF Total	795	620	0	1
NFS/RPC Client	519	N/A	0	N/A
NFS/RPC Server	0	N/A	0	N/A
NFS Client	519	N/A	0	N/A
NFS Server	0	N/A	0	N/A
NFS/RPC Total	N/A	519	0	0

(Note: N/A -> Not Applicable)

- To display detailed data of active sockets, enter the following command:

```
netstat -aon
```

Output similar to the following is displayed:

```
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        (state)
tcp4    0      0 *.13                  *.*                    LISTEN
      so_options: (ACCEPTCONN|REUSEADDR)
      q0len:0 qlen:0 qlimit:1000      so_state: (PRIV)
      timeo:0 uid:0
      so_special: (LOCKBALE|MEMCOMPRESS|DISABLE)
      so_special2: (PROC)
      sndbuf:
        hiwat:16384 lowat:4096 mbcnt:0 mbmax:65536
      rcvbuf:
        hiwat:16384 lowat:1 mbcnt:0 mbmax:65536
      sb_flags: (SEL)
      TCP:
        mss:512
tcp     0      0 *.21                  *.*                    LISTEN
      so_options: (ACCEPTCONN|REUSEADDR)
      q0len:0 qlen:0 qlimit:1000      so_state: (PRIV)
      timeo:0 uid:0
      so_special: (LOCKBALE|MEMCOMPRESS|DISABLE)
      so_special2: (PROC)
      sndbuf:
```

```

        hiwat:16384 lowat:4096 mbcnt:0 mbmax:65536
rcvbuf:
        hiwat:16384 lowat:1 mbcnt:0 mbmax:65536
        sb_flags: (SEL)
TCP:
mss:512

.....
.....

```

12. To display the routing table, enter the following command:

```
netstat -rn
```

Output similar to the following is displayed:

```

Routing tables
Destination      Gateway          Flags    Refs      Use  If    PMTU  Exp  Groups

Route Tree for Protocol Family 2 (Internet):
default          9.3.149.65      UG       0         24  en0   -    -
9.3.149.64       9.3.149.88      UHSb    0         0   en0   -    - =>
9.3.149.64/27    9.3.149.88      U        1         0   en0   -    -
9.3.149.88       127.0.0.1       UGHS    0         1   lo0   -    -
9.3.149.95       9.3.149.88      UHSb    0         0   en0   -    -
127/8            127.0.0.1       U       11        174  lo0   -    -

Route Tree for Protocol Family 24 (Internet v6):
::1              ::1              UH       0         0   lo0   -    -

```

Note: AIX 5.3 does not display the PMTU column.

The character => at the end of the line means the line is a duplicate route of the route on the next line.

The loopback route (9.3.149.88, 127.0.0.1) and the broadcast routes (with the flags field containing b indicating broadcast) are automatically created when an interface is configured. Two broadcast routes are added: one to the subnet address and one to the broadcast address of the subnet. The presence of the loopback routes and broadcast routes improve performance.

13. To display the routing table of a workload partition named wpar1, enter the following command:

```
netstat -rn@ wpar1
```

Output similar to the following is displayed:

```

Routing tables
WPAR Destination      Gateway          Flags    Refs      Use  If    Exp  Groups

Route Tree for Protocol Family 2 (Internet):
wpar1 default          9.4.150.1       UG       1        13936  en1   -    -
wpar1 9.4.150.0       9.4.150.57      UHSb    0         0   en1   -    - =>
wpar1 9.4.150/24     9.4.150.57      U        0         0   en0   -    -
wpar1 9.4.150.57     127.0.0.1       UGHS    0         0   lo0   -    -
wpar1 9.4.150.255    9.4.150.57      UHSb    0         3   en0   -    -

```

14. >| To display the RAS artifacts for the TCP sockets on an AIX logical partition, enter the following command:

```
netstat -K tcp -w
```

The following interactive menu is displayed where you can select a socket artifact:

```

-----
remote      num      local      local      remote
state      alerts  address    port        address
port
-----
12  closed    0      *          smux        *          *
13  listen    0      *          filenet-nch *          *
14  closed    0      *          filenet-rmi *          *
15  listen    0      *          filenet-pa  *          *
16  listen    0      *          filenet-re  *          *
17  listen    0      *          32775       *          *
18  closed    2      192.168.120.2 *          9.3.80.16  nim
19  closed    0      *          writesrv    *          *
20  closed    0      *          clcomd_caa  *          *
21  closed    0      *          nut          *          *
22  closed    0      *          tvdumtray-port *        *
23  closed    0      *          ssh         *          *
24  closed    0      *          ssh         *          *
25  closed    0      *          rmc         *          *
26  closed    2      192.168.120.2 *          9.3.80.16 nim

```

Press <ret> to display socket details.
Press <esc> to exit.

When you select a socket artifact and press Enter, the **netstat** command displays detailed information and any associated alert messages about the socket artifact as shown in the following example:

CLOSED TCP SOCKET

```

Socket Open Time      02/08/22 12:52:16
Socket Close Time    02/08/22 12:53:31
Lifetime              75 sec
Local Address         192.168.120.2
Local Port            0 (*)
Remote Address        9.3.80.16
Remote Port           1058 (nim)
Protocol State        closed
Send Packets          0
Send Bytes            0
Receive Packets       0
Receive Bytes         0
Retransmit Timeouts   4
Keep Alive Timeouts   1
Keep Alive Drops      1
Send Control          5

tcp_timer.c(889) [02/08/22 12:53:01] : TCP Largesend disabled
tcp_subr.c(1023) [02/08/22 12:53:31] : TCP Dropped (errno:78)

```

Press <F> to display flag information.
Press <esc> to exit.

If you press the F key in this menu, the **netstat** command displays the flag details for the socket artifact as shown in the following example:

FLAGS

```

Protocol can be used by IPv4
Protocol can be used by any
Press <esc> to exit.

```



newaliases Command

Purpose

Builds a new copy of the alias database from the mail aliases file.

Syntax

newaliases

Description

The **newaliases** command builds a new copy of the alias database from the **/etc/aliases** file. It must be run each time this file is changed in order for the changes to take effect. Running this command is equivalent to running the **sendmail** command with the **-bi** flag.

Exit Status

Item	Description
0	Exits 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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Files

Item	Description
/usr/sbin/newaliases	Contains the newaliases command.
/etc/mail/html	

newform Command

Purpose

Changes the format of a text file.

Syntax

newform [**-s**] [**-f**] [**-a** [*Number*]] [**-b** [*Number*]] [**-c** [*Character*]] [**-e** [*Number*]] [**-i** [*TabSpec*]] [**-l** [*Number*]] [**-o** [*TabSpec*]] [**-p** [*Number*]] [*File ...*]

Description

The **newform** command takes lines from the files specified by the *File* parameter (standard input by default) and writes the formatted lines to standard output. Lines are reformatted in accordance with the command-line flags in effect.

Except for the **-s** flag, you can enter command-line flags in any order, repeated, and mixed with the *File* parameter. However, the system processes command-line flags in the order you specify. For example, the

-c flag modifies the behavior of the **-a** and **-p** flags, so specify the **-c** flag before the **-p** or **-a** flag for which it is intended. The **-l** (lowercase L) flag modifies the behavior of the **-a**, **-b**, **-e**, and **-p** flags, so specify the **-l** flag before the flags for which it is intended. For example, flag sequences like **-e15 -l60** yield results that are different from **-l60 -e15**. Flags are applied to all files specified on the command line.

An exit value of 0 indicates normal execution; an exit value of 1 indicates an error.

Note:

1. The **newform** command normally only keeps track of physical characters; however, for the **-i** and **-o** flags, the **newform** command keeps track of backspaces to line up tabs in the appropriate logical columns.
2. The **newform** command does not prompt you if the system reads a *TabSpec* variable value from standard input (by use of the **-i-** or **-o-** flag).
3. If you specify the **-f** flag, and the last **-o** flag you specified was **-o-** preceded by either an **-o-** or an **-i-**, the tab-specification format line is incorrect.
4. If the values specified for the **-p**, **-l**, **-e**, **-a**, or **-b** flag are not valid decimal numbers greater than 1, the specified value is ignored and default action is taken.

Flags

Item	Description
-a [<i>Number</i>]	Adds the specified number of characters to the end of the line when the line length is less than the effective line length. If no number is specified, the -a flag defaults to 0 and adds the number of characters necessary to obtain the effective line length. See also the -c [<i>Character</i>] and -p [<i>Number</i>] flags.
-b [<i>Number</i>]	Truncates the specified number of characters from the beginning of the line if the line length is greater than the effective line length. If the line also contains fewer characters than specified by the <i>Number</i> parameter, the entire line is deleted and a blank line is displayed in its place. See also the -I [<i>Number</i>] flag. If you specify the -b flag with no <i>Number</i> variable, the default action truncates the number of characters necessary to obtain the effective line length. This flag can be used to delete the sequence numbers from a COBOL program, as follows: <pre>newform -l1-b7 file-name</pre> The -l1 flag must be used to set the effective line length shorter than any existing line in the file so that the -b flag is activated.
-c [<i>Character</i>]	Changes the prefix/add character to that specified by the <i>Character</i> variable. Default character is a space and is available when specified before the -a and -p flags.
-e [<i>Number</i>]	Truncates the specified number of characters from the end of the line. Otherwise, the flag is the same as the -b [<i>Number</i>] flag.
-f	Writes the tab-specification format line to standard output before any other lines are written. The displayed tab-specification format line corresponds to the format specified by the final -o flag. If no -o flag is specified, the line displayed contains the default specification of -8.

Item	Description
-i [<i>TabSpec</i>]	<p>Replaces all tabs in the input with the number of spaces specified by the <i>TabSpec</i> variable.</p> <p>This variable recognizes all tab specification forms described in the tabs command.</p> <p>If you specify a - (minus sign) for the value of the <i>TabSpec</i> variable, the newform command assumes that the tab specification can be found in the first line read from standard input. The default <i>TabSpec</i> value is -8. A <i>TabSpec</i> value of -0 expects no tabs. If any are found, they are treated as having a value of -1.</p>
-l [<i>Number</i>]	<p>Sets the effective line length to the specified number of characters. If no <i>Number</i> variable is specified, the -l flag defaults to 72. The default line length without the -l flag is 80 characters. Note that tabs and backspaces are considered to be one character (use the -i flag to expand tabs to spaces). You must specify the -l flag before the -b and -e flags.</p>
-o [<i>TabSpec</i>]	<p>Replaces spaces in the input with a tab in the output, according to the tab specifications given. The default <i>TabSpec</i> value is -8. A <i>TabSpec</i> value of -0 means that no spaces are converted to tabs on output.</p>
-p [<i>Number</i>]	<p>Appends the specified number of characters to the beginning of a line when the line length is less than the effective line length. The default action is to append the number of characters that are necessary to obtain the effective line length. See also the -c flag.</p>
-s	<p>Removes leading characters on each line up to the first tab and places up to 8 of the removed characters at the end of the line. If more than 8 characters (not counting the first tab) are removed, the 8th character is replaced by an * (asterisk) and any characters to the right of it are discarded. The first tab is always discarded.</p> <p>The characters removed are saved internally until all other specified flags are applied to that line. The characters are then added to the end of the processed line.</p>

Note: The values for the **-a**, **-b**, **-e**, **-l** (lowercase L), and **-p** flags cannot be larger than **LINE_MAX** or 2048 bytes.

Examples

To convert from a file with:

- Leading digits
- One or more tabs
- Text on each line

to a file:

- Beginning with the text, all tabs after the first expanded to spaces
- Padded with spaces out to column 72 (or truncated to column 72)
- Leading digits placed starting at column 73

type the following:

```
newform -s -i -l -a -e filename
```

The **newform** command displays the following error message and stops if the **-s** flag is used on a file without a tab on each line.

newgrp Command

Purpose

Changes a user's real group identification.

Syntax

```
newgrp [ - ] [ -l ] [ Group ]
```

Description

The **newgrp** command changes a user's real group identification. When you run the command, the system places you in a new shell and changes the name of your real group to the group specified with the *Group* parameter. By default, the **newgrp** command changes your real group to the group specified in the **/etc/passwd** file.

Note: The **newgrp** command does not take input from standard input and cannot be run from within a script.

The **newgrp** command recognizes only group names, not group ID numbers. Your changes only last for the current session. You can only change your real group name to a group you are already a member of. If you are a root user, you can change your real group to any group regardless of whether you are a member of it or not.

Note: When you run the **newgrp** 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.

Flags

Ite	Description
-----	-------------

m

- Changes the environment to the login environment of the new group.
- l Indicates the same value as the - flag.

Security

Access Control: 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.

Exit Status

If the **newgrp** command succeeds in creating a new shell execution environment, regardless if the group identification was changed successfully, the exit status will be that of the current shell. Otherwise, the following exit value is returned:

Ite	Description
-----	-------------

m

- >0 An error occurred.

Examples

1. To change the real group ID of the current shell session to `admin`, enter:

```
newgrp admin
```

2. To change the real group ID back to your original login group, enter:

```
newgrp
```

Files

Item	Description
<code>/etc/passwd</code>	Indicates the password file; contains user IDs.

newkey Command

Purpose

Creates a new key in the `/etc/publickey` file.

Syntax

```
/usr/sbin/newkey [ -h HostName ] [ -u UserName ]
```

Description

The **newkey** command creates a new key in the `/etc/publickey` file. This command is normally run by the network administrator on the Network Information Services (NIS) master machine to establish public keys for users and root users on the network. These keys are needed for using secure Remote Procedure Call (RPC) protocol or secure Network File System (NFS).

The **newkey** command prompts for the login password of the user specified by the *UserName* parameter. Then, the command creates a new key pair in the `/etc/publickey` file and updates the **publickey** database. The key pair consists of the user's public key and secret key and is encrypted with the login password of the given user.

Use of this program is not required. Users may create their own keys using the **chkey** command.

You can use the System Management Interface Tool (SMIT) **smit newkey** fast path to run this command.

Flags

Item	Description
<code>-h <i>HostName</i></code>	Creates a new public key for the root user at the machine specified by the <i>HostName</i> parameter. Prompts for the root password of this parameter.
<code>-u <i>UserName</i></code>	Creates a new public key for a user specified by the <i>UserName</i> parameter. Prompts for the NIS password of this parameter.

Examples

1. To create a new public key for a user, enter:

```
newkey -u john
```

In this example, the **newkey** command creates a new public key for the user named john.

2. To create a new public key for the root user on host zeus, enter:

```
newkey -h zeus
```

In this example, the **newkey** command creates a new public key for the root user on the host named zeus.

Files

Item	Description
<code>/etc/publickey</code>	Stores encrypted keys for users.

news Command

Purpose

Writes system news items to standard output.

Syntax

```
news [ -a | -n | -s | Item ... ]
```

Description

The **news** command writes system news items to standard output. This command keeps you informed of news concerning the system. Each news item is contained in a separate file in the `/var/news` directory. Most users run the **news** command followed by the **-n** flag each time they log in by including it in their `$HOME/.profile` file or in the system's `/etc/profile` file. Any user having write permission to this directory can create a news item. It is not necessary to have read permission to create a news item.

If you run the **news** command without any flags, it displays every current file in the `/var/news` file, showing the most recent first. This command, used with the **-a** flag, displays all news items. If you specify the **-n** flag, only the names of the unread news items are displayed. Using the **-s** flag displays the number of unread news items. You can also use the *Item* parameter to specify the files that you want displayed.

Each file is preceded by an appropriate header. To avoid reporting old news, the **news** command stores a currency time. The **news** command considers your currency time to be the date the `$HOME/.news_time` file was last modified. Each time you read the news, the modification time of this file changes to that of the reading. Only news item files posted after this time are considered current.

Pressing the Interrupt (Ctrl-C) key sequence during the display of a news item stops the display of that item and starts the next. Pressing the Ctrl-C key sequence again ends the **news** command.

Note: News items can contain multibyte characters.

Flags

Item	Description
-a	Displays all news items, regardless of the currency time. The currency time does not change.
-n	Reports the names of current news items without displaying their contents. The currency time does not change.
-s	Reports the number of current news items without displaying their names or contents. The currency time does not change.

Examples

1. To display the items that have been posted since you last read the news, enter:

```
news
```

2. To display all the news items, enter:

```
news -a | pg
```

All of the news items display a page at a time (`| pg`), regardless of whether you have read them yet.

3. To list the names of the news items that you have not read yet, enter:

```
news -n
```

Each name is a file in the `/var/news` directory.

4. To display specific news items, enter:

```
news newusers services
```

This command sequence displays news about `newusers` and `services`, which are names listed by the `news -n` command.

5. To display the number of news items that you have not yet read, enter:

```
news -s
```

6. To post news for everyone to read, enter:

```
cp schedule /var/news
```

This copies the `schedule` file into the system `/var/news` directory to create the `/var/news/schedule` file. To do this, you must have write permission to the `/var/news` directory.

Files

Item	Description
<code>/usr/bin/news</code>	Contains the <code>news</code> command.
<code>/etc/profile</code>	Contains the system profile.
<code>/var/news</code>	Contains system news item files.
<code>\$HOME/.news_time</code>	Indicates the date the <code>news</code> command was last invoked.

next Command

Purpose

Shows the next message.

Syntax

```
next [ +Folder ] [ -header | -noheader ] [ -showproc CommandString | -noshowproc ]
```

Description

The `next` command displays the number the system will assign to the next message filed in a Message Handler (MH) folder. The `next` command is equivalent to the `show` command with the `next` value specified as the message.

The `next` command links to the `show` program and passes any switches on to the `showproc` program. If you link to the `next` value and call that link something other than `next`, your link will function like the `show` command, rather than like the `next` command.

The `show` command passes flags it does not recognize to the program performing the listing. The `next` command provides a number of flags for the listing program.

Flags

Item	Description
<code>+Folder</code>	Specifies the folder that contains the message you want to show.
<code>-header</code>	Displays a one-line description of the message being shown. The description includes the folder name and message number. This is the default.
<code>-help</code>	Lists the command syntax, available switches (toggles), and version information. Note: For MH, the name of this flag must be fully spelled out.
<code>-noheader</code>	Prevents display of a one-line description of each message being shown.
<code>-noshowproc</code>	Uses the <code>/usr/bin/cat</code> file to perform the listing. This is the default.
<code>-showproc</code> <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 associated with this command, see the `lssecattr` command or the `getcmdattr` subcommand.

Examples

1. To see the next message in the current folder, enter:

```
next
```

The system responds with a message similar to the following:

```
(Message schedule: 10)
```

The text of the message is also displayed. In this example, message 10 in the current folder `schedule` is the next message.

2. To see the next message in the `project` folder, enter:

```
next +project
```

The system responds with the text of the message and a header similar to the following:

```
(Message project: 5)
```

Files

Item	Description
<code>\$HOME/.mh_profile</code>	Specifies a user's MH profile.
<code>/usr/bin/next</code>	Contains the next command.

nfs.clean Command

Purpose

Stops NFS and NIS operations.

Syntax

```
/etc/nfs.clean [-d][-y][-t nfs|nis]
```

Description

The `/etc/nfs.clean` command is used to shut down operations of NFS, NIS, or both. This script is used by the `shutdown` command but can be used to stop operations of only NFS or NIS. By default, all NFS and NIS daemons are stopped.

This command is recommended instead of using `stopsvc -g nfs` since the `nfs.clean` command shuts daemons down in the correct order. The `stopsvc` command has no notion of stopping daemons of a group in the proper order. This can cause problems if the `statd` and `lockd` daemons are running and the `statd` daemon is stopped before the `lockd` daemon.

Flags

Item	Description
-d	Stops only server-specific daemons. Daemons that can run on clients are not stopped.
-y	Stops only server-specific NIS daemons. This flag is presumed if the <code>-d</code> flag is used.
-t	Stops only the specified system. If <code>-t nfs</code> is specified, only the NFS daemons are stopped. If <code>-t nis</code> is specified, only the NIS daemons are stopped.

Exit Status

Item	Description
0	Command completed successfully.
1	Argument error.

Examples

1. To stop all NFS and NIS daemons, type:

```
/etc/nfs.clean
```

2. To stop only NFS, type:

```
/etc/nfs.clean -t nfs
```

3. To stop only NFS service daemons, type:

```
/etc/nfs.clean -d -t nfs
```

Location

`/etc/nfs.clean`

nfs4cl Command

Purpose

Displays or modifies current NFSv4 statistics and properties.

Syntax

```
/usr/sbin/nfs4cl [subcommand] [path] [argument]
```

Description

Use the **nfs4cl** command to display all the fsid information on the client or modify filesystem options of an fsid.

Note: The **nfs4cl** updates affect newly accessed files in the filesystem. An unmount and remount are required to affect all previously accessed files.

Subcommands

resetfsoptions Subcommand

This subcommand resets all the options for the fsid back to the default options.

Note: The **cio** and **dio** options can be reset with the **resetfsoptions** subcommand, but the **cio** and **dio** behavior is not actually turned off until the NFS filesystem is unmounted and then remounted.

setfsoptions Subcommand

This subcommand will take a path and an argument. The path specifies the target fsid structure and the argument is the file system options. It will set the internal fsid to use the options specified by the argument. Here is the list of possible arguments:

Item	Description
rw	Specifies that the files or directories that bind to this path (fsid) are readable and writable.
ro	Specifies that the files or directories that bind to this path (fsid) are read only.
acdirmax	Specifies the upper limit for the directory attribute cache time out value.
acdirmin	Specifies the lower limit for the directory attribute cache time out value.
acregmax	Specifies the upper limit for the file attribute cache time out value.
acregmin	Specifies the lower limit for the file attribute cache time out value.
cio	Specifies the filesystem to be mounted for concurrent readers and writers. I/O on files in this filesystem behave as if the file was opened with O_CIO specified in the open() system call.
cior	Specifies to allow read-only files to open in the file system. I/O on files in this filesystem will behave as if they had been opened with O_CIO O_CIOR specified in the open() system call.
dio	Specifies that I/O on the filesystem behaves as if all of the files were opened with O_DIRECT specified in the open() system call.
hard	Specifies that this fsid will use hard mount semantics.
intr	Specifies that the fsid operations are interruptible.

Item	Description
maxpout=value	Specifies the pageout level for files on this filesystem at which threads should be slept. If maxpout is specified, minpout must also be specified. This value must be non-negative and greater than minpout . The default is the kernel maxpout level.
minpout=value	Specifies the pageout level for files on this filesystem at which threads should be readied. If minpout is specified, maxpout must also be specified. This value must be non-negative. The default is the kernel minpout level.
noac	Does not use attribute cache.
nocto	Specifies no close-to-open consistency.
nointr	Specifies that the fsid is non-interruptible.
prefer=servername	Administratively sets the preferred server to use when data exists at multiple server locations. The server name can be in short name, long name, IPv4, or IPv6 format, but the client must be able to resolve the server name when the <code>nfs4cl</code> command is run.
rbr	Utilizes the release-behind-when-reading capability. When sequential reading of a file in this filesystem is detected, the real memory pages used by the file will be released once the pages are copied to internal buffers.
rsize	Specifies the read size for the RPC calls to the server.
retrans	Specifies the number of RPC retransmits to attempt with soft semantics.
soft	Specifies the fsid operation that will use soft mount semantics.
timeo	Specifies the time out value for the RPC calls to the server.
wsize	Specifies the write size for the RPC calls to the server.
nodircache	Does not use directory cache.

showfs Subcommand

This subcommand displays filesystem specific information on the server that is currently accessed by the client. The information includes server address, remote path, fsid, and local path. If path is provided, additional information, such as `fs_locations` and `fsid` options, are displayed.

showstat Subcommand

This subcommand shows information similar to what the **df** command prints out for each fsid that exists on the client. The information includes fields such as, Filesystem, 512-blocks, Free, %Used, Iused, %Iused, and Mounted on.

delegreturn Subcommand

This subcommand accepts file path as its input argument. This subcommand will allow a system administrator to instruct NFS v4 client to return delegations on the file specified by the input path name.

help Subcommand

This subcommand prints the usage statement.

Exit Status

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

Examples

1. To display all the fsid structure on the client, type:

```
nfs4cl showfs
```

2. To set the file system options of **/mnt/usr/sbin** to include only `retrns=3`, type:

```
nfs4cl setfsoptions /mnt/usr/sbin retrns=3
```

3. To reset the filesystem options for **/mnt/use/sbin**, type:

```
nfs4cl resetfsoptions /mnt/user/sbin
```

4. To show **df** command output for **/mnt/usr/sbin**, type:

```
nfs4cl showstat /mnt/usr/sbin
```

5. To make the client failover to server `boo` when replication occurs in **/mnt/usr/sbin**, type:

```
nfs4cl setfsoptions /mnt/usr/sbin prefer=boo
```

Location

/usr/sbin/nfs4cl

nfs4smctl Command

Purpose

Administers revocation of NFSv4 State.

Syntax

```
/usr/sbin/nfs4smctl -r [hostname | IP_address]
```

Description

Administers revocation of NFS v4 State.

Flags

Item	Description
<code>-r hostname</code> <code>IP_address</code>	Specifies the client of which state is to be revoked using either the <i>hostname</i> or <i>IP_address</i> parameter.

Files

Item	Description
<code>/usr/sbin/nfs4smctl</code>	Location of the <code>nfs4smctl</code> command.

nfsauthreset Command

Purpose

Notifies the Network File System (NFS) kernel extension to destroy the appropriate Generic Security Service API (GSSAPI) credentials from the kernel credentials cache.

Syntax

nfsauthreset

Description

To mark the cached context, the **nfsauthreset** command depends on whether a Process Authentication Group (PAG) is set in the process. If a PAG is set in the process, it marks the cached GSSAPI context having the same User ID (UID) and PAG to be destroyed. Otherwise, it marks the cached GSSAPI context having the same UID to be destroyed.

Examples

To destroy the cached kernel credentials after you have specified the **kinit** and the **kdestroy** commands, enter:

```
nfsauthreset
```

Files

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

nfsd Daemon

Purpose

Services client requests for file system operations.

Syntax

```
/usr/sbin/nfsd [-a | -p { tcp | udp } ] [ -c max_connections ] [ -gp on | off ] [ -gpx count ]  
[ -gpbypass ] [ -w max_write_size ] [ -r max_read_size ] [ -root directory ] [ -public directory ]  
nservers
```

```
/usr/sbin/nfsd -getnodes
```

```
/usr/sbin/nfsd -getreplicas
```

Description

The **nfsd** daemon runs on a server and handles client requests for file system operations.

Each daemon handles one request at a time. Assign the maximum number of threads based on the load you expect the server to handle.

The **nfsd** daemon is started and stopped with the following System Resource Controller (**SRC**) commands:

```
startsrc -s nfsd
```

```
stopsrc -s nfsd
```

To change the number of daemons started with the SRC commands, use the **chnfs** command. To change the parameters of an SRC controlled daemon, use the **chssys** command.

Note: If the number of **nfsd** daemons is not sufficient to serve the client, a nonidempotent operation error is returned to the client. For example, if the client removes a directory, an ENOENT error is returned even though the directory on the server is removed.

Flags

Item	Description
-a	Specifies UDP and TCP transport will be serviced.
-c <i>max_connections</i>	Specifies the maximum number of TCP connections allowed at the NFS server.
-gp on off	Controls the NFSv4 Grace Period enablement. The possible values are <code>on</code> or <code>off</code> . If no <code>-gp</code> option is specified, the grace period is disabled by default.
-gpbypass	Controls the NFSv4 Grace Period bypass. When this option is specified, the grace period will be bypassed regardless of how the <code>-gp</code> option is specified.
-gpx <i>count</i>	Controls the NFSv4 Grace Period automatic extension. The <i>count</i> parameter specifies the total number of automatic extensions allowed for the grace period. If no <code>-gpx</code> option is specified, the number of allowed automatic extensions defaults to 1. A single extension cannot extend the grace period for more than the length of the NFSv4 lease period. The NFSv4 subsystem uses runtime metrics (such as the time of the last successful NFSv4 reclaim operation) to detect reclamation of the state in progress, and extends the grace period for a length of time up to the duration of the given number of iterations.
<i>nserver</i>	Specifies the maximum number of concurrent requests that the NFS server can handle. This concurrency is achieved by dynamic management of threads within the NFS server, up to the maximum. The default maximum is 3891. The chnfs , chsys , or nfs command is used to change the maximum. Changing the maximum setting from the default is not recommended as this may limit server performance.
-p <i>tcp</i> or -p <i>udp</i>	Transports both UDP and TCP to the NFS clients (default). You can only specify UDP or TCP. For example, if -p tcp is used, the NFS server only accepts NFS client requests using the TCP protocol.
-r <i>max_read_size</i>	Specifies for NFS Version 3, the maximum size allowed for file read requests. The default and maximum allowed is 64K.
-w <i>max_write_size</i>	Specifies for NFS Version 3, the maximum size allowed for file write requests. The default and maximum allowed is 64K.
-root <i>directory</i>	Specifies the directory which should be the root node the NFS version 4 exported filesystem. By default, the root node is <code>/</code> . If the root node is set to something other than <code>/</code> , use chnfs -r to reset the node to <code>/</code> . This flag may be used while <code>nfsd</code> is running to change the root node, but only if no filesystems are currently exported. This flag might be removed in a future release. Use chnfs -r instead.
-public <i>directory</i>	Specifies the directory which should be the public node of the NFS version 4 exported filesystem. By default, the public node is the same as the root node. This flag may be use while <code>nfsd</code> is running to change the public node. The public node must be a descendant of the root node. This flag might be removed in a future release. Use chnfs -p instead.
-getnodes	Prints the current root and public nodes for the NFS version 4 server. This option will not cause the NFS server daemon to start.
-getreplicas	Prints the current replication enablement mode. If replicas have been specified for the <code>nfsroot</code> , they will be displayed.

Examples

1. To start **nfsd** daemons using an **src** command, enter:

```
startsrc -s nfsd
```

In this example, the `startsrc -s nfsd` entry starts the number of daemons specified in the script.

2. To change the number of daemons running on your system, enter:

```
chssys -s nfsd -a 6
```

In this example, the `chssys` command changes the number of `nfsd` daemons running on your system to 6.

nfshostkey Command

Purpose

Configures the host keys for an Network File System (NFS) server.

Syntax

```
nfshostkey -l | -L | { -p principal -f file } | { -a principal -i address } | { -d principal -i address }
```

Description

An NFS server (or full client) using RPCSEC_GSS RPC security must be able to acquire credentials for its host principal to accept requests. Use the **nfshostkey** command to configure this information.

All full clients and NFS servers must have a primary host principal. The following is the format of the host principal that the **nfshostkey** command sets:

```
nfs/fully_qualified_domain_name
```

After you set the primary host principal, you can use the **nfshostkey** command to set additional host principals for other network addresses. The server searches the list of addresses to find the one that an incoming request was sent to and use the appropriate principal. If none is found, the primary principal is used. The secondary host principals must have entries in the same keytab file that was passed in for the primary principal. They will not be used by full clients.

Flags

Item	Description
<code>-a</code>	Adds a new secondary host principal.
<code>-d</code>	Deletes a secondary host principal.
<code>-f file</code>	Specifies the path to a keytab file for the host principals.
<code>-i address</code>	Specifies the IP address corresponding to the secondary principal.
<code>-l</code>	Lists the primary host principal and keytab.
<code>-L</code>	Lists the primary host principal, keytab, and secondary host principals.
<code>-p principal</code>	Specifies the principal for this host.

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 set a primary host principal, enter:

```
nfshostkey -p principal -f keytab file
```

2. To add a secondary host principal, enter:

```
nfshostkey -a -p principal -i ip address
```

3. To delete a host principal, enter:

```
nfshostkey -d -p principal -i ip address
```

nfshostmap Command

Purpose

Manage mapping from hosts to principals for an nfs client.

Syntax

```
/usr/sbin/nfshostmap -a hostname alias1 alias2 | -d hostname | -e hostname alias1 alias2 | -l
```

Description

All hosts defined as aliases will be mapped to the host defined as a *hostname* when constructing a kerberos request to the server. This is useful if, for example, a server has interfaces `wizard.sub.austin.ibm.com` and `wizard.austin.ibm.com`; if this server's kerberos principal is `wizard.austin.ibm.com`, `nfshostmap -a wizard.austin.ibm.com wizard.sub.austin.ibm.com` run on the client will take care of this problem.

This modifies `/etc/nfs/princmap`, which is read by the `gssd` daemon on startup.

Flags

Item	Description
<code>-a <i>hostname alias1 alias2</i></code>	Adds a mapping from the aliases to <i>hostname</i> ,
<code>-d <i>hostname</i></code>	Deletes all aliases for <i>hostname</i> .
<code>-e <i>hostname alias1 alias2</i></code>	Removes all previous mappings for <i>hostname</i> and replaces them with the given alias list.
<code>-l</code>	Prints the existing state of the respective files on 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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

nfso Command

Purpose

Manages Network File System (NFS) tuning parameters.

Syntax

nfso [**-p** | **-r**] [**-c**] { **-o** *Tunable* [=*newvalue*] }

nfso [**-p** | **-r**] { **-d** *Tunable* }

nfso [**-p** | **-r**] **-D**

nfso [**-p** | **-r**] **-a** [**-F**] [**-c**]

nfso **-h** [*Tunable*]

nfso **-l** [*hostname*]

nfso [**-F**] **-L** [*Tunable*]

nfso [**-F**] **-x** [*Tunable*]

nfso [**-@** *WparName*] [**-p** | **-r**] **-a** [**-c**]

nfso [**-@** *WparName*] [**-p** | **-r**] [**-c**] { **-o** *Tunable* [=*newvalue*] }

nfso **-H** {*ha operation*}

Note: Multiple flags **-o**, **-d**, **-x**, and **-L** are allowed.

Description

Use the **nfso** command to configure Network File System tuning parameters. The **nfso** command sets or displays current or next boot values for Network File System 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

Extreme care should be taken when using this command. If used incorrectly, the **nfso** command can make your system inoperable.

Before modifying any tunable parameter, you should first carefully read about all its characteristics in the Tunable Parameters section below, 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 should probably never change this parameter unless specifically directed by AIX development.

Flags

Item	Description
-a	Displays the current, reboot (when used in conjunction with -r) or permanent (when used in conjunction with -p) value for all tunable parameters, one per line in pairs <i>Tunable = Value</i> . For the permanent options, a value is only displayed for a parameter if its reboot and current values are equal. Otherwise NONE displays as the value.

Item	Description
-c	Changes the output format of the nfso command to colon-delineated format.
-d <i>Tunable</i>	Sets the <i>Tunable</i> variable back to its default value. If a <i>Tunable</i> needs to be changed that is, . it is currently not set to its default value) and is of type Bosboot or Reboot, or if it is of type Incremental and has been changed from its default value, and -r is not used in combination, it will not be changed but a warning displays instead.
-D	Sets all <i>Tunable</i> variables back to their default value. If <i>Tunables</i> needing to be changed are of type Bosboot or Reboot, or are of type Incremental and have been changed from their default value, and the -r flag is not used in combination, they will not be changed but warnings display instead.
-F	Forces restricted tunable parameters to be displayed when the options -a , -L or -x are specified on the command line. 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 <i>Tunable</i> parameter if one is specified. Otherwise, displays the nfso command usage statement.
-H {ha operation}	Runs an high availability (HA) operation. HA operations follow: enable_ha Turns on the HA function. disable_ha Turns off the HA function. sm_register <hostname> PowerHA SystemMirror registers this host. sm_unregister <hostname> PowerHA SystemMirror unregisters this host. sm_gethost PowerHA SystemMirror gets the host. dump_dupcache <log device> Dumps HA dupcache.
-l <i>hostname</i>	Allows a system administrator to release NFS file locks on an NFS server. The <i>hostname</i> variable specifies the host name of the NFS client that has file locks held at the NFS server. The nfso -l command makes a remote procedure call to the NFS server's rpc.lockd network lock manager to request the release of the file locks held by the <i>hostname</i> NFS client. If there is an NFS client that has file locks held at the NFS server and this client has been disconnected from the network and cannot be recovered, the nfso -l command can be used to release those locks so that other NFS clients can obtain similar file locks. Note: The nfso command can be used to release locks on the local NFS server only.

Item**Description****-L** [*Tunable*]

Lists the characteristics of one or all *Tunable*, one per line, using the following format:

NAME	UNIT	TYPE	CUR	DEF	BOOT	MIN	MAX

DEPENDENCIES							

portcheck		D	0	0	0	0	1
On/Off							

udpchecksum		D	1	1	1	0	1
On/Off							

nfs_socketsize		D	600000	600000	600000	40000	1M
Bytes							

nfs_tcp_socketsize		D	600000	600000	600000	40000	1M
Bytes							

...							
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							

-o

Tunable [=*newvalue*
]

Displays the value or sets *Tunable* to *newvalue*. If a tunable needs to be changed (the specified value is different than current value), and is of type Bosboot or Reboot, or if it is of type Incremental and its current value is bigger than the specified value, and **-r** is not used in combination, it will not be changed but a warning displays instead.

When **-r** is used in combination without a new value, the nextboot value for the *Tunable* displays. When **-p** is used in combination without a *newvalue*, a value displays only if the current and next boot values for the *Tunable* are the same. Otherwise NONE displays as the value.

-p

Makes changes apply to both current and reboot values, when used in combination with **-o**, **-d** or **-D**, that is, it 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 cannot 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, it turns on the updating of the **/etc/tunables/nextboot** file. If any parameter of type Bosboot is changed, the user is 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.

Item	Description
-x [<i>Tunable</i>]	Lists characteristics of one or all tunables, one per line, using the following (spreadsheet) format:

```
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
```

-@ <i>WparName</i>	Sets or displays tunables for the specified workload partition. The -@ flag can only be used when the nfso command is run in the global partition.
---------------------------	--

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 (using a command line specifying the **-r** or **-p** options), results in an error log entry that identifies the list of these modified tunables.

If you make any change (with **-o**, **-d**, or **-D**) to a parameter of type Mount, it results in a warning message that the change is only effective for future mountings.

If you make any change (with **-o**, **-d** or **-D**) to a parameter of type Connect, it results in **inetd** being restarted, and a warning message that the change is only effective for future socket connections.

If you make any change (with **-o**, **-d**, or **-D**) to a parameter of type Bosboot or Reboot without **-r**, it results in an error message.

If you make any change (with **-o**, **-d**, or **-D** but without **-r**) to the current value of a parameter of type Incremental with a new value smaller than the current value, it results in an error message.

Note: Tunable variables that apply to the entire system can not be modified within a workload partition.

Note: When the **nfso** command is run within a workload partition (or if the **-@** flag is specified), only the following tunables can be set with the **-o** flag:

- **nfs_dynamic_retrans**
- **nfs_iopace_pages**
- **nfs_use_reserved_port**
- **nfs_v4_fail_over_timeout**
- **utf8_validation**
- **nfs_auth_rbr_trigger**
- **client_delegation**

Tunable Parameters Type

All the tunable parameters manipulated by the tuning commands (**no**, **nfso**, **vmo**, **ioo**, **schedo**, and **raso**) 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

Item	Description
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 **nfso** command only includes Dynamic, Mount, and Incremental types.

Compatibility Mode

When running in pre 5.2 compatibility mode (controlled by the **pre520tune** attribute of **sys0**, see [AIX 5.2 compatibility mode](#)), 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 5L Version 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 details about the new 5.2 mode.

Tunable Parameters

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

Note: Starting with AIX Version 6.1 with the 6100-02 Technology Level, the following parameters are obsolete because the network file system (NFS) and the virtual memory manager (VMM) dynamically tunes the number of **buf** structures and page device tables (PDTs) based on workload:

- nfs_v2_pdts
- nfs_v2_vm_bufs
- nfs_v3_pdts
- nfs_v3_vm_bufs
- nfs_v4_pdts
- nfs_v4_vm_bufs

Item	Description
client_delegation	<p>Purpose: Determine if the NFS version 4 client will accept delegations for open files.</p> <p>Tuning: A value of 0 disables delegations. A value of 1 enables delegations.</p>
nfs_hang_log	<p>Purpose: Sets the priority at which the NFS mount messages that are hung get logged to the <code>syslog</code> log file.</p> <p>Tuning: The values are in the range 1 - 7.</p> <ul style="list-style-type: none"> • 1: LOG_ALERT • 2: LOG_CRIT • 3: LOG_ERR • 4: LOG_WARNING • 5: LOG_NOTICE • 6: LOG_INFO • 7: LOG_DEBUG <p>The default value is 6.</p>
nfs_max_read_size	<p>Purpose: Allows the system administrator to control the NFS RPC sizes at the server.</p> <p>Tuning: Useful when all clients must have changes in the read sizes, and when the clients cannot be changed. Use the values of the client mount as the default value. The default value is required to reduce the V3 read sizes when the mounts cannot be manipulated directly on the clients, in particular during the NIM installations on networks where the network is dropping packets with the default read sizes. In this case, set the maximum size of 512 KB to a smaller value such that the value works on the network. This parameter is also useful when the network devices are dropping packets and a generic change is desired for communications with the server. The default value is 64 KB and the maximum value is 512 KB.</p>
nfs_max_write_size	<p>Purpose: Allows the system administrator to control the NFS RPC sizes at the server.</p> <p>Tuning: Useful when all clients must have changes in the write sizes, and when the clients cannot be changed. Use the values of the client mount as the default value. The default value is required to reduce the V3 read sizes when the mounts cannot be manipulated directly on the clients, in particular during the NIM installations on networks where the network is dropping packets with the default write sizes. In this case, set the maximum size of 512 KB to a smaller value such that the value works on the network. This parameter is also useful when the network devices are dropping packets and a generic change is desired for communications with the server. The default value is 64 KB and the maximum value is 512 KB.</p>

Item	Description
nfs_rfc1323	<p>Purpose: Enables very large TCP window size negotiation (greater than 65535 bytes) to occur between systems.</p> <p>Tuning: If using the TCP transport between NFS client and server, and both systems support it, this allows the systems to negotiate a TCP window size in a way that will allow more data to be in-flight between the client and server. This increases the throughput potential between client and server. Unlike the rfc1323 option of the no command, this only affects NFS and not other applications in the system. Value of 0 means this is disabled, and value of 1 means it is enabled. If the no command parameter rfc1323 is already set, this NFS option does not need to be set.</p>
nfs_securenfs_authtimeout	<p>Purpose: Sets the number of seconds for which a DES credential.</p> <p>Tuning: Value of 0 disables DES credential timeouts.</p>
nfs_server_base_priority	<p>Purpose: Sets the base priority of nfsd daemons.</p> <p>Tuning: By default, the nfsd daemons run with a floating process priority. Therefore, as they increase their cumulative CPU time, their priority will change. This parameter can be used to set a static parameter for the nfsd daemons. The value of 0 represents the floating priority (default). Other values within the acceptable range will be used to set the priority of the nfsd daemon when an NFS request is received at the server. This option can be used if the NFS server is overloading the system (lowering or making the nfsd daemon less favored). It can also be used if you want the nfsd daemons be one of the most favored processes on the server. Use caution when setting the parameter because it can render the system almost unusable by other processes. This situation can occur if the NFS server is very busy and will essentially lock out other processes from having run time on the server.</p>

Item	Description
nfs_server_cread	<p>Purpose: This option allows the NFS server to be very aggressive about the reading of a file. The NFS server can only respond to the specific NFS-read request from the NFS client. However, the NFS server can read data in the file which exists immediately after the current read request. This is normally referred to as read-ahead. The NFS server does read-ahead by default.</p> <p>Tuning: May be useful in cases where server memory is low and a lot of disk-to-memory activity is going on. With the nfs_server_cread option enabled, the NFS server becomes very aggressive about doing read-ahead for the NFS client. If value is 1, then aggressive read-ahead is done; If value is 0, normal system default read-ahead methods are used. Normal system read-ahead is controlled by VMM (for JFS file systems) and JFS2 (for JFS2 file systems). This more aggressive top-half read-ahead enabled via the nfs_server_cread option is less susceptible to read-ahead breaking down due to out-of-order requests (which are typical in the NFS server case). When the mechanism is activated, it will read an entire cluster (128 KB, the LVM logical track group size) at a time.</p>
nfs_server_close_delay	<p>Purpose Determines if the NFS version 4 server must avoid sending an NFS4ERR_DELAY response if the expected delay is not too long. If NFS clients are used that pause applications for a long time when encountering a NFS4ERR_DELAY response from the server, the server attempts to process the delay on the server by using the nfs_server_close_delay option, which avoids pausing the application.</p> <p>Tuning A value of 0 turns off this feature. The default value is 0. A value of 1 enables local processing of short delays on the server side.</p>
nfs_use_reserved_ports	<p>Purpose: Specifies using non-reserved IP port number.</p> <p>Tuning: Value of 0 will use non-reserved IP port number when the NFS client communicates with the NFS server.</p>
nfs_v3_server_readdirplus	<p>Purpose: Determines if REaddirPLUS calls are supported by the server.</p> <p>Tuning: Value of 0 disables REaddirPLUS processing.</p>

Item	Description
nfs_v4_fail_over_timeout	<p>Purpose: Specifies a time out period which the NFS version 4 client operation will fail over to the replica provided by the NFS version 4 server. Measured in seconds.</p> <p>Tuning: If value of 0 is specified, the timeout value will be the timeout value for tcp multiplied by 4. Values from 1 to 4 are reserved and the NFS version 4 client will treat it as 0. NFS version 4 allows client to fail over to other replica server if the main server is not responding. This value will determine how long a client has to wait for a respond from the server before it switch all the NFS version 4 request for that fsid to other replica server.</p>
portcheck	<p>Purpose: Checks whether an NFS request originated from a privileged port.</p> <p>Tuning: Value of 0 disables the port-checking that is done by the NFS server. A value of 1 directs the NFS server to do port checking on the incoming NFS requests. This is a configuration decision with minimal performance consequences.</p>
server_delegation	<p>Purpose: Determine if the NFS version 4 server will issue read delegations for open files.</p> <p>Tuning: A value of 0 disables delegation granting. A value of 1 enables delegation granting.</p>
utf8_validation	<p>Purpose: Determine if the NFS version 4 client and server will check string data for UTF-8 correctness.</p> <p>Tuning: A value of 0 disables the UTF-8 checking. A value of 1 enables the UTF-8 checking.</p>
gss_window	<p>Purpose: Enable or disable GSS window size checking.</p> <p>Tuning: A value of 0 disables GSS window size checking. A value of 1 enables GSS window size checking. The default value is 1.</p>

Examples

1. To set the **portcheck** tunable parameter to a value of zero, type:

```
nfsd -o portcheck=0
```

2. To set the **udpchecksum** tunable parameter to its default value of 1 at the next reboot, type:

```
nfsd -r -d udpchecksum
```

3. To print, in colon-delimited format, a list of all tunable parameters and their current values, type:

```
nfsd -a -c
```

4. To list the current and reboot value, range, unit, type and dependencies of all tunables parameters managed by the **nfs** command, type:

```
nfs -L
```

5. To display help information on **nfs_tcp_duplicate_cache_size**, type:

```
nfs -h nfs_tcp_duplicate_cache_size
```

6. To permanently turn off **nfs_dynamic_retrans**, type:

```
nfs -p -o nfs_dynamic_retrans=0
```

7. To list the reboot values for all Network File System tuning parameters, type:

```
nfs -r -a
```

8. To list (spreadsheet format) the current and reboot value, range, unit, type and dependencies of all tunables parameters managed by the **nfs** command, type:

```
nfs -x
```

nfsrgyd daemon

Purpose

Services translation requests between names and ids from servers and clients using NFS V4 or RPCSEC-GSS.

Syntax

```
nfsrgyd [ -f ] [ -T heartBeatInt ]
```

Description

The `nfsrgyd` daemon provides a name translation service for NFS servers and clients. This daemon must be running in order to perform translations between NFS string attributes and UNIX numeric identities.

The environment variables `NFS_NOBODY_USER` and `NFS_NOBODY_GROUP` affect the anonymous user and group owner strings used in the name translations. If these environment variables are not set, their default values of `nobody` will be used. They may be set in the file `/etc/environment`, or on the command line before `nfsrgyd` is started.

The local NFS domain must be set before running the `nfsrgyd` daemon. This may be set by using the `chnfsdom` command.

Note: The `nfsrgyd` daemon uses an ephemeral port.

Flags

Item	Description
-f	Creates a new process to flush the name translation cache and exits.
-T	Specifies the time interval between subsequent LDAP server reconnections. The valid values are 60-3600 seconds. The default value is 300.

Examples

1. The `nfsrgyd` daemon is started from the `/etc/rc.nfs` file. Using the following System Resource Controller (SRC) commands, you can start and stop the `nfsrgyd` daemon:

```
startsrc -s nfsrgyd
stopsrc -s nfsrgyd
```

2. To change the parameters passed to the `nfsrgyd` daemon using the `chssys` command, enter:

```
chssys -s nfsrgyd -a "-T 360"
```

Tip: The change does not take effect until the daemon is restarted. The value of the `heartBeatInt` interval will then be persistent after the `nfsrgyd` daemon is restarted.

Security

Users must have root authority.

Files

Item	Description
<code>/etc/environment</code>	Contains NFS environment variables.

nfsstat Command

Purpose

Displays statistical information about the Network File System (NFS) and Remote Procedure Call (RPC) calls.

Syntax

```
/usr/sbin/nfsstat [ -@ WparName ] [ -c ] [ -d ] [ -s ] [ -n ] [ -r ] [ -m [-i] ] [ -4 ] [ -z ] [ -t ] [ -b ] [ -g ]
```

Description

The `nfsstat` command displays statistical information about the NFS and Remote Procedure Call (RPC) interfaces to the kernel. You can also use this command to reinitialize this information. If no flags are given, the default is the `nfsstat -csnr` command. With this option, the command displays everything, but reinitializes nothing.

RPC Server Information

The server RPC display includes the following fields:

Item	Description
<code>calls</code>	Total number of RPC calls received. This number includes the NFS version 4 calls if the <code>-4</code> flag is used. Otherwise, only the version 2 and version 3 total is displayed.
<code>badcalls</code>	Total number of calls rejected by the RPC layer. This number includes the NFS version 4 calls if the <code>-4</code> flag is used. Otherwise, only the version 2 and version 3 total is displayed.
<code>nullrecv</code>	Number of times an RPC call was not available when it was thought to be received.
<code>badlen</code>	Number of RPC calls with a length shorter than a minimum-sized RPC call.
<code>xdrCALL</code>	Number of RPC calls whose header could not be XDR decoded.
<code>dupchecks</code>	Number of RPC calls that looked up in the duplicate request cache.

Item	Description
dupreqs	Number of duplicate RPC calls found.

RPC Client Information

Item	Description
calls	Total number of RPC calls made
badcalls	Total number of calls rejected by the RPC layer
badxid	Number of times a reply from a server was received that did not correspond to any outstanding call
timeouts	Number of times a call timed out while waiting for a reply from the server
newcreds	Number of times authentication information had to be refreshed
badverfs	The number of times the call failed due to a bad verifier in the response.
timers	The number of times the calculated time-out value was greater than or equal to the minimum specified timed-out value for a call.
cantconn	The number of times the call failed due to a failure to make a connection to the server.
nomem	The number of times the calls failed due to a failure to allocate memory.
interrupts	The number of times the call was interrupted by a signal before completing.
retrans	The number of times a call had to be retransmitted due to a time-out while waiting for a reply from the server. This is applicable only to RPC over connection-less transports
dupchecks	The number of RPC calls that looked up in the duplicate request cache.
dupreqs	The number of duplicate RPC calls found.

NFS Server Information

The NFS server displays the number of NFS calls received (calls) and rejected (badcalls), as well as the counts and percentages for the various kinds of calls made.

NFS Client Information

The NFS client information displayed shows the number of calls sent and rejected, as well as the number of times a CLIENT handle was received (clgets), the number of times the client handle had no unused entries (clatoomany), and a count of the various kinds of calls and their respective percentages.

NFS Registry Daemon Information

The NFS registry daemon display shows the number of requests from the client and server to translate between UID/GID and string names.

-m Information

The **-m** flag displays information about **mount** flags set by **mount** options, **mount** flags internal to the system, and other **mount** information. See the [mount](#) command for more information.

The following **mount** options are set by **mount** flags:

Item	Description
auth	Provides one of the following values: none No authentication. unix UNIX style authentication (UID, GID). des des style authentication (encrypted timestamps).
hard	Hard mount.
soft	Soft mount.
intr	Interrupts allowed on hard mount.
nointr	No interrupts allowed on hard mount.
noac	Client is not caching attributes.
rsize	Read buffer size in bytes.
wsize	Write buffer size in bytes.
retrans	NFS retransmissions.
nocto	No close-to-open consistency.
llock	Local locking being used (no lock manager).
grpuid	Group ID inheritance.
vers	NFS version.
proto	Protocol.

The following **mount** options are internal to the system:

Item	Description
printed	Not responding message printed.
down	Server is down.
dynamic	Dynamic transfer size adjustment.
link	Server supports links.
symlink	Server supports symbolic links.
readdir	Use readdir instead of readdirplus .

-t Information

The **-t** flag displays information relating to translation requests of the NFS identity mapping subsystem.

Item	Description
ids_to_strin gs	The number of id-to-string translation requests.
strings_to_i ds	The number of string-to-id translation requests.
resolve_erro rs	The number of failed translation requests due to missing data.
badowners	The number of failed translation requests due to invalid inputs.

Item	Description
cache_hits	The number of translation requests handled by the translation cache.
cache_misses	The number of translation requests not handled by the translation cache.
cache_entries	The number of entries in the translation cache.
cache_recycles	The number of entries in the translation cache that have expired.

Flags

Item	Description
-@ <i>WparName</i>	Displays statistics for the specified workload partition. The -@ flag can only be used when the nfsstat command is executed in the global partition. If the -@ flag is not used when the nfsstat command is executed from a workload partition, the statistics for the current workload partition are displayed. If the -@ flag is not used when the nfsstat command is executed from the global partition, the sum statistics of all active workload partitions (and the global partition) are displayed. Note: If you use the -@ WparName flag together with the -m flag, the nfsstat command displays statistics for the global partition instead of the specified workload partition.
-b	Displays additional statistics for the NFS version 4 server.
-c	Displays client information. Only the client side NFS and RPC information is printed. Allows the user to limit the report to client data only. The nfsstat command provides information about the number of RPC and NFS calls sent and rejected by the client. To print client NFS or RPC information only, combine this flag with the -n or -r option.
-d	Displays information related to NFS version 4 delegations.

Item	Description
-g	<p>Displays RPCSEC_GSS information. The RPCSEC_GSS information sections contain:</p> <p>activegss Active RPCSEC_GSS contexts</p> <p>discardgss Discarded RPCSEC_GSS messages</p> <p>krb5est Established krb5 contexts</p> <p>krb5iest Established krb5i contexts</p> <p>krb5pest Established krb5p contexts</p> <p>expgss Expired RPCSEC_GSS contexts</p> <p>badaccept gss_accept_sec_context failures</p> <p>badverify gss_verify_mic failures</p> <p>badgetmic gss_get_mic failures</p> <p>badwrap gss_wrap failures</p> <p>badunwrap gss_unwrap failures</p>
-m	<p>Displays statistics for each NFS file system mounted along with the server name, mount flags, current read and write sizes, retransmission count, and the timers used for dynamic retransmission.</p> <p>Note: If you provide the -m option when you use the nfsstat command, you always get statistics for the global partition.</p>
-i	When used along with -m , it prints the server's ip address as well. This flag is valid only along with the -m flag.
-n	Displays NFS information . Prints NFS information for both the client and server. To print only the NFS client or server information, combine this flag with the -c and -s options.
-r	Displays RPC information.
-s	Displays server information.
-t	Displays statistics related to translation requests of the NFS identity mapping subsystem. To print only the NFS client or server information, combine with the -c and -s options.
-4	When combined with the -c , -n , -s , or -z flags, includes information for the NFS version 4 client or server, in addition to the existing NFS version 2 and version 3 data. Without this option, the output is identical to output from the nfsstat command in AIX versions prior to version 5.3.
-z	Re-initializes statistics. This flag is for use by the root user only and can be combined with any of the above flags to zero particular sets of statistics after printing them.

Examples

1. To display information about the number of RPC and NFS calls sent and rejected by the client, enter:

```
nfsstat -c
```

2. To display and print the client NFS call-related information, enter:

```
nfsstat -cn
```

3. To display statistics for each NFS mounted file system, enter:

```
nfsstat -m
```

4. To display and print RPC call-related information for the client and server, enter:

```
nfsstat -r
```

5. To display information about the number of RPC and NFS calls received and rejected by the server, enter:

```
nfsstat -s
```

6. To reset all call-related information to zero on the client and server, enter:

```
nfsstat -z
```

Note: You must have root user authority to use the **-z** flag.

7. To display information about the NFS client statistics for workload partition **abc**, enter:

```
nfsstat -@ abc -cn
```

nice Command

Purpose

Runs a command at a lower or higher priority.

Syntax

```
nice [ -i Increment ] -n Increment ] Command [ Argument ... ]
```

Description

The **nice** command lets you run a command at a priority lower than the command's normal priority. The *Command* parameter is the name of any executable file on the system. If you do not specify an *Increment* value the **nice** command defaults to an increment of 10. You must have root user authority to run a command at a higher priority. The priority of a process is often called its nice value.

The nice value can range from -20 to 19, with 19 being the lowest priority. For example, if a command normally runs at a priority of 10, specifying an increment of 5 runs the command at a lower priority, 15, and the command runs slower. The **nice** command does not return an error message if you attempt to increase a command's priority without the appropriate authority. Instead, the command's priority is not changed, and the system starts the command as it normally would.

The nice value is used by the system to calculate the current priority of a running process. Use the **ps** command with the **-l** flag to view a command's nice value. The nice value appears under the **NI** heading in the **ps** command output.

Note: The **cs** command contains a built-in command named **nice**. The **/usr/bin/nice** command and the **cs** command's **nice** command do not necessarily work the same way. For information on the **cs** command's **nice** command, see the **cs** command.

Flags

Item	Description
<code>-Increment</code>	Increments a command's priority up or down. You can specify a positive or negative number. Positive increment values reduce priority. Negative increment values increase priority. Only users with root authority can specify a negative increment. If you specify an increment value that would cause the nice value to exceed the range of -20 to 19, the nice value is set to the value of the limit that was exceeded. This flag is equivalent to the <code>-n Increment</code> flag.
<code>-n Increment</code>	This flag is equivalent to the <code>-Increment</code> flag.

Exit Status

If the command specified by the *Command* parameter is started, the exit status of the **nice** command is the exit status of the command specified by the *Command* parameter. Otherwise, the **nice** command exits with one of the following values:

Item	Description
1-125	An error occurred in the nice command.
126	The command specified by the <i>Command</i> parameter was found but could not be invoked.
127	The command specified by the <i>Command</i> parameter could not be found.

Examples

1. To specify a very low priority, enter:

```
nice -n 15 cc -c *.c &
```

This example runs the **cc** command in the background at a lower priority than the default priority set by the **nice** command.

2. To specify a very high priority, enter:

```
nice --10 wall <<end  
System shutdown in 2 minutes!  
end
```

This example runs the **wall** command at a higher priority than all user processes, which slows down everything else running on the system. The `<<end` and `end` portions of the example define a *here document*, which uses the text entered before the end line as standard input for the command.

Note: If you do not have root user authority when you run this command, the **wall** command runs at the normal priority.

3. To run a command at low priority, enter:

```
nice cc -c *.c
```

This example runs the **cc** command at low priority.

Note: This does not run the command in the background. The workstation is not available for doing other things.

4. To run a low-priority command in the background, enter:

```
nice cc -c *.c &
```

This example runs the **cc** command at low priority in the background. The workstation is free to run other commands while the **cc** command is running. Refer to the **Shells** in *Operating system and device management* for more information on background (asynchronous) processing.

Files

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

nim Command

Purpose

Performs operations on Network Installation Management (NIM) objects.

Syntax

```
nim { -o Operation } [ -F ] [ -t Type ] [ -a Attribute=Value . . . ] {ObjectName}
```

Description

The **nim** command performs an operation on a NIM object. The type of operation that is performed is dependent on the type of object that is specified by the *ObjectName* parameter. Possible operations include initializing environments and managing resources. You can use the **lsnim** command to display the list of supported operations.

Flags

Item	Description
-a <i>Attribute = Value . . .</i>	Assigns the specified value to the specified attribute. Use the lsnim -q Operation -t Type command to get a list of valid attributes for a specific operation.
-F	Overrides some safety checks.

Item

-o *Operation*

Description

Specifies an operation to perform on a NIM object. The possible operations are:

activate

Starts a managed system.

allocate

Allocates a resource for use.

alt_disk_install

Performs an alternate disk installation.

alt_disk_mig

Creates a copy of **rootvg** to a free disk (or disks) and simultaneously upgrades it to a new version or release level of AIX.

bos_inst

Performs a BOS installation.

change

Changes an object's attributes.

check

Checks the status of a NIM object.

chwpar

Changes the characteristics of managed workload partitions.

create

Creates an instance of a managed system.

cust

Performs software customization.

deactivate

Stops a managed system.

deallocate

Deallocates a resource.

define

Defines an object.

destroy

Removes an instance of a managed system.

diag

Enables a system to boot a diagnostic image.

dkls_init

Initializes a diskless environment of a system.

dtls_init

Initializes a dataless environment of a system.

fix_query

Lists the fix information for an APAR or keyword.

linux_inst

Installs the Linux operating system on stand-alone clients.

lppchk

Verifies installed filesets on NIM systems and SPOTs.

lppmgr

Eliminates unnecessary software images in an **lpp_source**.

lspp

Lists licensed program information about an object.

lswpar

Shows the characteristics of managed workload partitions.

maint

Performs software maintenance.

maint_boot

Enables a system to boot in maintenance mode.

reboot

Reboots a NIM client system.

Item

-o *Operation* (Continued)

Description**remove**

Removes an object.

reset

Resets an object's NIM state.

restvg

Performs a **restvg** operation.

select

Includes and excludes group members from operations that are performed on the group.

showlog

Displays a NIM client's installation, boot or customization log, or a SPOT's installation log from the NIM master.

showres

Displays the contents of a NIM resource.

sync

Synchronizes the NIM database with an alternate master.

sync_roots

Synchronizes root directories for diskless and dataless clients for a specific Shared Product Object Tree (SPOT).

syncwpar

Synchronizes the managed workload partition software with the managing system.

takeover

Allows a machine that is configured as an **alternate_master** to take control of the NIM environment.

unconfig

Unconfigures the NIM master fileset.

update

Adds software to an **lpp_source** or removes software from an **lpp_source**.

updateios

Performs software customization and maintenance on a virtual input-output server (VIOS) management server that is of the **vios** or **ivm** type.

Use the **lsnim -POt Type** command to get a list of the valid operations for a specific type.

Item	Description
-t Type	<p>Specifies the type of the NIM object for define operations. The possible types are:</p> <p>resource types:</p> <p>adapter_def Directory containing secondary adapter definition files.</p> <p>boot An internally managed NIM resource that is used to indicate that a boot image is allocated to a client.</p> <p>bosinst_data Configure file that is used during base system installation.</p> <p>devexports Device exports the file for workload partitions.</p> <p>dump Parent directory for client dump files.</p> <p>exclude_files Contains files to be excluded from a mksysb image.</p> <p>fb_script Executable script that is run during the first reboot of a machine.</p> <p>fix_bundle Fix (keyword) input file for the cust or fix_query operation.</p> <p>home Parent directory for client /home directories.</p> <p>image_data Configure file that is used during base system installation.</p> <p>installp_bundle Installp bundle file.</p> <p>ios_mksysb Represents a backup image that is taken from a VIOS management server that is of the vios or ivm type.</p> <p>linux_source Represents the Linux installation media.</p> <p>log Captures log data during a network installation.</p> <p>lpp_source Source device for optional product images.</p> <p>mksysb mksysb image.</p> <p>nas_filer A network-attached storage (NAS) device.</p> <p>nim_script An internally managed NIM resource that is used to indicate that NIM must run a script as a part of a NIM operation.</p> <p>paging Parent directory for the paging files of the client.</p> <p>root Parent directory for client / (root) directories.</p> <p>resolv_conf Name server configuration file.</p> <p>savevg A savevg image.</p> <p>savewpar Workload partition backup image.</p> <p>script Executable file that is run on a client.</p> <p>secattr Security attributes file for workload partitions.</p>

Item	Description
-t <i>Type</i> (Continued)	Specifies the type of the NIM object for define operations. The possible types are:
	shared_home /home directory that is shared by clients.
	shared_root / (root) directory that is shared by clients
	spot Shared Product Object Tree (SPOT) - equivalent to /usr file system.
	tmp Parent directory for client /tmp directories.
	vg_data Configuration file that is used during volume group restoration.
	wpar_spec Specification file for creating workload partitions.
	machine types:
	alternate_master A system that is reserved as a backup in case the primary NIM master ceases to function properly.
	diskless All file systems and resources remote.
	dataless Local paging, dump; remote ./usr ; others remote or local.
	standalone Local file systems and resources.
	master System that controls the NIM environment.
	wpar Workload partition hosted by the managing system.
	management types:
	bcmm A blade management module hardware.
	cec A central electronic complex hardware.
	hmc A Hardware Management Console system.
	ivm An integrated virtual management system.
	vios A Virtual I/O Server.
	network types:
	tok Token-Ring network.
	ent Ethernet network.
	fddi FDDI network.
	atm ATM network.
	generic Other TCP/IP networks.
	hfi Host Fabric Interface (HFI) network.
	group types:
	mac_group Group of machines.
	res_group Group of resources.

Security

Access Control: You must have root authority to run the **nim** 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

The following examples are grouped by operation.

activate

1. To start the managed `wpar1` workload partition, type:

```
nim -o activate wpar1
```

2. To start the managed `wpar1` workload partition with additional **startwpar** command flags with verbose output, type:

```
nim -o activate -a cmd_flags="-v" wpar1
```

allocate

1. To allocate resources to a diskless workstation with the name `syzygy` and SPOT attribute value of `spot1`, type:

```
nim -o allocate -a spot=spot1 syzygy
```

2. To perform a base system installation on the system that is named `krakatoa`, resources must be allocated initially by entering:

```
nim -o allocate -a spot=myspot -a lpp_source=images krakatoa
```

The NIM environment can be initialized to support the installation by performing the **bos_inst** operation, type:

```
nim -o bos_inst krakatoa
```

3. To install the software product, `adt`, into a standalone system, `stand1`, given that the installable option, `adt`, in the **lpp_source**, `images`, type:

```
nim -o allocate -a lpp_source=images stand1
```

Then type:

```
nim -o cust -a filesets="adt" stand1
```

4. To install software products into a standalone system, `stand1`, such that the image for the installable option, `adt`, in the **lpp_source**, `images`, and the **installp_bundle**, `bundle1`, contains the name of the installable option, type:

```
nim -o allocate -a lpp_source=images \  
-a installp_bundle=bundle1 stand1
```

Then type:

```
nim -o cust stand1
```

5. To automatically configure a machine with name resolution services after a BOS installation, create the `/exports/resolv.conf` file, with contents similar to the following:

```
nameserver      129.35.143.253  
nameserver      9.3.199.2  
domain          austin.ibm.com
```

then type:

```
nim -o define -t resolv_conf -a location=/exports/resolv.conf \  
-a server=master rconf1
```

Prior to issuing the **bos_inst** operation, allocate this resource with other required and optional resources by typing:

```
nim -o allocate -a spot=spot1 -a lpp_source=images1 \  
-a bosinst_data=bid1 -a resolv_conf=rconf1 client1
```

6. To allocate all resources applicable to standalone machines from the NIM resource group `res_grp1`, to the machine `mac1`, type:

```
nim -o allocate -a group=res_grp1 mac1
```

alt_disk_install

1. To install a **mksysb** resource `all_devices_msysb` to client `roundrock`, on `hdisk4` and `hdisk5`, using the **image_data** resource `image_data_shrink`, with debug turned on, type:

```
nim -o alt_disk_install -a source=mksysb\  
-a image_data=image_data_shrink\  
-a debug=yes\  
-a disk='hdisk4 hdisk5' roundrock
```

2. To clone a **rootvg** on client `austin` to `hdisk2`, but only run phase1 and phase2 (leaving the **/alt_inst** file systems mounted), type:

```
nim -o alt_disk_install -a source=rootvg\  
-a disk='hdisk2'\  
-a phase=12 austin
```

bos_inst

1. To install the machine `blowfish`, using the resources `spot1`, `images1`, `bosinst_data1`, and `rconf1`, first allocate the resources by typing:

```
nim -o allocate -a spot=spot1 -a lpp_source=images1 \  
-a bosinst_data=bosinst_data1 -a resolv_conf=rconf1 blowfish
```

Then, perform the BOS installation by typing:

```
nim -o bos_inst blowfish
```

2. To install the machine `blowfish` while allocating the resources `spot1`, `images1`, `bosinst_data1`, and `rconf1` automatically when the **bos_inst** operation starts, type:

```
nim -o bos_inst -a spot=spot1 -a lpp_source=images1 \  
-a bosinst_data=bosinst_data1 -a resolv_conf=rconf1 blowfish
```

3. To use the default resources when installing the machine `mac1`, type:

```
nim -o bos_inst mac1
```

4. To install a machine, `deadfish`, with `spot1` and `lpp_source1` and use an **adapter_def** resource, `adapter_def1`, to configure secondary adapters, type:

```
nim -o bos_inst -a spot=spot1 -a lpp_source=lpp_source1 \  
-a adapter_def=adapter_def1 deadfish
```

5. To install the machine `blowfish` and accept software license agreements, type:

```
nim -o bos_inst -a spot=spot1 -a lpp_source=images1 \  
-a accept_licenses=yes -a resolv_conf=rconf1 blowfish
```

change

1. Machines on the BLDG905 network use the gateway905 gateway to reach the OZ network. Machines on the OZ network use the gatewayOZ gateway to reach the BLDG905 network. To add a route between two networks named BLDG905 and OZ, type:

```
nim -o change -a routing1="OZ gateway905 gatewayOZ" BLDG905
```

2. The adapter that is identified by the host name sailfish2.austin.ibm.com is attached to a token ring network. To define a secondary interface for this adapter on the NIM master and instructing NIM to locate the NIM network representing the attached ethernet network and, if not found, have NIM define a NIM network with subnetmask 255.255.255.128, type:

```
nim -o change -a if2="find_net sailfish2.austin.ibm.com 0" \  
-a net_definition="tok 255.255.255.128" -a ring_speed2=16 master
```

Note: A default name is generated for the network, and no routing information is specified for the new network.

3. To define default routes for the networks net1 and net2 that use default gateways gw1 and gw2 respectively, type the following two commands:

```
nim -o change -a routing1="default gw1" net1  
nim -o change -a routing1="default gw2" net2
```

4. To designate the resources that are defined by the resource group res_grp1 as the set of resources that are always allocated by default during any operation in which these resources are applicable, type:

```
nim -o change -a default_res=res_grp1 master
```

check

1. To have NIM check on the usability of a SPOT named myspot, type:

```
nim -o check myspot
```

2. To check the status of an **lpp_source** named images, type:

```
nim -o check images
```

chwpar

To add rset rs/cpus23 to the resource control attributes for the wpar1 workload partition, type:

```
nim -o chwpar -a cmd_flags="-R rset=rs/cpu23" wpar1
```

create

1. To create the wpar1 workload partition with host name and specification file resource basic_wpar, type:

```
nim -o create -a wpar_spec=basic_wpar wpar1
```

2. To create the wpar1 workload partition with the wpar-specification file resource wpar1_spec, type:

```
nim -o create -a wpar_spec=wpar1_spec wpar1
```

3. To create the wpar1 workload partition from the **savewpar** backup image resource wpar1_backup, type:

```
nim -o create -a savewpar=wpar_backup wpar1
```

cust

1. To install a software product into a spot, `spot1`, such that the image for the installable option, `adt`, resides in the **lpp_source**, `images`, type:

```
nim -o cust -a lpp_source=images -a filesets=adt spot1
```

2. To install a software product into a spot, `spot1`, such that the image for the installable option, `adt`, resides in the **lpp_source**, `images`, and the **installp_bundle**, `bundle1`, contains the name of the installable option, type:

```
nim -o cust -a lpp_source=images -a installp_bundle=bundle1 spot1
```

3. To install a software product into a spot, `spot1`, such that the image for the installable option, `adt`, resides on a tape that is in the tape drive that is local to the machine where the spot resides, type:

```
nim -o cust -a lpp_source=/dev/rmt0 -a filesets=adt spot1
```

4. To install a software product into a spot, `spot1`, such that the image for the installable option, `adt`, resides on a tape that is in the tape drive that is local to the machine where the spot resides, type:

```
nim -o cust -a lpp_source=/dev/rmt0 -a filesets=adt spot1
```

5. To install all fileset updates associated with APAR `IX12345`, residing on the tape `/dev/rmt0` into `spot1` and any diskless or dataless clients to which `spot1` is currently allocated, type:

```
nim -F -o cust -afixes=IX12345 -a lpp_source=/dev/rmt0 spot1
```

6. To update all software installed on the client `Standalone1`, with the latest updates in the **lpp_source** named `updt_images`, type:

```
nim -o allocate -a lpp_source=updt_images Standalone1  
nim -o cust -afixes=update_all Standalone1
```

7. To install the machine `catfish` with the contents of the **installp_bundle** `bundle1`, first allocate the resources by typing:

```
nim -o allocate -a installp_bundle=bundle1 \  
-a lpp_source=images1 catfish
```

Then, perform the `cust` operation by typing:

```
nim -o cust catfish
```

8. To update all software that is installed on the client `Standalone1`, with the latest updates in the **lpp_source** named `updt_images`, type:

```
nim -o cust -a lpp_source=updt_images -a fixes=update_all \  
Standalone1
```

9. To install the machine `catfish` with the contents of the **installp_bundle** `bundle1`, while allocating this resource and the **lpp_source** `images1` when the **cust** operation runs, type:

```
nim -o cust -a installp_bundle=bundle1 -a lpp_source=images1 \  
catfish
```

10. To configure secondary adapters on a client machine, `deadfish`, by using the secondary adapter configuration file in the **adaper_def** resource, `adapter_def1`, type:

```
nim -o cust -a adapter_def=adapter_def1 deadfish
```

deactivate

1. To stop the managed `wpar1` workload partition, type:

```
nim -o deactivate wpar1
```

2. To force the stop of the managed wpar1 workload partition, type:

```
nim -Fo deactivate wpar1
```

3. To stop the managed wpar1 workload partition with more **stopwpar** command flags to halt after 85 seconds, type:

```
nim -o deactivate -a cmd_flags="-t 85" wpar1
```

deallocate

To deallocate an **lpp_source** named images from the standalone machine client1, type:

```
nim -o deallocate -a lpp_source=images client1
```

define

1. To define a resource that is a directory that contains installable images that is on the server altoid and has a path name of /usr/sys/inst.images, and name that resource images, type:

```
nim -o define -t lpp_source -a server=altoid \  
-a location=/usr/sys/inst.images images
```

2. To create a new SPOT resource named myspot on the NIM master in the /export/exec directory, by using an **lpp_source** named image, type:

```
nim -o define -t spot -a server=master -a location=/export/exec \  
-a source=images myspot
```

3. To define a network object named BLDG905, with a subnetmask of 255.255.240.0 and an address of 129.35.129.0, type:

```
nim -o define -t tok -a snm=255.255.240.0 \  
-a net_addr=129.35.129.0 BLDG905
```

4. To define a **mksysb** resource, mksysb1, from an existing mksysb image that is located in /resources/mksysb.image on the master, type:

```
nim -o define -t mksysb -a server=master \  
-a location=/resources/mksysb.image mksysb1
```

5. To define a NIM network named ATMnet with a subnet mask of 255.255.240 and an address of 129.35.101.0 to represent an ATM network, use the generic network type as follows:

```
nim -o define -t generic -a snm=255.255.240.0 \  
-a net_addr=129.35.101.0 ATMnet
```

6. To define a machine group named Disklsmacs1 with members that are NIM diskless machines named diskls1, diskls2, and diskls3, type:

```
nim -o define -t mac_group -a add_member=diskls1 \  
-a add_member=diskls2 -a add_member=diskls3 Disklsmacs1
```

7. To define a resource group named DisklsRes1 with resources spot1, root1, dump1, paging1, home1, tmp1, type:

```
nim -o define -t res_group -a spot=spot1 -a root=root1 \  
-a dump=dump1 -a paging=paging1 -a home=home1 -a tmp=tmp1 \  
DisklsRes1
```

8. To display the space that is required to define a **mksysb** resource, mksysb2, and create a mksysb image of the client, client1, during the resource definition where the image is located in /resources/mksysb.image on the master, type:

Note: This action shows the space that is required for the operation, **mksysb**, or resource creation does NOT take place.


```
nim -o define -t mksysb -a server=master \
-a location=/resources/mksysb.image -a source=client1 \
-a mk_image=yes -a size_preview=yes mksysb2
```

9. To define a **mksysb** resource, `mksysb2`, and create the `mksysb` image of the client, `client1`, during the resource definition where the image is in `/resources/mksysb.image` on the master, type:

```
nim -o define -t mksysb -a server=master \
-a location=/resources/mksysb.image -a source=client1 \
-a mk_image=yes mksysb2
```

10. To define a **mksysb** resource, `mksysb2`, and create a `mksysb` image of the client, `client1`, during the resource definition where the `mksysb` flags used to create the image are **-em**, and the image is in `/resources/mksysb.image` on the master, type:

```
nim -o define -t mksysb -a server=master \
-a location=/resources/mksysb.image -a source=client1 \
-a mk_image=yes -a mksysb_flags=em mksysb2
```

11. To define an **exclude_files** resource, `exclude_file1`, located in `/resources/mksysb.image` on the master, type:

```
nim -o define -t exclude_files -a server=master \
-a location=/resources/exclude_file1 exclude_file1
```

12. A machine that is called `redfish`, hostname `redfish_t.lab.austin.ibm.com`, has its primary interface that is attached to a token-ring network with ring speed of 16 Megabits. To define `redfish` as a standalone machine in the NIM environment and instructing NIM to locate the name of the network that the machine's primary interface is attached, type:

```
nim -o define -t standalone -a if1="find_net \
redfish_t.lab.austin.ibm.com 0" -a ring_speed1=16 redfish
```

13. A machine that is called `bluefish`, hostname is `bluefish_e.lab.austin.ibm.com`, has its primary interface that is attached to an ethernet network with **cable_type** of **bnc**. To define `bluefish` as a diskless machine in the NIM environment and instructing NIM to locate the name of the network that the machine's primary interface is attached, and if not found, have NIM define a NIM network with the name `ent_net`, subnetmask of `255.255.255.128` and default route by using the gateway with hostname `lab_gate`, type:

```
nim -o define -t diskless -a if1="find_net \
bluefish_e.lab.austin.ibm.com 0" -a net_definition="ent \
255.255.255.128 lab_gate 0 ent_net" -a cable_type=bnc bluefish
```

Note: Specify 0 in place of the master gateway in the **net_definition** attribute if a default route for the master exists, otherwise you must specify the master gateway.

14. To define the **/export/nim/adapters** directory as an **adapter_def** resource, `adapter_def1`, on the master, type:

```
nim -o define -t adapter_def -a server=master \
-a location=/export/nim/adapters adapter_def1
```

To populate the **adapter_def** resource with secondary adapter configuration files, run the **nimadapters** command.

15. To display the space that is required to define a **savevg** resource, `savevg2`, and create a **savevg** image of the client, `client1`, during the resource definition where the image is in `/export/nim/savevg` on the master and the **volume_group** to to backup is `myvg`, type:

```
nim -o define -t savevg -a server=master \
-a location=/export/nim/savevg/savevg2 -a source=client1 \
-a mk_image=yes -a size_preview=yes -a volume_group=myvg savevg2
```

Note: This action shows the space that is required for the operation. **savevg** or resource creation does not take place.

16. To define a **savevg** resource, `savevg2`, and create the **savevg** image of the client, `client1`, during the resource definition where the image is in `/export/nim/savevg` on the master and the **volume_group** to backup is `myvg`, type:

```
nim -o define -t savevg -a server=master \  
-a location=/export/nim/savevg -a source=client1 \  
-a mk_image=yes -a volume_group=myvg savevg2
```

17. To define a **savevg** resource, `savevg2`, and create a **savevg** image of the client, `client1`, during the resource definition where the **savevg** flags used to create the image are **-em**, and the image is in `/export/nim/savevg` on the master, type:

```
nim -o define -t savevg -a server=master \  
-a location=/export/nim/savevg -a source=client1 \  
-a mk_image=yes -a volume_group=myvg -a savevg_flags=em savevg2
```

18. To define a **vg_data** resource, `my_vg_data`, on the master at the location `/export/nim`, type:

```
nim -o define -t vg_data -a server=master -a location=/export/nim/my_vg_data my_vg_data
```

19. To define the `wpar1` workload partition that is managed by the `yogi` managing standalone machine with `wpar1` as both the host name and the name of the workload partition on the managing system, type:

```
nim -o define -t wpar -a mgmt_profile1="yogi wpar1" -a if1="find_net wpar1 0" wpar1
```

20. To define a **savewpar** resource named `wpar1backup` and create the **savewpar** image of the `yogi` workload partition on the `sterling` server, type:

```
nim -o define -t savewpar \  
-a server=sterling -a location=/resources/wpar1.image \  
-a source=wpar1 -a mk_image=yes wpar1backup
```

21. To define a **savewpar** resource named `wpar1backup` and create the **savewpar** image of the `yogi` workload partition on the `sterling` server, excluding file patterns in the **exclude_files** resource `wparexclude`, and passing the flag to the **savewpar** resource to exclude files and creates a **image.data** file, type:

```
nim -o define -t savewpar \  
-a server=sterling -a location=/resources/wpar1.image -a source=wpar1 \  
-a exclude_files=wparexclude -a cmd_flags="-ei" mk_image=yes wpar1backup
```

22. To define a **ios_mksysb** resource such as **ios_mksysb1**, and create the **ios_mksysb** image of the **vios** client as **vios1**, during the resource definition where the image is located in `/export/nim/ios_mksysb` on the master, type:

```
nim -o define -t ios_mksysb -a server=master \  
-a location=/export/nim/ios_mksysb -a source=vios1 \  
-a mk_image=yes ios_mksysb1
```

destroy

1. To remove the managed `wpar1` workload partition from its managing system, type:

```
nim -o destroy wpar1
```

2. To force the removal of the managed `wpar1` workload partition, type:

```
nim -Fo destroy wpar1
```

dkls_init

1. To initialize the environment for a diskless workstation with the name of `syzygy`, by using the resources `spot1`, `root1`, `dump1`, and `paging1`, you must allocate the resources by typing:

```
nim -o allocate -a spot=spot1 -a root=root1 -a dump=dump1 \  
-a paging=paging1 syzygy
```

Then initialize the resources for the client machine by typing:

```
nim -o dkls_init syzygy
```

2. To initialize the environment for a diskless workstation with the name of syzygy, type:

```
nim -o dkls_init syzygy
```

3. To exclude the member named diskls2 from operations on the machine group DisklsMac1, and then initialize the remaining members while allocating the diskless resources defined by the resource group named DisklsRes1, type the following two commands:

```
nim -o select -a exclude=diskls2 DisklsMac1  
nim -o dkls_init -a group=DisklsRes1 DisklsMac1
```

4. To initialize the group of diskless machines that are defined by the machine group dtgrp1, while allocating the required and optional resources defined by the resource group dk_resgrp1, when the **dkls_init** operation runs, type:

```
nim -o dkls_init -a group=dtgrp1 dk_resgrp1
```

dtls_init

1. To initialize the environment for a dataless workstation with the name of syzygy, using the resources spot1, root1, and dump1, first allocate the resources by typing:

```
nim -o allocate -a spot=spot1 -a root=root1 -a dump=dump1 syzygy
```

Then initialize the resources for the client machine by typing:

```
nim -o dtls_init syzygy
```

2. To initialize the environment for a dataless workstation with the name of syzygy, type:

```
nim -o dtls_init syzygy
```

3. To exclude the member named dataless1 from operations on the machine group Data1sMac1, and then initialize the remaining members while allocating the dataless resources defined by the resource group named Data1sRes1, type the following two commands:

```
nim -o select -a exclude=dataless2 Data1sMac1  
nim -o dtls_init -a group=Data1sMac1 Data1sRes1
```

4. To initialize the group of dataless machines defined by the machine group Data1sMac1, while allocating the required and optional resources defined by the resource group Data1sRes1, when the **dtls_init** operation runs, type:

```
nim -o dtls_init -a group=Data1sMac1 Data1sRes1
```

fix_query

To list information about fixes installed on client Standalone1 for 20 APAR numbers, create the file /tmp/apar.list with one APAR number per line, as shown:

```
IX123435  
IX54321  
IX99999  
...
```

then type:

```
nim -o define -t fix_bundle -allocation=/tmp/apar.list \
    -aserver=master fix_bun
nim -o allocate -a fix_bundle=fix_bun Standalone1
nim -o fix_query Standalone1
```

lppchk

1. To check fileset version and requisite consistency on the SPOT spot1, type:

```
nim -o lppchk spot1
```

2. To verify the file checksums for all packages beginning with the name bos on NIM targets in the group of standalone machines macgrp1, and displaying detailed error information and updating the software database to match the actual file checksum when inconsistencies are found, type:

```
nim -o lppchk -a lppchk_flags='-c -m3 -u' \
-a filesets='bos*' macgrp1
```

Because the **lppchk** operation runs in the background on group members by default, to view the output from the **lppchk** operation type:

```
nim -o showlog -a log_type=lppchk macgrp1
```

lppmgr

1. To list the names of duplicate base level filesets which should be removed from lpp_source1 with space usage information, type:

```
nim -o lppmgr -a lppmgr_flags="-lsb" lpp_source1
```

2. To remove duplicate base and update filesets and superseded updates from lpp_source1, type:

```
nim -o lppmgr -a lppmgr_flags="-rbux" lpp_source1
```

3. To remove all non-SIMAGES (filesets that are not required for a bos install) from lpp_source1, type:

```
nim -o lppmgr -a lppmgr_flags="-rX" lpp_source1
```

4. To remove all language support except 'C' from lpp_source1, type:

```
nim -o lppmgr -a lppmgr_flags="-r -k C" lpp_source1
```

lswpar

1. To list the characteristics of the managed wpar1 workload partition, type:

```
nim -o lswpar wpar1
```

2. To list the network characteristics of the managed wpar1 workload partition, type:

```
nim -o lswpar -a cmd_flags="-N" wpar1
```

3. To list the general characteristics of the workload partitions managed by the global1 standalone system, type:

```
nim -o lswpar -a cmd_flags="-G" global1
```

maint

1. To cleanup from an interrupted software installation on a spot, spot1, type:

```
nim -o maint -a installp_flags="-C" spot1
```

2. From the master, to clean up from an interrupted software installation on a standalone machine, stand1, type:

```
nim -o maint -a installp_flags="-C" stand1
```

maint_boot

To enable the NIM standalone client, `stand1`, to boot in maintenance mode, type:

```
nim -o maint_boot stand1
```

This sets up the maintenance boot operation, but you must initiate the network boot locally from `stand1`.

remove

To remove a resource named `dump_files`, type:

```
nim -o remove dump_files
```

showlog

To view the boot logs of the machines that are defined by the group `Disk1sMac1`, type:

```
nim -o showlog -a log_type=boot Disk1sMac1
```

showres

1. To show the contents of the configure script `script1`, type:

```
nim -o showres script1
```

2. To show the contents of the `bosinst.data` resource `bosinst_data1`, type:

```
nim -o showres bosinst_data1
```

3. To list all the filesets in the `lpp_source` `lpp_source1`, type:

```
nim -o showres lpp_source1
```

4. To list all the filesets in the `lpp_source` `lpp_source1` relative to what is installed on the machine `machine1`, type:

```
nim -o showres -a reference=machine1 lpp_source1
```

5. To list all the problems that are fixed by software on the `lpp_source` `lpp_source1`, use:

```
nim -o showres -a instfix_flags="T" lpp_source1
```

6. To show the contents of the secondary adapter configuration file in the **adapter_def** resource, `adapter_def1`, for client, `deadfish`, type:

```
nim -o showres -a client=deadfish adapter_def1
```

7. To show the contents of every secondary adapter configuration file in the **adapter_def** resource, `adapter_def1`, type:

```
nim -o showres adapter_def1
```

8. To show the contents of the **savevg** resource, `savevg1`, type:

```
nim -o showres savevg1
```

syncwpar

1. To synchronize the software of the managed `wpar1` workload partition with its managing system, type:

```
nim -o syncwpar wpar1
```

2. To synchronize the software of all the workload partitions managed by the `global1` standalone system, type:

```
nim -o syncwpar -a cmd_flags="-A" global1
```

update

1. To add all the filesets on `/dev/cd0` to `lpp_source1`, type:

```
nim -o update -a packages=all -a source=/dev/cd0 lpp_source1
```

2. To add the `bos.games 7.1.0.0` and `bos.terminfo` filesets to `lpp_source1`, type:

```
nim -o update -a packages="bos.games 7.1.0.0 bos.terminfo" \  
-a source=/dev/cd0 lpp_source1
```

3. To remove `bos.games` from `lpp_source1`, type:

```
nim -o update -a rm_images=yes -a packages="bos.games" lpp_source1
```

4. To recover the missing SIMAGES for `lpp_source1` from the AIX Installation CD, type:

```
nim -o update -a recover=yes -a source=/dev/cd0 lpp_source1
```

updateios

1. To install fixes or to update VIOS with the `vioserver1` NIM object name to the latest maintenance level, type:

```
nim -o updateios -a lpp_source=lpp_source1 -a preview=no vioserver1
```

The updates are stored in `lpp_source` and `lpp_source1` files.

Note: The `updateios` operation runs a preview during installation. Running the `updateios` operation from NIM runs a preview unless the preview flag is set to no. During the installation, you must run a preview when you use the `updateios` operation with `updateios_flags=-install`. With the preview, you can check whether the preview installation is running accurately before you proceed with the VIOS update.

2. To reject fixes for a VIOS with the `vioserver1` NIM object name, type:

```
nim -o updateios -a updateios_flags=-reject vioserver1
```

3. To clean up partially installed updates for a VIOS with the `vioserver1` NIM object name, type:

```
nim -o updateios -a updateios_flags=-cleanup vioserver1
```

4. To commit updates for a VIOS with the `vioserver1` NIM object name, type:

```
nim -o updateios -a updateios_flags=-commit vioserver1
```

5. To remove a specific update such as `update1` for a VIOS with the `vioserver1` NIM object name, type:

```
nim -o updateios -a updateios_flags=-remove-a filesets="update1" vioserver1
```

6. To remove updates for a VIOS with the `vioserver1` NIM object name by using an `installp_bundle bundle1`, where `bundle1` contains the updates to be removed, type:

```
nim -o updateios -a updateios_flags=remove -a installp_bundle=bundle1 vioserver1
```

Files

Item	Description
<code>/etc/niminfo</code>	Contains variables that are used by NIM.

nim_clients_setup Command

Purpose

Define clients and initialize BOS install operation on NIM client objects.

Syntax

```
nim_clients_setup [ -m mksysb_resource] [ -n] [ -c] [ -r] [ -v] client_object(s)
```

Description

The `nim_clients_setup` command defines new client objects and initializes the BOS install operation for clients in the NIM environment by performing the following tasks:

- Exports the environment variable `NIM_LICENSE_ACCEPT=yes`.
 - Used for accepting software license agreement during network install.
- Adds variable entry `NSORDER=local,bind` in `/etc/environment`.
 - Necessary for name resolution when hosts only exist in `/etc/host`.
- Defines client objects using `client.defs` file (if `-c` flag specified).
 - User must edit stanzas in `/export/nim/client.defs` file prior to using `nim_clients_setup`.
- Prepares client objects for install.
 - If `-c` flag is used, defined clients are initialized for install.
 - If client objects are given, specified clients are initialized for install.
 - If `-c` or client objects are omitted, all existing NIM clients are initialized for install.
- Resources contained in the group name `basic_res_grp` are used as resources during the BOS install operation.

Note: The `basic_res_grp` resource group is populated with resources created during `nim_master_setup` command execution. If this group is not present, it must be defined with NIM install resources prior to using the `nim_clients_setup` command.

Flags

Item	Description
<code>-m</code> <i>mksysb_resource</i>	Specifies an alternate backup image to restore during BOS install. The value for <i>mksysb_resource</i> may specify a NIM object name or absolute path location used for defining a new <code>mksysb</code> resource. By default, the <code>mksysb</code> resource is assigned from the <code>basic_res_grp</code> NIM resource group.
<code>-n</code>	Enables native (<code>rte</code>) install and ignores restoring backup image (<code>mksysb</code>) during BOS install. By default, <code>mksysb restore</code> is performed during BOS install.
<code>-c</code>	Defines client objects from the <code>client.defs</code> file. The <code>/export/nim/client.defs</code> file must exist and have valid client definition information. The <code>client.defs</code> file is created during <code>nim_master_setup</code> command execution. If the file is not present, a sample <code>client.defs</code> file may be copied from <code>/usr/samples/nim/client.defs</code> and edited by the user.
<code>-r</code>	Reboots client objects after initiating BOS install operation. By default, clients are not rebooted. Resources are assigned for install and clients may be rebooted when desired.
<code>-v</code>	Enables verbose debug output during command execution.

Security

Access Control: You must have root authority to run the `nim_clients_setup` command.

Location

`/usr/sbin/nim_clients_setup`

Examples

1. To define client objects from `/export/nim/client.defs` file, initialize the newly defined clients for BOS install using resources from the `basic_res_grp` resource group, and reboot the clients to begin install, type:

```
nim_clients_setup -c -r
```

2. To initialize clients `client1` and `client2` for BOS install, using the backup file `/export/resource/NIM/530mach.sysb` as the restore image, type:

```
nim_clients_setup -m /export/resource/NIM/530mach.sysb \ client1 client2
```

3. To initialize all clients in the NIM environment for native (`rte`) BOS install using resources from the `basic_res_grp` resource group, type:

```
nim_clients_setup -n
```

Files

Item	Description
<code>/etc/niminfo</code>	Contains variables used by NIM.

nim_master_recover Command

Purpose

Restores a backup of the Network Installation Management (NIM) database to a different machine and updates the database to reflect this change.

Syntax

```
nim_master_recover [ -f mstr_fileset_dir ]  
[ -n nimdef_file ]  
[ -r nimdb_file ]  
[ -i mstr_interface ]  
[ -D ][ -R ][ -S ][ -p ][ -s ][ -u ][ -v ]  
[ -N mstr_net_info [-t net_def ] ]
```

Description

The `nim_master_recover` command can restore and update the NIM database from a backup tar file. To backup the NIM database on the old master, run the `smit nim_backup_db` command. This creates a tar file named `/usr/objrepos/nimdb.backup` by default. Once the `nimdb.backup` is copied to the new master, pass the `-r` flag with the full path to the file. If the path to the tar file is `/usr/objrepos/nimdb.backup`, then pass `-r /usr/objrepos/nimdb.backup` to the `nim_master_recover` script.

The script updates the master definition in the NIM database based on the master's primary network interface. The `-i` flag specifies the primary interface to use for the master. To use `en0`, pass `-i en0` to the `nim_master_recover` script.

Note: A restored NIM database may be incorrect if you restore from a database that has network definitions containing static routes. The `nim_master_recover` command removes all the interfaces in the old master definition before adding the primary interface for the new master. Check that the routing information is correct after running the `nim_master_recover` command, by running `lsnim -lc networks`. If all the NIM network definitions in the restored database contain dynamic routes, then you should not run into this situation.

Along with restoring and updating the NIM database, the script performs several other optional functions. One is to install the `bos.sysmgt.nim.master` fileset if the `-f` flag is passed with the location of the `bos.sysmgt` package. For instance, if the `bos.sysmgt` package is located in the `/export/latest/installp/ppc` directory, then you would pass `-f /export/latest/installp/ppc` to the `nim_master_recover` script.

The script always resets each client. If the `-u` flag is passed, the script attempts to unexport NIM resources that the database states are allocated to clients. Each client stores the hostname of its NIM master in its `/etc/niminfo` file. To update the `niminfo` file on each client, pass the `-s` flag.

Note: Any NIM client that is not running, does not have a network connection, does not allow the new master `rhost` permissions, or does not have at least the `bos.sysmgt.nim.client 5.1.0.10` package, will not have its `niminfo` updated. The `nim_master_recover` script will report any clients which fail to have their `niminfo` files updated.

New clients can be added to the environment by specifying a `nimdef` file with the `-n` flag. Consult the AIX Installation Guide for more information on `nimdef` files.

Finally, the script will check to see if the resources in the NIM database exist. The script deletes resources that don't exist. For example if the new master is unable to communicate with a NIM server, then the resources defined on that server will be removed from the NIM database. Passing the `-R` flag prevents the script from checking resources.

Note: Resources that were defined on the master where the database was backed up, will not be available once the database is restored unless the resources were copied to the new master before running `nim_master_recover`.

All output will be logged to `/var/adm/ras/nim.recover`. Once the script is complete you should verify that no errors were logged.

The `nim_master_recover` command behaves differently when it is called with the `-N` flag. This allows the master to have its hostname, IP address, and NIM network changed in its `if1` attribute. Optionally, a new NIM network may be created if the `-t` flag is provided with the `-N` flag. The command should be run with these flags before the master's network name or address is actually changed so that the NIM environment will work properly once the change actually takes place. When the master's NIM attributes are changed, the command will attempt to update `.rhosts` and `/etc/niminfo` of each standalone client defined in the environment. Any clients for which this attempt fails must have its NIM master information updated manually. Also, after a standalone client has had its NIM master's network name changed, it will not be able to execute any NIM operations until the master is up and running with its new network name.

Flags

Item	Description
<code>-D</code>	Deletes all client definitions from the restored database.
<code>-f directory</code>	Directory containing the <code>bos.sysmgt.nim.master</code> fileset to install.
<code>-i interface</code>	Primary network interface of the machine where you are running the command.
<code>-n nimdef</code>	Optional <code>nimdef</code> file that will be used to define new machines.

Item	Description
-N <i>mstr_net_info</i>	Changes the master's if1 attribute and attempts to update each standalone client defined in the environment with the master's new network information. The <i>mstr_net_info</i> variable consists of the following: "nim_net_name [hostname] [cable_type]"; where <i>hostname</i> and <i>cable_type</i> are optional.
-p	Print the machine states before the script resets the machines.
-r <i>nimdb.backup</i>	The NIM database backup tar file that will be restored.
-R	Do not check the resources to see if each one exists. The default behavior is for the script to check each resource and if it does not exist, remove its definition from the database.
-S	Do not check the SPOT resources. The default behavior is for the script to check every SPOT to ensure it is ready to support an install. For example, the check ensures the boot images are created.
-s	Attempt to update the niminfo file on each client. Any NIM client that is not running, does not have a network connection, does not allow the new master rhost permissions, or does not have at least the <code>bos.sysmgt.nim.client 5.1.0.10</code> package installed, will not have its niminfo updated.
-t <i>net_def</i>	Creates a new NIM network if the master's IP address changes and there is no existing NIM network that could contain the master. This flag is only valid when the -N flag is also specified. The <i>net_def</i> variable consists of the following: "nim_net_name net_type net_addr net_snm default_route"; where <i>net_type</i> can be ent, tok, atm, or fddi.
-u	Unexport all resources that are listed as allocated in the restored database. The default behavior is for the script to delete the allocation from the NIM database without attempting to deallocate the resource.
-v	Enables verbose debug output during command execution.

Location

/usr/sbin/nim_master_recover

Exit Status

Returns zero (0) upon success.

Security

Access Control: You must have root authority to run the `nim_master_recover` 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 recover the NIM master using the `/export/nim/nimdb.backup` file and the primary interface `en0`, type:

```
nim_master_recover -r /usr/objrepos/nimdb.backup -i en0
```

2. To install the `bos.sysmgmt.nim.master` fileset from `/export/lpp_source/installp/ppc` before recovering the NIM master, type:

```
nim_master_recover -f /export/lpp_source/installp/ppc \  
-r /usr/objrepos/nimdb.backup -i en0
```

3. To recover the NIM master without checking if each resource exists and without checking the SPOTs to rebuild boot images, type:

```
nim_master_recover -R -S -r /usr/objrepos/nimdb.backup -i en0
```

4. To recover the NIM master while unexporting any resources that are allocated and printing the state of the clients before each one is reset, type:

```
nim_master_recover -u -p -r /usr/objrepos/nimdb.backup -i en0
```

5. To recover the NIM master and update the `/etc/niminfo` file on each client, type:

```
nim_master_recover -s -r /usr/objrepos/nimdb.backup -i en0
```

6. To recover the NIM master, delete each client from the database, and define new clients from the `nimdef` file `/export/nim/nimdef`, type:

```
nim_master_recover -D -n /export/nim/nimdef -r /usr/objrepos/nimdb.backup -i en0
```

7. To change the master's hostname to `newhost.domain.com` and move it to a different existing NIM network, called `net2`, but preserve the value of the current `cable_type` attribute, type:

```
nim_master_recover -N "net2 newhost.domain.com"
```

8. To change the master's hostname to `newhost.domain.com`, change its `cable_type` to `bnc`, and move it to a new NIM ethernet network called `new_nim_net` whose address is `192.168.1.0`, subnet mask is `255.255.255.0`, and default gateway is `192.168.1.1`, type:

```
nim_master_recover -N "new_nim_net newhost.domain.com bnc" \  
-t "new_nim_net ent 192.168.1.0 255.255.255.0 192.168.1.1"
```

Files

Item	Description
<code>/etc/niminfo</code>	Contains variables used by NIM.
<code>/var/adm/ras/nim.recover</code>	Contains log information from command execution.

nim_master_setup Command

Purpose

Initializes the Network Installation Management (NIM) master fileset, configures the NIM master, and creates the required resources for installation.

Syntax

```
nim_master_setup [ -a [ mk_resource={yes|no} ] [ file_system=fs_name ] [ volume_group=vg_name ]  
[ disk=disk_name ] [ device=device ] ] [ -B ] [ -F ] [ -L ] [ -c ] [ -v ]
```

Description

The `nim_master_setup` command initializes the NIM master fileset and configures the NIM environment. Once initialized, the `nim_master_setup` command configures the NIM environment by performing the following tasks:

- Determines which volume group and file system will contain the NIM resources.
- If necessary, creates the volume group and file system.
- Creates a NIM `mksysb` of the master.
 - Backup image.
- Creates a NIM `lpp_source` resource.
 - Source for product images.
- Creates a NIM `spot` resource.
 - Shared Product Object Tree (SPOT) - equivalent to `/usr` file system.
- Creates a NIM `bosinst_data` resource.
 - `config` file used during BOS installation.
- Creates a NIM `resolv_conf` resource.
 - Name-server configuration file.
- Defines a default resource group for use during install. The default resource group will contain all NIM resources defined during command execution.
- Copies a sample `client.defs` configuration file into the defined NIM file system.
 - Sample file which may be edited for adding clients in the NIM environment.

Flags

Item	Description
-a	Assigns the following <code>attribute=value</code> pairs: <ul style="list-style-type: none">mk_resource={yes no} Specifies if NIM resources should be created. If set to <code>no</code>, NIM resources will not be created during command execution. By default, the value is <code>yes</code>.file_system=fs_name Specifies the absolute path location for creating NIM resources. If <code>fs_name</code> does not exist, a logical volume will be created in the volume group defined from <code>vg_name</code>. By default, <code>fs_name</code> is <code>/export/nim</code>.volume_group=vg_name Specifies the volume group name used for creating new logical volumes. If <code>vg_name</code> does not exist, a volume group will be created using the physical volume (disk) defined from <code>disk_name</code>. By default, <code>vg_name</code> is <code>rootvg</code>.disk=disk_name Specifies the physical volume used when creating the <code>vg_name</code> volume group. If <code>disk_name</code> is not specified, the next available (empty) physical volume will be used.device=device Specifies the absolute path location for install images used during NIM master fileset installation and resource creation. By default, <code>device</code> is <code>/dev/cd0</code>.
-B	Disables the creation of the backup image.

Item	Description
> > -c	Creates Secure Sockets Layer (SSL) keys and SSL certificates that must be used during NIM client communication when OpenSSL is installed on the NIM master machine. The SSL certificates are later copied to NIM clients by using the nimclient -c command. Note: This option is supported only for AIX 7.3 with Technology Level 2, and later.
	<<
-F	Disables the creation of the file system.
-L	Disables the creation of the lpp_source resource.
-v	Enables verbose debug output during command execution.

Location

/usr/sbin/nim_master_setup

Exit Status

Returns zero (0) upon success.

Security

Access Control: You must have root authority to run the `nim_master_setup` 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 install the NIM master fileset and initialize the NIM environment using install media located in device `/dev/cd1`, type:

```
nim_master_setup -a device=/dev/cd1
```

2. To initialize the NIM environment without creating NIM install resources, type:

```
nim_master_setup -a mk_resource=no
```

3. To initialize the NIM environment, create NIM install resources without creating a backup image, using install media located under mount point `/cdrom`, type:

```
nim_master_setup -a device=/cdrom -B
```

4. To define NIM resources in an existing NIM environment, using install media located in device `/dev/cd0`, and create a new file system named `/export/resources/NIM` under volume group `nimvg`, type:

```
nim_master_setup -a volume_group=nimvg \
-a file_system=/export/resources/NIM
```

Note: If the file system `/export/resources/NIM` does not currently exist, then it will be created under the volume group `nimvg`. If the `nimvg` volume group does not exist, it will be created using the next empty physical volume (disk) since the disk attribute was not specified.

5. >|>| To initialize the NIM environment and to create SSL keys and certificates for NIM communication during NIM master initialization, enter the following command:

```
nim_master_setup -c
```



Files

Item	Description
/etc/niminfo	Contains variables used by NIM.
/var/adm/ras/ nim.setup	Contains log information from command execution.

nim_move_up Command

Purpose

Facilitates the enablement of new hardware in AIX environments.

Syntax

```
nim_move_up { [-S] | [-K [ -h control_host ] ] | [-r [ -R ] [ -u ] ] } { [-c NIM_client ] [ -i target_ip  
[ -ending_ip ] ] [ -s subnet_mask ] [ -g gateway ] [ -h control_host ] [ -m managed_sys ] [ -V vio_server  
[ -e ] [ -D ] ] [ -I img_src ] [ -l resource_dir ] [ -t seconds ] [ -p loops ] [ -j nimadm_vg ] [ -L lpp_source ]  
[ -U spot ] [ -B bosinst_data ] [ -E exclude_files ] [ -C script_resource ] [ -b installp_bundle ] [ -f  
fix_bundle ] { [ -n ] [ -d ] } | -O } [ -q ] [ -T ] [ -M manual_configuration_filenames ] }
```

Description

The `nim_move_up` command enables users of existing AIX environments to take advantage of the capabilities available on new POWER servers. The command provides an interface that migrates an existing AIX system onto an LPAR residing on a POWER server. The level of AIX on the original machine is raised to a level that supports operation on newer hardware. The original system's hardware resources are closely replicated on the newer hardware. By the end of the migration, the same system is fully running on the new LPAR.

In addition, `nim_move_up` can use the Virtual I/O capabilities of POWER servers by optionally migrating a client onto virtualized hardware, such as virtual disks and virtual Ethernet.

The `nim_move_up` command relies on the functionality of NIM and the NIM master's capability of remotely managing and installing NIM clients on the network. The `nim_move_up` command attempts to use the NIM master and the `nimadm` command to complete the following actions on an existing NIM client:

1. Create a system backup of the client
2. Migrate the backup's level of AIX
3. Install the backup onto an LPAR that resides on the new POWER server, which is be represented in the NIM environment as a new standalone client.

Before the new hardware is installed, the NIM master (on which the `nim_move_up` command is run) and the NIM clients on the existing hardware must be configured. The clients are the starting point of the migration and eventually turn into the new LPAR.

After a successful migration, the following conditions are true:

- The NIM master remains the same.
- The LPAR on the new POWER server correspond to the original NIM clients and are controlled by the NIM master.

- An HMC controls the LPAR on the new POWER servers by communicating with the NIM master through SSH.
- The original NIM clients remain unaffected and still in control of the NIM master.

The entire migration takes place without any downtime required on the part of the original client. The process can be completed in phases executed sequentially, which allows more control over the process, or can be executed all at once, so that no user interaction is required. The command is delivered as part of the `bos.sysmgt.nim.master` fileset and requires a functional NIM environment in order to run.

Required Flags

Item	Description
<code>-c NIM_client</code>	Specifies either a NIM standalone client (standalone object type) or a NIM machine group (<code>mac_group</code> object type). The client indicated must be reachable using the network from the NIM master and must allow the NIM master to run commands on them. If a NIM machine group is specified in this argument, it must reside in the same NIM network. The client is the target machine that will be migrated onto the new LPAR on a POWER server.
<code>-g gateway</code>	Specifies the IP address of the default gateway that the clients will be configured with after the migration to the POWER server.
<code>-h control_host</code>	Specifies the host name or IP address of the HMC that is used for hardware control of the POWER server.
<code>-i target_ip[-ending_ip]</code>	Specifies the IP address that the new migrated client will be configured with after it is installed on the POWER server. If a NIM machine group is supplied to the <code>-c</code> option, a range of IP addresses must be supplied here and there must be enough addresses in the range to enumerate the amount of clients that are to be migrated.
<code>-I img_src</code>	Specifies the path to the source of the installation images used to create the NIM resources required for migration and installation. This path can be a device (such as <code>dev/cd0</code> if using AIX product media) or a path to a location on the file system containing the installation images.
<code>-l resource_dir</code>	Specifies the path to a location on the file system that will contain any new NIM resources created through the <code>nim_move_up</code> command. The location must have enough space to accommodate an LPP_Source and a spot unless existing resources were provided through the <code>-L</code> and <code>-U</code> options.
<code>-m managed_sys</code>	Specifies the name of the managed system corresponding to the POWER server as tracked by the HMC.
<code>-s subnet_mask</code>	Specifies the subnet mask that the clients will be configured with after the migration to the POWER server.

Execution and Control Flags

Item	Description
-d	Executes <code>nim_move_up</code> in the background and returns control of the terminal to the caller. The progress of <code>nim_move_up</code> can be tracked through the <code>-S</code> flag.
-K	Configures SSH keys on the specified HMC. This allows the unattended remote execution of commands from the NIM master without password prompts. This flag cannot be used with any other options except the <code>-h</code> option.
-n	Runs only the next phase of the <code>nim_move_up</code> migration process. The <code>nim_move_up</code> command exits when the phase completes or fails. If this flag is not provided, all the subsequent phases are run and <code>nim_move_up</code> exits when they have all run or one of them has failed.
-O	Saves only supplied values. Save values provided through other options and then exits without executing any phases. This flag cannot be used with any other of the Execution and Control Flags.
-q	Specifies quiet mode. No output is displayed to the terminal (but is instead kept in the logs). This flag has no effect if <code>nim_move_up</code> runs with the <code>-d</code> flag.
-r	Unconfigures <code>nim_move_up</code> . This resets all saved data, including saved options, phase-specific data, and current phase information. This operation must be run if the migration process is to be started over for the migration of a new client or set of clients.
-R	Removes all NIM resources created by <code>nim_move_up</code> in addition to unconfiguring the environment. This flag can only be used with the <code>-r</code> option.
-S	Displays the status of the current phase or the next phase to be run. All saved values are displayed as well. The <code>nim_move_up</code> command exits immediately after displaying the information. This flag cannot be used with any other options.

Optional Flags

Item	Description
<code>-b installp_bundle</code>	Specifies an existing <code>installp_bundle</code> NIM resource whose software are installed on each of the newly migrated LPAR in phase 10 (post-installation customization) if the option is provided.

Item	Description
<i>-B bosinst_data</i>	Specifies an existing <code>bosinst_data</code> NIM resource used by <code>nim_move_up</code> to install the new clients onto the new LPAR. If this option is not provided, <code>nim_move_up</code> generates a <code>bosinst_data</code> resource with default unattended installation values.
<i>-C script_resource</i>	Specifies an existing <code>script</code> NIM resource that, if provided, <code>nim_move_up</code> will execute in phase 10 (post-installation customization) on all of the new migrated LPAR.
<i>-D</i>	Forces the use of physical storage controllers instead of virtual SCSI adapters in creating the new LPAR on the POWER server when a Virtual I/O server LPAR is specified. This flag is only valid when used with the <code>-V</code> option.
<i>-e</i>	Forces the use of physical network adapters instead of shared Ethernet adapters in creating the new LPAR on the POWER server when a Virtual I/O server LPAR is specified. This flag is only valid when used with the <code>-V</code> option.
<i>-E exclude_files</i>	Specifies an existing <code>exclude_files</code> NIM resource that <code>nim_move_up</code> uses to create a <code>mksysb</code> of the original clients. If this option is not provided, <code>nim_move_up</code> generates an <code>exclude_files</code> resource that excludes the contents of <code>/tmp</code> from the backup.
<i>-f fix_bundle</i>	Specifies an existing <code>fix_bundle</code> NIM resource whose APARs are installed on each of the newly migrated LPAR in phase 10 (post-installation customization) if the option is provided.
<i>-j nimadm_vg</i>	Specifies the volume group to be used by the underlying <code>nimadm</code> call for data caching. If this option is not provided, the default value is <code>rootvg</code> .
<i>-L lpp_source</i>	Specifies an existing <code>LPP_Source</code> NIM resource to whose AIX level the target clients will be migrated to. If this option is not provided, <code>nim_move_up</code> attempts to create a new <code>LPP_Source</code> from the installation image source provided through the <code>-I</code> option.
<i>-M manual_configuration_filenames</i>	Specifies phase4 to use these manual configuration files to the associated back-level AIX machines. This flag is effective only in phase4 of the nim_move_up command. For more information about this flag, see the Advanced usage section.

Item	Description
-p <i>loops</i>	Specifies the number of times to execute system analysis tools on the target NIM clients in analyzing resource utilization. The final resource usage data will be the average of the values obtained from each loop. This data will be taken into account when determining the equivalent POWER server resources from which the migrated LPAR will be derived. If this option is not provided, the default is 1 loop.
-t <i>seconds</i>	Specifies the number of seconds each loop runs for. If this option is not provided, the default is 10 seconds.
-T	Transports user-defined volume groups from the original clients to the new migrated LPAR.
-u	Enables <code>nim_move_up</code> to completely "roll back" entire <code>nim_move_up</code> migration. Must be used with the <code>-r</code> flag.
-U <i>spot</i>	Specifies an existing spot NIM resource that will be used in the migration and installation of the clients. If this option is not provided, a new spot is created from the <i>lpp_source</i> NIM resource provided by the <code>-L</code> and <code>-I</code> options.
-V <i>vio_server</i>	Specifies the LPAR name of a Virtual I/O server that resides on the POWER server denoted by the <code>-m</code> flag.

Exit Status

Item	Description
0	Successful completion.
<i>nonzero</i>	An error occurred.

Security

Only the root user can run this command.

Examples

1. To run the first phase and configure all the required options (`nim_move_up` must not be already configured and running), type:

```
nim_move_up -c client1 -i 192.168.1.100 -s 255.255.255.0 -g 192.168.1.1 -h hmc1.mydomain.com
-m \
my-p5 -l /big/dir -I /dev/cd0 -n
```

2. To display the status of the `nim_move_up` command's environment, including all saved configuration input and which phase is to be executed next, type:

```
nim_move_up -S
```

3. To change the saved host name to a new name and run the next phase while suppressing output, type:

```
nim_move_up -h hmc2.mydomain.com -n -q
```

4. To run all remaining phases in the background, save your agreement to accept all licenses, and have the prompt returned after the phases begin running, type:

```
nim_move_up -Y -d
```

5. To unconfigure `nim_move_up`, discard all saved input, and reset the command to run phase 1, type:

```
nim_move_up -r
```

All NIM resources previously created by `nim_move_up` remain unaffected in the NIM environment and will be used by `nim_move_up` as necessary to migrate another client.

Restrictions

The following NIM master requirements must be met before running the `nim_move_up` application:

- Running AIX 5L Version 5.3 with the 5300-03 Recommended Maintenance package, or later.
- Perl 5.6 or later.
- OpenSSH (from the Linux Toolbox CD)
- At least one standalone NIM client running AIX 4.3.3 update or later in the environment
- Product media version AIX 5L Version 5.2 with the 5200-04 Recommended Maintenance package or later, or product media version AIX 5.3 or later (the equivalent LPP_Source and spot NIM resources can also be used).

In addition, the following prerequisites must be available:

- A POWER server with sufficient hardware resources to support the target clients' equivalent POWER server configuration.
- An installed and configured Virtual I/O server is, if virtual resources will be used to migrate the clients.
- An HMC controlling the POWER server, along with sufficient privileges to power-on, power-off, and create LPAR.

The `nim_move_up` command will fail to execute properly if all of the preceding requirements are not met or if the command is executed by a non-root user.

Implementation Specifics

The `nim_move_up` command takes a phased approach to migrating an existing client onto a new LPAR. The following phases make up the process:

1. **Create NIM resources.** The NIM resources required to perform the migration steps are created if they do not already exist.
2. **Assess premigration software.** An assessment of which software is installed and which software cannot be migrated is performed on each target client. Any software missing from the LPP_Source is added from the source of the installation images (such as product media) that is provided to `nim_move_up`.
3. **Collect client hardware and usage data.** Data about each target client's hardware resources are gathered. Also, an attempt to assess the average use of those resources over a given amount of time is made.
4. **Collect POWER server resource availability data and translate client resource data.** The managed system that is provided is searched for available hardware resources. The data gathered in the previous phase is used to derive an equivalent LPAR configuration that uses the managed system's available resources. If a Virtual I/O server LPAR was provided to work with, the derived client LPAR is created with virtual I/O resources instead of physical I/O resources. The appropriate adapters and configuration are created on the Virtual I/O server as needed.
5. **Create system backups of target clients.** After NIM performs a `mksysb` of each target client, the corresponding `mksysb` NIM resources are created.

6. **Migrate each system backup.** Using the NIM resources designated by `nim_move_up`, each `mksysb` resource is migrated to the new level of AIX by the `nimadm` command. The original `mksysb` NIM resources are preserved and new `mksysb` NIM resources are created for the new migrated `mksysb` resources.
7. **Allocate NIM resources to new LPAR.** NIM standalone client objects are created for each new derived LPAR created in phase 4 using the network information provided to `nim_move_up`. Appropriate NIM resources are allocated and a `bos_inst` pull operation is run on each NIM client (NIM does not attempt to boot the client).
8. **Initiate installation on LPAR.** Each LPAR is rebooted using the control host (HMC) and the installation is initiated. The phase's execution stops after the installation has begun (that is, the progress of the installation is not monitored).
9. **Assess post-migration software.** After each installation has completed, the overall success of the migration is assessed, and a report of software problems encountered during migration is generated. If any filesets failed to migrate, the errors reported for that fileset must be corrected manually.
10. **Customize post-installation.** If an alternate `LPP_Source`, fileset list, or customization script was provided, a customized NIM operation is performed on each client with the values provided. This allows for the optional installation of additional software applications or for any additional customization.

In order to successfully migrate a NIM client onto a new LPAR, each of these phases (with the exception of phase 10, which is optional) must be executed completely successfully. If all phases completed successfully, a new NIM client object will be present in the NIM environment that represents the migrated LPAR, which will be running the level of AIX supplied through the `nim_move_up` source of installation resources.

After all prerequisites needed to run `nim_move_up` have been satisfied, the `nim_move_up` command runs in two phases: configuration and phase execution.

Configuration

Before the `nim_move_up` command can begin its phases, input must be provided to the application. The required input includes a list of the NIM clients to be migrated, TCP/IP configuration information of the new migrated LPAR, and the POWER server name. For a complete list of required `nim_move_up` configuration options, refer to the Required Flags (they also are denoted by a * (asterisk) in the `nim_move_up_config` SMIT menu). Optional input, such as whether a Virtual I/O server is specified, also affects the behavior of `nim_move_up` and the end result of the migration process (if a Virtual I/O server is specified, virtual I/O resources are used to create the migrated LPAR).

To populate the required and optional input through the SMIT interface, enter one of the following commands:

```
smitty nim_move_up_config
```

or

```
smitty nim_move_up
```

and select the `Configure nim_move_up Input Values` option.

At the menu, fill in the options with values that reflect the requirements of your environment. For further information about the `nim_move_up` command's SMIT interface, see the SMIT usage section below.

After the `nim_move_up` command's environment has been configured with the needed input, those values are remembered through subsequent runs of the `nim_move_up` command until the `nim_move_up` command environment is unconfigured. The values can be changed at any time through the SMIT menu interface or by providing the new values through command line flags. The command line interface can also be used to configure the `nim_move_up` command environment.

Note:

If you use the command line interface, the `nim_move_up` command, by default, also attempts to execute phases whenever configuration values are provided to it. To prevent phases from being executed when calling the command directly, use the `-0` flag.

Phase Execution

After all input is supplied, phase execution begins at phase 1 and continues sequentially. If a phase encounters an error, `nim_move_up` attempts to execute the failed phase the next time it runs. Optionally, you can specify that `nim_move_up` start only the next phase or attempt all remaining phases.

To start `nim_move_up` phases through the SMIT interface, type one of the following commands:

```
smitty nim_move_up_exec
```

or

```
smitty nim_move_up
```

and select the `Execute the nim_move_up Phases` option. Answer the `Execute All Remaining Phases?` option and press Enter. The phases begin executing.

To specify that `nim_move_up` execute only the next phase using the command line, type the following command:

```
nim_move_up -n
```

To specify that `nim_move_up` execute all remaining phases, type the following command:

```
nim_move_up
```

In addition to executing phases, this command can also modify saved configuration options if the appropriate flag is supplied.

SMIT Usage

The `nim_move_up` SMIT menus can be accessed using the `nim_move_up` fastpath. To invoke the root menu of `nim_move_up`, type the following command:

```
smitty nim_move_up
```

The following SMIT screens are accessible through the root menu:

Display the Current Status of `nim_move_up`

Equivalent to running `nim_move_up` with the `-S` flag. The next phase to be executed and a listing of all the saved options are displayed.

Configure `nim_move_up` Input Values

Through this screen, all required and optional input to `nim_move_up` can be configured. All values entered into the fields are saved and are remembered through subsequent runs of `nim_move_up` and through subsequent uses of this SMIT screen. This screen can be used at any time to modify saved values after phases have been run.

Execute `nim_move_up` Phases

Provides a simple interface to execute `nim_move_up` phases. The phases can be executed one at a time or all at once, depending on how the questions in this phase are answered.

Configure SSH Keys on Target HMC

Provides a simple interface for setting up SSH keys on the remote control host (HMC). This does the equivalent work of passing the `-K` flag on the command line. Configuring SSH keys on the remote control host enables the unattended remote execution of commands from the NIM master, which is necessary for completing all the phases (some of which remotely execute commands on this system).

Unconfigure `nim_move_up`

Provides an interface to unconfigure the `nim_move_up` command's environment. This removes all state information, including which phase to execute next, saved data files generated as a result of the

execution of some phases, and all saved input values. Optionally, all NIM resources created through `nim_move_up` can be removed as well. This screen does the equivalent work of the `-x` command line option.

Advanced Usage: Understanding the mig2p5 Framework

The `mig2p5` framework consists of the `/var/mig2p5` directory and serves as a means for `nim_move_up` to remember its state between subsequent invocations. Its existence and its use by `nim_move_up` is completely transparent to the user: the directory is created by `nim_move_up` and its values are initialized if it does not exist. It is removed when `nim_move_up` is unconfigured. The contents of this directory are easily readable and can be very helpful in troubleshooting problems with `nim_move_up`; the directory contains all of the logs generated in the phases and contains editable files that affect the behavior of `nim_move_up` in ways that are not allowed by the command line (such as forcing `nim_move_up` to run a certain phase out of order).

The following list describes the purpose and contents of each file in the `/var/mig2p5` directory:

config_db

Contains all of the saved configuration options passed to `nim_move_up` through the command line arguments or the `nim_move_up_config` SMIT menu. Each line in the file takes the following form:

```
option_name:value
```

current_phase

Contains the number of the phase that will be executed at the next invocation of `nim_move_up`. Before running this phase, `nim_move_up` ensures that all previous phases have run successfully. This information is also maintained elsewhere with the `mig2p5` framework.

global_log

Contains the output of all phases that have been run since the last time the `mig2p5` framework was initialized.

client_data/

Contains files that are generated by `nim_move_up` during phases 3 and 4, in which each of the original clients' system resources and utilization are monitored and quantified into configuration files. The available resources in the POWER server are also quantified into corresponding text files. All the data in these files will be taken into account when determining the hardware profile of the newly derived LPAR on the POWER server. These files are intended to be machine-readable data files for the `nim_move_up` command's internal use. Do not manually modify or create them.

phase#/

Contains data specific to the corresponding phase denoted by the number in its name (`#`). Every phase has a directory (for example, `phase1/`, `phase2/`, and so on).

phase#/log

Contains all output displayed during a phase's run. If a phase runs multiple times (such as after an error has been corrected), all new output is appended to any text already existing in the file. This log is helpful in investigating failures related to this phase after they have occurred. The `global_log` file is composed of all the phases' log files, and all output in that file is arranged in the order that it was originally displayed.

phase#/status

Indicates whether this phase succeeded or failed when it was last run. This file is used by `nim_move_up` to determine whether a subsequent phase can be run. A phase can run only if all of the previous phases' status files contain the string `success`. The status file contains the `failure` string if the phase encountered an error that caused it to fail the last time it was run.

pid

Contains the `nim_move_up` process ID number when `nim_move_up` is running in the background. This file and is cleaned up when the process finishes. As long as this file exists and contains a process ID, `nim_move_up` cannot run phases because concurrent runs of `nim_move_up` are not supported.

With the exception of the log files and the contents of the `client_data/` directory, the files in `/var/mig2p5` that comprise the `mig2p5` framework can be read and modified so that `nim_move_up` performs

tasks that it would not do through its command line and SMIT interfaces. Users are encouraged to manipulate the mig2p5 environment to make `nim_move_up` meet any specific need and to aid in the troubleshooting of any problems that might arise during the migration process.

Note: Customizing the mig2p5 framework is considered advanced usage and can yield unsatisfactory results if done incorrectly. The mig2p5 environment should only be directly modified by users who understand the changes being performed and their effect on the behavior of the `nim_move_up` application.

What is the manual configuration file and why is it needed?

During **phase4** of the `nim_move_up` command, the tool calculates various resource requirements based on the back-level AIX machine and appropriately creates an LPAR on a target POWER server. When you meet any of the following situations, you can specify what modifications you need in the manual configuration file in a predefined format and run the `nim_move_up` command:

- There is a need for more memory than that determined by the `nim_move_up` command.
- There is a virtual SCSI adapter (`vhost#`) created on a Virtual I/O server that you want to use for a Volume Group.
- You want to use a different Virtual Local Area Network (VLAN) ID than the one generated by the `nim_move_up` tool.

After the successful completion of the `nim_move_up` command, the manual configuration is applied on the target LPAR.

How do I write a manual configuration file?

Note: You must create the manual configuration file before initiating the `nim_move_up` command. You can create these files for each of the clients to be migrated and specify these files as arguments to the **-M** flag to enable the `nim_move_up` command to use the manual configuration. The file name must be of the form `path/manual_cfginfo_client_host_name`. The `path` value is the directory where the manual configuration file is located, and the `client_host_name` value is the host name of the client machine to be migrated.

For each client that is migrated to a POWER platform, the `nim_move_up` command does all of the hardware configuration-related calculations by default. This file enables you to alter or tune the configuration of the target machine as you choose.

You can change the amount of memory, the size of the volume groups and the Virtual I/O server resources to be used. For example, you can change the VSCSI server adapter to be used for the volume groups created for the target LPAR. You can also change the VLAN IDs to be used for the Ethernet adapters created for the target LPAR.

The following is a sample of the manual configuration file:

```
# manual_cfgfile_dennis file
# MEMORY = min_MB desired_MB max_MB
MEMORY = 256 512 1024
# VIO_VG_INFO = vname_src, size_in_MB, vhost_to_use
#   Where vname_src is the VG name in source machine, and
#   vhost_to_use is the virtual adapter to be used for
#   the VG specified in the VIO Server.
VIO_VG_INFO = rootvg,15344,vhost4
# VIO_VLAN_INFO = vlan_id, lpar_name, slot_number
VIO_VLAN_INFO = 1,VIO-server,2
```

The file can have any blank lines. You can add comments to the file with a `#` at the beginning of the line.

All of the `min_MB`, `desired_MB`, and `max_MB` values must be in megabytes (MB). There is no restriction on the number of spaces between these numbers.

min_MB

The minimum memory required for AIX to run.

desired_MB

The amount of memory that you want the logical partition to have when activated.

max_MB

The maximum memory when dynamic logical-partitioning operations are performed on the partition.

The values of the VIO_VG_INFO field must be comma separated. The *vgroup_name_src* value is the Volume Group in the source machine for which the manual data must be given. The *size_in_MB* value is the size of the Volume Group on the target machine and the *vhost_to_use* value is the *vhost** (virtual SCSI server adapter) to be used for this Volume Group on the target POWER server.

Similarly, the values of the VIO_VLAN_INFO field must be comma separated. The *vlan_id* value is used instead of the one used by the **nim_move_up** command for the target LPAR's Ethernet adapter. The *lpar_name* value is the LPAR name of the Virtual I/O server having the shared Ethernet adapter (SEA), and the *slot_number* value is the slot number of this SEA on the Virtual I/O server.

It is not necessary to provide all of these values. The **nim_move_up** command receives the specified values from the manual file and generates the rest based on the client configuration.

Files

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

nim_update_all Command

Purpose

Updates NIM resources and customizes NIM clients.

Syntax

```
nim_update_all [ -d device ] [ -l lpp_source resource ] [ -s spot resource ] [ -B ] [ -u ] [ -v ] client object(s)
```

Description

The `nim_update_all` command updates the install resources and clients in the NIM environment. Flags may be used for specifying which NIM resources need updating and also to disable the updating of NIM clients. The `nim_update_all` command updates the NIM environment by performing the following tasks:

- Exports the environment variable `NIM_LICENSE_ACCEPT=yes`.
 - Used for accepting software license agreement during update install.
- Adds variable entry `NSORDER=local,bind` in `/etc/environment`.
 - Necessary for name resolution when hosts only exist in `/etc/host`.
- Obtains the update level information from the media.
 - The default media location is `/dev/cd0`.
 - The media location may be modified by using the `-d` flag.
- Updates the `lpp_source`, `spot`, and `mksysb` resources.
 - The `lpp_source` resource name may be specified by using the `-l` flag.
 - The `spot` resource name may be specified by using the `-s` flag.
 - The `mksysb` resource name is obtained from the `mksysb` resource contained in the `basic_res_grp` resource group. Specify the `-B` flag to disable updating the `mksysb` resource.
- Performs an `update_all` operation on NIM clients.
 - If client objects are given, specified clients are updated.

- If client objects are omitted, all existing NIM clients are updated.
- If -u flag is used, no clients are updated.

Flags

Item	Description
-d <i>device</i>	Specifies the absolute path location for update images used during command execution. By default, <i>device</i> is /dev/cd0.
-l <i>lpp_source resource</i>	Specifies the object name for the <i>lpp_source resource</i> to update. By default, the resource name is obtained from <i>basic_res_grp</i> .
-s <i>spot resource</i>	Specifies the object name for the <i>spot resource</i> to update. By default, the resource name is obtained from <i>basic_res_grp</i> .
-B	Disables the updating of the backup image contained in <i>basic_res_grp</i> .
-u	Disables the updating of client objects.
-v	Enables verbose debug output during command execution. Security

Location

/usr/sbin/nim_update_all

Exit Status

Returns zero (0) upon success.

Security

Access Control: You must have root authority to run the *nim_update_all* command.

Examples

1. To update install resources 520lpp_res (*lpp_source*), 520spot_res (*spot*), and master_sysb (*mksysb*) contained in the *basic_res_grp* resource group, using update images located in device /dev/cd2, and update all clients in the NIM environment, type:

```
nim_update_all -d /dev/cd2
```

2. To update install resources lpp1 (*lpp_source*), spot1 (*spot*), and disable updating the *mksysb* image, using update images located in device /dev/cd0, and update the client object *machine1* in the NIM environment, type:

```
nim_update_all -l lpp1 -s spot1 \  
-B machine1
```

3. To update install resources 520lpp_res (*lpp_source*), 520spot_res (*spot*), and disable updating the *mksysb* image contained in the *basic_res_grp* resource group, using update images located in device /dev/cd0, and disable updating clients in the NIM environment, type:

```
nim_update_all -B -u
```

Files

Item	Description
/etc/niminfo	Contains variables used by NIM.
/var/adm/ras/nim.update	Contains log information from command execution.

nimadapters Command

Purpose

Defines Network Installation Management (NIM) secondary adapter definitions from a stanza file.

Syntax

```
nimadapters {-p | -d | -r } -f SecondaryAdapterFileName adapter_def_name
```

or

```
nimadapters {-p | -d | -r } -a client=Client [-a info=AttributeList] adapter_def_name
```

Description

The `nimadapters` command parses a secondary adapters stanza file to build the files required to add NIM secondary adapter definitions to the NIM environment as part of an `adapter_def` resource. The `nimadapters` command does not configure secondary adapters. The actual configuration takes place during a `nim -o bos_inst` or `nim -o cust` operation that references the `adapter_def` resource.

Note: Before using the `nimadapters` command, you must configure the NIM master. For more information, see **Configuring the NIM Master and Creating Basic Installation Resources** in *Installation and migration* guide.

Secondary Adapters File Rules

The format of the secondary adapters file must comply with the following rules:

- After the stanza header, follow attribute lines of the form: `Attribute = Value`
- If you define the value of an attribute multiple times within the same stanza, only the last definition is used.
- If you use an invalid attribute keyword, that attribute definition is ignored.
- Each line of the file can have only one header or attribute definition.
- More than one stanza can exist in a definition file for each machine host name.
- Each stanza for a machine host name represents a secondary adapter definition on that NIM client. No two secondary adapter definitions for the same machine host name can have the same location or interface_name. There should be only one definition per adapter or interface on a given NIM client.
- If the stanza header entry is the keyword `default`, this specifies to use that stanza for the purpose of defining default values.
- You can specify a default value for any secondary adapter attribute. However, the `netaddr` and `secondary_hostname` attribute must be unique. Also, the location and interface_name must be unique on a NIM client.
- If you do not specify an attribute for a secondary adapter but define a default value, the default value is used.
- You can specify and change default values at any location in the definition file. After a default value is set, it applies to all definitions following it.
- To turn off a default value for all following machine definitions, set the attribute value to nothing in a default stanza.
- To turn off a default value for a single machine definition, set the attribute value to nothing in the machine stanza.
- You can include comments in a client definition file. Comments begin with the `#` character.
- Tab characters and spaces are ignored when parsing the definition file for header and attribute keywords and values.

Note: During a `nim -o bos_inst` or `nim -o cust` operation, if NIM examines the configuration data on the client and determines that a secondary adapter is already configured with precisely the attributes requested in the `adapter_def` resource, this secondary adapter is not reconfigured.

Secondary Adapter File Keywords

The secondary adapter file uses the following keywords to specify machine attributes:

Required Attributes

machine_type = secondary | etherchannel | install

Specifying the `machine_type` attribute as `secondary` clearly distinguishes the `nimadapters` input from `nimdef` input. If a secondary adapters file is mistakenly passed to the `nimdef` command, the error can be easily detected. Stanzas with a `machine_type` of `install` will be ignored.

netaddr

Specifies the network address for the secondary adapter.

network_type = en | et | sn | ml | vi

Specifies the type of network interface, which can be one of `en`, `et`, `sn`, `ml`, or `vi`. This attribute replaces the deprecated `network_type` attribute.

subnet_mask

Specifies the subnet mask used by the secondary adapter.

Optional Attributes

adapter_attributes

Blank-separated list of physical adapter attributes and values (for example, "Attribute1=Value1 Attribute2=Value2"). To see the list of attributes that can be set for the requested physical adapter, run the command `lsattr -E -l AdapterName`.

interface_attributes

Blank-separated list of interface attributes and values (for example, "Attribute1=Value1 Attribute2=Value2"). To see the list of attributes that can be set for the requested interface, run the command `lsattr -E -l InterfaceName`. This attribute replaces the **attributes** attribute.

cable_type

Specifies the cable type (optional if `network_type` is `en` or `et`).

comments

Specifies a comment to include in the secondary adapter definition. Enclose the comment string in double quotes (").

interface_name

Specifies the name of the network interface for the secondary adapter (for example, `en1`, `sn0`, `ml0`). Do not specify both location and `interface_name`.

Note: The `interface_name` must be consistent with the `interface_type`.

location

Specifies the physical location of the adapter corresponding to this network interface. Do not specify both location and `interface_name`.

Note: Except for the multilink pseudo-device, use of the location is highly recommended. If the location is not specified and the user adds multiple adapters or adds an adapter at the same time that the operating system is reinstalled, the adapter and network interface names might be reassigned by the operating system in unexpected ways.

multiple_physloc

This attribute can be used with `etherchannel` or `VIPA` stanzas to specify the physical adapters to associate with the interface.

media_speed

Specifies the media speed (optional if `network_type` is `en` or `et`).

secondary_hostname

Host name to save in the `/etc/hosts` file with the `netaddr` attribute. This host name will not be set using the `hostname` command or `uname -S` command.

bos_preconfig

Specifies that the **tunchange** command is to change the value of tuning parameters. With the **bos_preconfig** attribute, you can change tunable parameters that have been set by the **/usr/lpp/bos.sysmgt/nim/methods/c_cfgadptrs** script with default values. The **bos_preconfig** attribute is used for the **nim -o bos_inst** command. For more information about the valid stanza and the respected stanza commands for tunable values, see the **tunchange** command.

The format for the **bos_preconfig** attribute is as follows:

```
bos_preconfig="tunchange -f nextboot -t Stanza [ -o tunable=value ... ]"
```

Requirement: You must restart the system in order for any new setting you made using the **bos_preconfig** attribute to take effect.

cust_preconfig

Specifies that the **vmo** command is to change the value of tuning parameters. With the **cust_preconfig** attribute, you can change tunable parameters that have been set by the **/usr/lpp/bos.sysmgt/nim/methods/c_cfgadptrs** script with default values. The **cust_preconfig** attribute is used for the **nim -o cust** command. For more information about valid tunable parameters, see the **vmo** command.

The format for the **cust_preconfig** attribute is as follows:

```
cust_preconfig="vmo -r [ -o tunable=value ... ]"
```

Note: You must restart the system to use the **cust_preconfig** attribute to set tunable parameters.

route

Specifies the route value to be added into network routing tables. You must specify the following values, or leave a blank space for any value that you do not want to specify:

Destination IP

The host or network for directing the route to. Specify the value as a numeric address.

Destination subnet mask

The mask for determining which network the destination IP belongs to. Specify the value as a numeric address.

Gateway IP

The network to which the packets are sent. Specify the value as a numeric address.

Each value must be separated by a double colon (:), and each additional set of the three values must be separated by a comma (,). The format for the route attribute is as follows:

```
route="DestHostA::MaskHostA::GatewayHostA, DestHostB::MaskHostB::GatewayHostB, ..."
```

For values that do not apply, you can leave it as blank but they still must be separated by a double colon as in the following example:

```
route="1.2.3.4:::5.6.7.8"
```

When you add the route attribute, using the **nimadapters** command with the **-a info** flag, you must separate the value for route with a double colon, and you must separate each additional set of three values with a space.

Secondary Adapter File Stanza Errors

A secondary adapter stanza causes an error under any of the following conditions:

- The host name that was used in the stanza header for the definition cannot be resolved.
- A required attribute is missing.
- An invalid value was specified for an attribute.
- An attribute mismatch occurs. For example, if the `interface_type` is not `en` or `et`, you cannot specify `cable_type=bnc` or `media_speed=1000_Full_Duplex`.

- The stanza contains both a location attribute and an interface_name attribute.
- Secondary adapter definitions occur multiple times for the same adapter location and the same host name.
- Secondary adapter definitions occur multiple times for the same interface_name and the same host name.

If a secondary adapter stanza is incorrect, the errors are reported, the stanza is ignored, and the following input is processed without regard to the incorrect stanza.

Example Secondary Adapter File

The following is an example of how a secondary adapter file can look:

```
# Set default values.
default:
    machine_type = secondary
    subnet_mask  = 255.255.240.0
    network_type = en
    media_speed  = 100_Full_Duplex

# Define the machine "lab1"
# Take all defaults and specify 2 additional attributes.
# Unlike the case of the client definitions that are input to the
# nimdef command, the secondary adapter definition includes at least
# one required field that cannot be defaulted.
lab1:
    netaddr = 9.53.153.233
    location = P2-I1/E1

# Change the default "media_speed" attribute.
default:
    media_speed = 100_Half_Duplex

# define the machine "test1"
# Take all defaults and include a comment.
test1:
    comments = "This machine is a test machine."

# define a machine with a VIPA interface that uses interfaces en2 and en3.
lab2:
    machine_type      = secondary
    interface_type    = vi
    interface_name    = vi0
    netaddr           = 9.53.153.235
    subnet_mask       = 255.255.255.0
    secondary_hostname = lab3
    interface_attributes = "interface_names=en2,en3"

# define a machine with an etherchannel adapter that uses the adapters at
# the following location codes P1-I4/E1 and P1/E1
lab4:
    machine_type      = etherchannel
    interface_type    = en
    interface_name    = en2
    netaddr           = 9.53.153.237
    subnet_mask       = 255.255.255.0
    multiple_physloc  = P1-I4/E1,P1/E1

# define a machine with an etherchannel adapter that uses the
# ent2 and ent3 adapters and uses mode 8023ad.
```

```
lab6:
  machine_type      = etherchannel
  interface_type    = en
  interface_name    = en2
  netaddr           = 9.53.153.239
  subnet_mask       = 255.255.255.0
  adapter_attributes = "adapter_names=ent2,ent3 mode=8023ad"
```

Flags

Item	Description
-a	<p>Assigns the following attribute=value pairs:</p> <p>client=<i>nim_client_name</i> Specifies the NIM client that will have a secondary adapter definition added or removed. This option allows you to define one secondary adapter for a client. To define multiple secondary adapters, use a stanza file.</p> <p>info=<i>AttributeList</i> When previewing or defining a secondary adapter, the info attribute must be used when the client attribute is specified. <i>AttributeList</i> is a list of attributes separated by commas. The attributes must be specified in the following order: <i>interface_type</i>, <i>location</i>, <i>interface_name</i>, <i>cable_type</i>, <i>media_speed</i>, <i>netaddr</i>, <i>subnet_mask</i>, <i>interface_attributes</i>, <i>secondary_hostname</i>, <i>machine_type</i>, <i>adapter_attributes</i>, <i>multiple_physloc</i>, <i>bos_preconfig</i>, <i>cust_preconfig</i>, <i>route</i>. Use lowercase n/a to specify that a value will not be used.</p>
-d	<p>Defines secondary adapters. A <i>Client.adapter</i> file is created in the <i>adapter_def</i> location for each valid secondary adapter definition. If the <i>nimadapters</i> command encounters existing secondary adapter definitions for a NIM client, the existing definitions are replaced.</p>
-f	<p><i>SecondaryAdapterFileName</i> Specifies the name of the secondary adapter file.</p>
-p	<p>Displays a preview operation to identify any errors. This flag processes the secondary adapter file or info attribute but does not add adapter definitions to the NIM environment.</p> <p>The preview shows the following:</p> <ul style="list-style-type: none"> • All complete and valid secondary adapter stanzas. • All invalid secondary adapter stanzas and the reason for failure. <p>Note: Specify the -p flag to verify that all stanzas are correct before using the secondary adapter file for configuring secondary adapters.</p>
-r	<p>Removes the secondary adapter definitions of a specific client or all the clients listed in a secondary adapter stanza file. If the client attribute or secondary adapter stanza file are not specified, then all the secondary adapter definitions in the <i>adapter_def</i> resource will be removed.</p>

Parameters

Item	Description
<i>adapter_def</i>	<p>This parameter is required to run the <i>nimadapters</i> command. Specifies the <i>adapter_def</i> NIM resource that is the directory containing secondary adapter definition files. An <i>adapter_def</i> resource must be defined using the <i>nim -o define</i> operation before the <i>adapter_def</i> can be used with the <i>nimadapters</i> command.</p>

Exit Status

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

Security

Access Control: You must have root authority to run the `nimadapters` 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 add the NIM secondary adapters described in the secondary adapters definition file `secondary_adapters.defs` to the `my_adapter_def` resource, type:

```
nimadapters -d -f secondary_adapters.defs my_adapter_def
```

2. To preview the client definition file `secondary_adapters.defs`, type:

```
nimadapters -p -f secondary_adapters.defs my_adapter_def
```

3. To define a NIM secondary adapter for a client called `pilsner`, type:

```
nimadapters -d \  
-a info="en,P2-I1/E1,n/  
a,bnc,1000_Full_Duplex,9.53.153.233,255.255.254.0,n/a,n/a,n/a,n/a,n/a" \  
-a client=pilsner my_adapter_def
```

4. To remove the NIM secondary adapter definitions for a client called `pilsner` from the `my_adapter_def` resource, type:

```
nimadapters -r -a client=pilsner my_adapter_def
```

5. To remove the NIM secondary adapter definitions for clients defined in the file `secondary_adapters.defs`, type:

```
nimadapters -r -f secondary_adapters.defs my_adapter_def
```

6. To remove all the NIM secondary adapter definitions from the `my_adapter_def` resource, type:

```
nimadapters -r my_adapter_def
```

Files

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

nimadm Command

Purpose

The **nimadm** (Network Install Manager Alternate Disk Migration) command is a utility that allows the system administrator to do the following actions:

- Create a copy of rootvg to a free disk (or disks) and simultaneously migrate it to a new version or release level of AIX.
- Using a copy of rootvg, create a new NIM mksysb resource that has been migrated to a new version or release level of AIX.
- Using a NIM mksysb resource, create a new NIM mksysb resource that has been migrated to a new version or release level of AIX.
- Using a NIM mksysb resource, restore to a free disk (or disks) and simultaneously migrate to a new version or release level of AIX.

The **nimadm** command uses NIM resources to perform these functions.

Syntax

Perform alternate disk migration:

```
nimadm -l lpp_source -c NIMClient -s SPOT -d TargetDisks [-a PreMigrationScript] [-b installp_bundle] [-z PostMigrationScript] [-e exclude_files] [-i image_data] [-j VGname] [-m NFSMountOptions] [-o bosinst_data] [-P Phase] [-j VGname] [-Y] [-F] [>|< -A <] [-D] [-E] [-V] [-B | -r]
```

Cleanup alternate disk migration on client:

```
nimadm -C -c NIMClient -s SPOT [-F] [-D] [-E]
```

Wake-up volume group:

```
nimadm -W -c NIMClient -s SPOT -d TargetDisks [-m NFSMountOptions] [-z PostMigrationScript] [-F] [-D] [-E]
```

Put-to-sleep volume group:

```
nimadm -S -c NIMClient -s SPOT [-F] [-D] [-E]
```

Synchronize alternate disk migration software:

```
nimadm -M -s SPOT -l lpp_source [-d device] [-P] [-F]
```

mksysb to client migration:

```
nimadm -T NIMmksysb -c NIMClient -s SPOT -l lpp_source -d TargetDisks -j VGname -Y [-a PreMigrationScript] [-b installpBundle] [-z PostMigrationScript] [-i ImageData] [-m NFSMountOptions] [-o bosinst_data] [-P Phase] [-F] [-A] [-D] [-E] [-V] [-B | -r]
```

mksysb to mksysb migration:

```
nimadm -T NIMmksysb -O mksysbfile -s SPOT -l lpp_source -j VGname -Y [-N NIMmksysb] [-a PreMigrationScript] [-b installp_bundle] [-z PostMigrationScript] [-i image_data] [-m NFSMountOptions] [-o bosinst_data] [-P Phase] [-F] [-A] [-D] [-E] [-V]
```

Client to mksysb migration:

```
nimadm -c nim_client -O mksysbfile -s SPOT -l lpp_source -j VGname -Y [-N NIMmksysb] [-a PreMigrationScript] [-b installp_bundle] [-z PostMigrationScript] [-i image_data] [-m NFSMountOptions] [-o bosinst_data] [-P Phase] [-e exclude_files] [-F] [-A] [-D] [-E] [-V]
```


Description

The **nimadm** command is a utility that allows the system administrator to create a copy of **rootvg** to a free disk (or disks) and simultaneously migrate it to a new version or release level of AIX. The **nimadm** command uses NIM resources to perform this function.

There are several advantages to using the **nimadm** command over a conventional migration:

1. Reduced downtime. The migration is performed while the system is up and functioning normally. There is no requirement to boot from install media, and the majority of processing occurs on the NIM master.
2. The **nimadm** command facilitates quick recovery if migration fails. As the **nimadm** command uses **alt_disk_install** to create a copy of **rootvg**, all changes are performed to the copy (**altinst_rootvg**). In the event of serious migration installation failure, the failed migration is cleaned up and there is no need for the administrator to take further action. In the event of a problem with the new (migrated) level of AIX, the system can be quickly returned to the pre-migration operating system by booting from the original disk.
3. The **nimadm** command allows a high degree of flexibility and customization in the migration process. This is done with the use of optional NIM customization resources: **image_data**, **bosinst_data**, **exclude_files**, **pre-migration script**, **installp_bundle**, and **post-migration script**.

This document provides information pertaining to the **nimadm** command. For complete coverage of **alt_disk_install**, NIM, migration, and other related installation issues, see [Installing AIX](#).

nimadm local disk caching

Local disk caching allows the NIM master to avoid having to NFS write to the client, which can be useful if the **nimadm** operation is not performing optimally due to an NFS write bottle neck. If this function is invoked with the **-j VGname** flag, the **nimadm** command creates file systems on the specified volume group (on the NIM master) and uses streams to cache all of the data from the client to these file systems.

The advantages and disadvantages to this function are as follows:

Advantages:

1. Improved performance for **nimadm** operations that are on relatively slow networks.
2. Improved performance for **nimadm** operations that are bottle necked in NFS writes (NFS writes are expensive).
3. Decreased CPU usage on the client.
4. Client file systems are not exported.

Disadvantages:

1. Cache file systems take up space on the NIM master (you must have enough space to host the client's **rootvg** file systems and migration space for each client)
2. Increased CPU usage on the master.
3. Increased I/O on the master (for optimal performance use a volume group (disk) that does not contain the NIM resource being used in the operation).

How to execute disk caching:

1. Make sure you are at the latest level of **bos.alt_disk_install.rte** on the NIM master.
2. Add the **-j VGName** flag to any **nimadm** operations. For example:

```
nimadm -j rootvg ...
```

or

```
nimadm -j cachevg
```

You can exclude specific file systems (which are not involved in the migration) from being cached over the network (they are still copied locally to **altinst_rootvg** on the client). To specify a list of file systems to

be excluded from network caching, you must create a file in the location of the SPOT resource that is used for the migration. To get the exact location of the SPOT path, enter:

```
# lsrim -a location SpotName
```

You must name the file in the following format:

```
Nim_Client.nimadm_cache.excl
```

Note: This file applies to the NIM client specified in *Nim_Client*. The full path should be:

```
Spot_Location/Nim_Client.nimadm_cache.excl
```

For example: `/nim_resources/520spot/usr/myclient.nimadm_cache.excl`.

To exclude a file system from caching, enter one file system (to be excluded) per line in this file. While excluding a file system, ensure that you:

1. Do not exclude any file systems that are involved in the migration process. In other words, these file systems contain software files that are migrated. This can lead to unpredictable results.
2. Do not (cannot) exclude the following AIX file systems: `/`, `/usr`, `/var`, `/opt`, `/home`, and `/tmp`.

With disk caching, the `nimadm` command changes the following four phases (all other phases remain the same) :

Phase 2: The NIM master creates local cache file system in specified target volume group (on the NIM master).

Phase 3: The NIM master populates the cache file systems with the client's data.

Phase 9: The NIM master writes all migrated data to the client's alternate `rootvg`.

Phase 10: The NIM master cleans up and removes the local cache file systems.

nimadm requirements

The **nimadm** requirements are:

1. The NIM master must have the same level of **bos.alt_disk_install.rte** installed in its **rootvg** and the SPOT which is used to perform the migration. (Note: it is not necessary to install the **alt_disk_install** utilities on the client).
2. The selected **lpp_source** NIM resource, and selected SPOT NIM resource must match the AIX level to which you are migrating.
3. The NIM master must be at the same or higher AIX level then the level being migrated to.
4. The client (the system to be migrated) must be at AIX 4.3.3 or higher.
5. The client must have a disk (or disks) large enough to clone the **rootvg** and an additional 500 Megs (approximately) of free space for the migration. The total amount of required space depends on original system configuration and **nimadm** customization.
6. The target client must be a registered with the master as a standalone NIM client (see the **niminit** command for more information). The NIM master must be able to execute remote commands on the client using the **rshd** protocol.
7. The NIM master must be able to execute remote commands on the client using the **rshd** protocol.
8. The NIM master and client must both have a minimum of 128 megabytes of RAM.
9. A reliable network, which can facilitate large amounts of NFS traffic, must exist between the NIM master and the client. The NIM master and client must be able to perform NFS mounts and read/write operations.
10. The client's hardware and software must support the AIX level that is being migrated to and meet all other conventional migration requirements.

11. All application and database servers, such as DB2 and LDAP, must be stopped before you run the **nimadm** command to clone the rootvg of a client system. Otherwise, the application servers and the database servers do not start normally after the **nimadm** command operations are complete.

Note: If you cannot meet requirements 1-10, you must perform a conventional migration. If you cannot meet requirement 11, then migration is not possible.



Attention: Before performing a **nimadm** migration you must agree to all software license agreements for software to be installed. You can do this by specifying the **-Y** flag as an argument to the **nimadm** command or setting the **ADM_ACCEPT_LICENSES** environment variable to "yes".

nimadm limitations

The following limitations apply to the **nimadm** command:

1. If the trusted computing base (TCB) is turned on in the client's **rootvg**, you must disable it (permanently), use the disk caching option (**-j**), or perform a conventional migration. This limitation exists because TCB must access the file metadata, which is not accessible through network file system (NFS).
2. All NIM resources used by the **nimadm** command must be local to the NIM master.
3. Although there is almost no interference with the client's active **rootvg** volume group during the migration, the client might experience minor reduction in performance due to increased disk I/O, biod activity, and CPU usage associated with cloning the **alt_disk_install** command.
4. You might need to tune the NFS to optimize the **nimadm** performance.
5. The reserved directory names on the NIM client for **nimadm** are as follows: /ALT_MIG_SPOT, /ALT_MIG_EXCL, /ALT_MIG_IMAGES, and /ALT_MIG_IMD. The **nimadm** command fails if these names are used.

NIM resources used by nimadm:

SPOT resource (-s flag)

The NIM spot resource is required for all **nimadm** operations (migration, cleanup, wake-up, sleep). All **nimadm** and **alt_disk_install** utilities that are used by the client are installed in this resource. It is not necessary to install **nimadm** software on the client. The NIM cust operation must be used to install the following file sets into the spot:

- Required: **bos.alt_disk_install.rte** (must match the NIM master's level).
- Optional message catalog: **bos.msg.\$LANG.alt_disk_install.rte**

lpp_source resource (-l flag)

This NIM resource is the source of install images that are used to migrate the system. It is required for **nimadm** migration operations. The **lpp_source** must contain all system images for the level being migrated to (check the **lpp_source** images attribute in **lsnim -l lpp_source** output). It must also contain any optional **installp** images that need to be migrated.

pre-migration

This script resource that is run on the NIM master, but in the environment of the client's **alt_inst** file system that is mounted on the master (this is done by using the **chroot** command). This script is run before the migration begins.

post-migration

This script resource is similar to the **pre-migration** script, but it is executed after the migration is complete.

image_data

Specifies an **image_data** resource that is passed to **alt_disk_install** (as arguments to the **-i** flag). NIM allocates and mount this resource on the client before calling **alt_disk_install**.

exclude_files

Specifies an **exclude_files** resource that is passed to **alt_disk_install** (as an argument to the **-e** flag). NIM allocates and mount this resource on the client before calling **alt_disk_install**.

installp_bundle

This NIM resource specifies any additional software that the **nimadm** command installs after completing the migration.

bosinst_data

This NIM resource specifies various install settings that may be used by the **nimadm** command.

The nimadm migration process

The **nimadm** command performs migration in 12 phases. Each phase can be executed individually by using the **-P** flag. The **nimadm** phases must be run sequentially. The **nimadm** phases are as follows:

1. The master issues an **alt_disk_install** command to the client that makes a copy of the **rootvg** volume group to the target disks (coincidentally this is Phase 1 of the **alt_disk_install** process). In this phase **altinst_rootvg** (alternate rootvg) is created. If a target **mksysb** is specified, the **mksysb** is used to create a **rootvg** volume group by using local disk caching on the NIM master.
2. The master runs remote client commands to export all the **/alt_inst** file systems to the master. The file systems are exported as read/write with root access to the master. If a target **mksysb** is specified, the cache file systems are created based on the image data from the **mksysb**.
3. The master NFS mounts the file systems exported in Phase 2. If a target **mksysb** is specified, the **mksysb** archive is restored in the cache file systems that was created in phase 2.
4. If the pre-migration script resource is specified, the script is executed at this time.
5. System configuration files are saved. Initial migration space is calculated and appropriate file system expansions are made. "bos" is restored and the device database is merged (similar to a conventional migration). The migration merge methods are executed and some miscellaneous processing takes place.
6. The file sets of the system are migrated by using the **installp**. Any required RPM images are installed during this phase.
7. If the **post-migration** script resource is specified, the script is executed at this time.
8. **bosboot** is executed to create a client boot image that is written to the client's boot logical volume (**hd5**).
9. The mounts that are made on the master in phase 3 are removed.
10. The client exports that are created in phase 2 are removed.
11. The **alt_disk_install** is called again (phase 3 of **alt_disk_install**) to make final adjustments and place **altinst_rootvg** to sleep. The bootlist is set to the target disk (unless the **-B** flag is used). If an output **mksysb** is specified, the cache is archived into a **mksysb** file and made into a NIM **mksysb** resource.
12. Cleanup is executed to end the migration. The client is rebooted, if the **-r** flag is specified.

Note: The **nimadm** command supports migrating several clients simultaneously.

>|Boot logical volume size|<

The **nimadm** command creates a boot logical volume with appropriate size to accommodate the new boot image. When you migrate your current AIX version to AIX 7.3, the size of the boot logical volume is 40 MB.

nimadm cleanup operation

This operation, indicated with the **"-C"** flag, is designed to clean up after a failed migration that for some reason did not perform a cleanup itself. It can also be used to clear a previous migration in order to perform a new migration.

nimadm wake-up and sleep

After a migration completes, the **nimadm** command can be used to "wake-up" the migrated **altinst_rootvg** or the original **rootvg** (if booted from the migrated disk). The **nimadm** wake-up (**-W** flag) performs an **alt_disk_install** wake-up, NFS exports the **/alt_inst** file systems, and mounts them on the NIM master. The **nimadm** sleep function (**-S** flag) reverses the wake-up by unmounting the NIM master mounts, unexporting the **/alt_inst** file systems, and executing the **alt_disk_install** sleep function on the client.

Flags

Item	Description
-a <i>PreMigrationScript</i>	Specifies the pre-migration NIM script resource.
> > -A	Adds a timestamp at the start of each phase of the nimadm operation. The timestamp is displayed in MM-DD-YYYY HH-MM-SS format. <<<
-b <i>installp_bundle</i>	Specifies the <i>installp_bundle</i> NIM resource.
-B	Specifies not running bootlist after the nimadm migration. If set, then -r flag cannot be used.
-c <i>ClientDisks</i>	Specifies the NIM defined client which is the target of this nimadm operation. This flag is required for all nimadm operations.
-C	Performs nimadm cleanup.
-d <i>TargetDisks</i>	Specifies the client target disk which is used to create altinst_rootvg (the volume group that is migrated).
-D	Sets the nimadm command into debug mode. This function must only be used to debug nimadm related problems and is not set by default.
-e <i>exclude_files</i>	Specifies the exclude_files NIM resource. This resource is used by the alt_disk_install command during Phase 1.
-E	Enters the nimadm debugger if a serious migration error occurs.
-F	Forces a client to unlock. Normally, the nimadm command locks a client to perform various operations. While the client is locked, other nimadm or NIM operations cannot be performed. This flag must ONLY be used in the unusual condition that a client is incorrectly locked (this can happen if for some reason the nimadm command could not call cleanup after a failure).
-i <i>image_data</i>	Specifies the image_data NIM resource. This resource is used by the alt_disk_install command during Phase 1 and 11.
-j <i>VGname</i>	Creates file systems on the specified volume group (on the NIM master) and uses streams to cache all of the data from the client to these file systems.
-l <i>lpp_source</i>	Specifies the <i>lpp_source</i> NIM resource to be used for this nimadm operation. This flag is required for migration operations.
-m <i>NFSMountOptions</i>	Specifies arguments that are passed to the mount command that mounts client resources on the master. This flag can be used to tune nimadm related NFS performance.
-M	Verifies that the levels of the alt_disk_install software (bos.alt_disk_install) on the NIM master, SPOT, <i>lpp_source</i> , and optional device are synchronized (match). If there is no match, the nimadm command installs the highest level found in the <i>lpp_source</i> or optional device.
-N <i>NIMmksysb</i>	Specifies the unique new NIM mksysb resource to create. If the -N flag is specified, the -O flag must be specified.
-o <i>bosinst_data</i>	Specifies bosinst_data NIM resource.
-O <i>mksysbfile</i>	Specifies the file pathname for the migrated mksysb . If the -O flag is specified, the -j flag and either the -c or -T flag must be specified.
-P <i>Phase</i>	The phase to execute during this invocation of the nimadm command. If there is more than one phase, the phases must be separated by spaces or commas. Valid phases are 1 through 12.
-r	Specifies that the client must reboot after nimadm migration is complete.

Item	Description
-s <i>SPOT</i>	Specifies the SPOT NIM resource to be used for this nimadm operation. This flag is required for all nimadm operations.
-S	Performs the nimadm "sleep" function. This function must be executed to end a nimadm "wake-up".
-T <i>NIMmksysb</i>	Specifies an existing NIM mksysb resource to migrate. If the -T flag is specified, the -j flag and either the -O or -c flag must be specified.
-V	Turns on verbose output.
-W	Performs the nimadm "wake-up" function.
-Y	Agrees to required software license agreements for software to be installed.
-z <i>PostMigrationScript</i>	Specifies the post-migration NIM script resource.

Exit Status

0

All the **nimadm** command related operations completed successfully.

>0

An error occurred.

Security

Access Control: You must have root authority to run the **nimadm** 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 execute **nimadm** migration to target NIM client **aix1**, using NIM **SPOT** resource **spot1**, NIM **lpp_source** resource **lpp1**, and target disks **hdisk1** & **hdisk2**. Note that the **-Y** flag agrees to all required software license agreements for software to be installed, enter the following:

```
nimadm -c aix1 -s spot1 -l lpp1 -d "hdisk1 hdisk2" -Y
```

2. To execute the same operation as in the example above to **hdisk2**, and also run pre-migration script **nimscript1** and post-migration script **nimscript2**, type the following:

```
nimadm -c aix1 -s spot1 -a nimscrip1 -z nimscript2 -l lpp1 -d hdisk1 -Y
```

3. To execute **nimadm** cleanup on client **aix1**, using NIM **SPOT** resource **spot1**, type the following:

```
nimadm -C -c aix1 -s spot1
```

4. To create a migrated new mksysb resource of a client with the filename **nim1**, type the following:

```
nimadm -c aix1 -s spot1 -l lpp1 -O /export/mksysb/mksysb1 -j vg00 -Y -N nim1
```

5. To create a new migrated mksysb resource with the filename **nim3** from an existing NIM mksysb resource, type the following:

```
nimadm -s spot1 -l lpp1 -j vg00 -Y -T nim2 -O /export/mksysb/m2 -N nim3
```

6. To migrate an existing NIM resource and put it on a client, type the following:

```
nimadm -c aix1 -s spot1 -l lpp1 -d hdisk1 -j vg00 -T nim2 -Y
```

Note: No changes are made to the nim2 NIM mksysb resource.

7. >| To show timestamps for **nimadm** migration to target NIM client `flan003` by using NIM **SPOT** resource `spot1`, NIM **lpp_source** resource `lpp1`, and target disk `hdisk1`, enter the following command:

```
nimadm -j nimadm -c flan003 -s spot1 -l lpp1 -d hdisk1 -Y -A
```

An output that is similar to the following example is displayed:

```
Initializing the NIM master.

Starting Alternate Disk Migration.
+-----+
01-12-2023 01:21:29 Executing nimadm phase 1.
+-----+
Cloning altinst_rootvg on client, Phase 1.

for backup and restore into the alternate file system...
Phase 1 complete.
+-----+
01-12-2023 01:22:32 Executing nimadm phase 2.
+-----+
Creating nimadm cache file systems on volume group nimadm.
```

|<

Files

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

nimclient Command

Purpose

Allows Network Installation Management (NIM) operations to be performed from a NIM client.

Syntax

To Enable or Disable the NIM Master's Push Permissions

```
nimclient { -p } | { -P }
```

To Enable or Disable Cryptographic Authentication for NIM Master Push Operations

```
nimclient { -c } | { -C }
```

To List Information about the NIM Environment

```
nimclient -l LsnimParameters
```

To Set the Date and Time to That of the NIM Master

```
nimclient -d
```

To Perform a NIM Operation

```
nimclient -o Operation [ -a Attribute=Value ] ...
```

Description

The **nimclient** command is used by workstations that are NIM clients to pull NIM resources. This command can enable or disable the NIM master server's ability to initiate workstation installation and customization for the workstation. The **nimclient** command can be used to generate a list of available NIM resources or display the NIM resources that have already been allocated to the client. A limited set of NIM operations can also be performed by the **nimclient** command using the **-o** flag.

Flags

Item	Description
-a <i>Attribute=Value</i>	Passes information to NIM operations. From the master Use the lsnim -q Operation -t Type command to get a list of valid attributes for a specific operation. From the client Use the nimclient -l -q Operation -t Type command to get a list of valid attributes for a specific operation.
-c	Enables SSL authentication during NIM master push operations. Note: OpenSSL certificates must be configured on the NIM master using the nimconfig -c command. The SSL certificate is copied from the NIM master when nimclient -c is executed.
-C	Disables SSL authentication and uses standard nimsh security during NIM master push operations.
-d	Sets the client's date and time to that of the master.
-e	Configures NIM client environment to support the dual-stack mode, which means implementation of IPv4 and IPv6 protocols over a single interface definition. Note: You must run the autoconf6 command and start the ndpd-host daemon prior to configuring the dual-stack environment.
-l <i>Lsnim parameters</i>	Executes the lsnim command on the master using the lsnim parameters that you specify. All the parameters which you use with this option must adhere to the syntax rules of the lsnim command. Note that some lsnim syntax requires the use of a NIM object name. To find out what the NIM name is for your machine, look in the /etc/niminfo file.

Item	Description
-o <i>Operation</i>	<p>Performs the specified operation. The possible operations are:</p> <p>allocate Allocates a resource for use.</p> <p>bos_inst Performs a BOS installation.</p> <p>change Changes an object's attributes.</p> <p>check Checks the status of a NIM object.</p> <p>cust Performs software customization.</p> <p>deallocate Deallocates a resource.</p> <p>diag Enables a machine to boot a diagnostic image.</p> <p>maint_boot Enables a machine to boot in maintenance mode.</p> <p>reset Resets an object's NIM state.</p> <p>showres Displays the contents of a NIM resource.</p>
-p	Enables the NIM master to push commands.
-P	Removes the NIM master's permissions to push commands.
	Note: The master can override this restriction by using the -F flag.

Security

Access Control: You must have root authority to run the **nimclient** 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 list all the NIM resources which are available to this machine when its NIM name is `pluto`, enter:

```
nimclient -l -L pluto
```

2. To list all the Shared Product Object Trees (SPOTs) which are available to this machine when its NIM name is `pluto`, enter:

```
nimclient -l -L -t spot pluto
```

3. To list the operations which may be initiated from this machine, enter:

```
nimclient -l -p -s pull_ops
```

4. To prevent the NIM master from running commands locally on the client, enter:

```
nimclient -P
```

5. To allocate a spot resource named `myspot`, an **lpp_source** resource named `images`, and an **installp** bundle file name `dept_bundle`, enter:

```
nimclient -o allocate -a spot=myspot -a lpp_source=images \  
-a installp_bundle=dept_bundle
```

6. To perform a base system installation after the required resources have been allocated, enter:

```
nimclient -o bos_inst
```

7. From a standalone client, to allocate an **lpp_source** and install a software product such that the image for the installable option, `adt`, is contained in the **lpp_source**, `images`, enter:

```
nimclient -o allocate -a lpp_source=images
```

Then enter:

```
nimclient -o cust -a filesets="adt"
```

8. From a standalone client, to allocate an **lpp_source** and install a software product such that the image for the installable option, `adt`, is contained in the **lpp_source**, `images`, and the name of the installable option is contained in the **installp_bundle**, `bundle3`, enter:

```
nimclient -o allocate -a lpp_source=images \  
-a installp_bundle=bundle3
```

Then enter:

```
nimclient -o cust
```

9. To install all fileset updates associated with APAR `IX12345`, residing in the **lpp_source** `updt_images`, enter:

```
nimclient -o allocate -a lpp_source=updt_images  
nimclient -o cust -afixes=IX12345
```

10. From the NIM stand-alone, to run a live update for an APAR `IX12345`, residing in the **lpp_source** `lpp_source1`, enter:

```
nimclient -o allocate -a lpp_source=lpp_source1  
nimclient -o cust -a live_update=yes -a filesets=IX12345
```

11. To update all installed software on the client with the latest updates from the `updt_images` **lpp_source**, enter:

```
nimclient -o allocate -a lpp_source=updt_images  
nimclient -o cust -afixes=update_all
```

12. To enable the system to boot in maintenance mode using a SPOT resource named `spot1`, enter:

```
nimclient -o maint_boot -a spot=spot1
```

This sets up the maintenance boot operation, but you must initiate the network boot locally.

13. To show the contents of the config script `script1`, enter:

```
nimclient -o showres -a resource=script1
```

14. To show the contents of the `bosinst.data` resource `bosinst_data1`, enter:

```
nimclient -o showres -a resource=bosinst_data1
```

15. To list all the filesets in the lpp_source lpp_source1 relative to what is currently installed on the machine machine1, from the NIM client machine machine1, enter:

```
nimclient -o showres -a resource=lpp_source1
```

The **reference** attribute is automatically supplied by the **nimclient** command.

16. To list all problems fixed by software on the lpp_source lpp_source1, use:

```
nimclient -o showres -a instfix_flags="T" -a resource=lpp_source1
```

17. To install the filesets listed in the NIM **installp_bundle** client_bundle using the **lpp_source** client_images, while automatically allocating these resources during the installation operation, enter:

```
nimclient -o cust -a installp_bundle=client_bundle \  
-a lpp_source=client_images
```

18. To perform a base system installation while automatically allocating all applicable resources from the NIM resource group named client_grp, enter:

```
nimclient -o bos_inst -a group=client_grp
```

19. To perform a base system installation while automatically allocating all applicable resources from the NIM group defined as the default resource group on the master, enter:

```
nimclient -o bos_inst
```

20. To copy an SSL certificate and enable SSL authentication, type:

```
nimclient -c
```

Note: OpenSSL must be installed on the NIM client prior to using this command option.

21. To configure a dual-stack environment, type:

```
nimclient -e en0
```

Files

Item	Description
/etc/niminfo	Contains variables used by NIM.

nimconfig Command

Purpose

Initializes the Network Installation Management (NIM) master package.

Syntax

To initialize the NIM master package:

```
nimconfig -a pif_name=Pif -a netname=Objectname [ -a master_port=PortNumber ] [ -a platform=Value ] [ -a registration_port=PortNumber ] [ -a ring_speed=Speed | -a cable_type=CableType ] [ -c ]
```

To configure Secure Sockets Layer (SSL) for the NIM Environment:

```
nimconfig -c
```

To rebuild the /etc/niminfo file:

nimconfig -r

Description

The **nimconfig** command initializes the NIM master package. You must initialize the package before any other NIM commands can be used. When you use the **-a** flag to supply the proper attributes, the **nimconfig** command initializes the NIM environment by performing the following tasks:

- Defines a network object specified by the *ObjectName* parameter to represent the network to which the NIM master's primary interface, specified by the *Pif* parameter, is connected.
- Completes the definition of the NIM master by connecting it to the newly defined network object.
- Defines a resource object to represent the network boot resource, which is managed automatically by NIM.
- Defines a resource object to represent the customization scripts that NIM automatically builds to perform customization.
- Starts the **NIM** communications daemon, **nimesis**.

Flags

Item	Description
-a	Assigns the following attribute=value pairs: pif_name=<i>Pif</i> Designates the primary network interface for the NIM master. This value must be a logical interface name (such as tr0 or en0) is in the available state. master_port=<i>PortNumber</i> Specifies the port number of the nimesis daemon used for NIM client communication. platform=<i>Value</i> Specifies the platform. The supported platforms are: rs6K Micro Channel-based, uniprocessor models for AIX 5.1 and earlier rs6ksmp Micro Channeled-based, symmetric multiprocessor models for AIX 5.1 and earlier rspc PowerPC® PCI bus-based, uniprocessor models for AIX 5.1 and earlier rspcsmp PowerPC PCI bus-based, symmetric multiprocessor models for AIX 5.1 and earlier netname=<i>ObjectName</i> Specifies the name you want the nimconfig command to use when creating the network object to represent the network to which the master's primary interface connects. ring_speed=<i>Speed</i> Speed in Mbps. When the pif_name refers to a token ring network, this value must be given. Acceptable values are: 4 16

Item	Description
-a (continued)	<p>cable_type=CableType Specifies the Ethernet cable type. When the pif_name refers to an Ethernet network, this value must be given. Acceptable values are:</p> <p>bnc dix N/A</p> <p>registration_port=PortNumber Specifies the port number used for NIM client registration.</p> <p>Note: If you do not specify port numbers on the command line, the port numbers in the /etc/services file for NIM are used. If the /etc/services file does not contain entries for the NIM ports nim and nimreg, the default values of 1058 for master_port and 1059 for registration_port are used.</p> <p>ip_env=6 Configures the NIM environment to support the dual-stack implementation of the IPv4 and IPv6 protocol over a single interface definition. This option creates an ent6 network object and defines a secondary interface (if2) using the ent6 network object information.</p> <p>When you specify the ip_env option, the following flags are also required:</p> <p>pif_name=Pif Specifies interface name.</p> <p>netname=ObjectName Specifies the name of the new ent object that represents the IPv6 network.</p> <p>cable_type=CableType Specifies the type of the Ethernet cable.</p> <p>Note: You must run the autoconf6 command and start the ndpd-host daemon prior to configuring the dual-stack environment.</p>
-c	Creates SSL keys and SSL certificates that must be used during NIM client communication when Open Secure Sockets Layer (OpenSSL) is installed on the NIM master machine. The SSL certificates are later copied to NIM clients by using the nimclient -c command.
-r	Rebuilds the /etc/niminfo file on the master using the information already exists in the NIM database. Note that if the bos.sysmgt.nim.master package has not been configured on this machine, this option will fail. This option is provided in case the /etc/niminfo file is accidentally removed by a user.

Security

Access Control: You must have root authority to run the **nimconfig** 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 initialize the NIM environment using token ring and the default NIM ports for network communications, enter the following command:

```
nimconfig -a pif_name=tr0 -a netname=net1 -a ring_speed=16
```

2. To initialize the NIM environment using Ethernet and the default NIM ports, enter the following command:

```
nimconfig -a pif_name=en0 -a master_port=1058 \  
-a netname = net2 -a cable_type=bnc
```

3. To rebuild the **/etc/niminfo** file on the NIM master when that machine has already been correctly configured as a master, enter the following command:

```
nimconfig -r
```

4. To initialize the NIM master using an ATM network interface, enter the following command:

```
nimconfig -a pif_name=at0 -a master_port=1058 -a netname=ATMnet
```

Note: Because an interface to an ATM network does not currently support booting over the network, this operation will define a generic network object corresponding to the master's subnet.

5. To initialize the NIM environment using TCP/IP port 1060 for NIM client communications and TCP/IP port 1061 for NIM client registration, enter the following command:

```
nimconfig -a pif_name=tr0 -a netname=net2 -a master_port=1060 \  
-a registration_port=1061 -a ring_speed=16
```

6. To create SSL keys and certificates for NIM communication, enter the following command:

```
nimconfig -c
```

Note: OpenSSL must be installed on the NIM master prior to using this command option.

7. **>** To create SSL keys and SSL certificates for NIM communication during NIM master initialization, enter the following command:

```
nimconfig -a netname=net1 -a pif_name=en0 -a cable_type=N/A -a verbose=4 -c
```

Note: For NIM communication, configuring SSL keys and SSL certificates during NIM master initialization is supported only for AIX 7.3 with Technology Level 2, and later.

|<

8. To configure a dual-stack environment, create an ent6 network, and define a secondary interface, enter the following command:

```
nimconfig -a ip_env=6 -a netname=dualstack_v6 -a pif_name=en0 -a cable_type=N/A
```

Files

Item	Description
/etc/niminfo	Contains variables used by NIM.

nimdef Command

Purpose

Defines Network Installation Management (NIM) clients from a stanza file.

Syntax

```
nimdef [ -p | -d | -c ] -f Name
```

Description

The **nimdef** command parses a definition stanza file to build the commands required to add NIM client definitions to the NIM environment.

The **nimdef** command can also create NIM networks and NIM machine groups automatically in the NIM environment to support the new client definitions.

Note: Before using the **nimdef** command, you must configure the NIM master. (See **Basic NIM operations and configuration** in *Installation and migration* for more information.)

Client Definition File Rules

The format of the client definition file must comply with the following rules:

- After the stanza header, follow attribute lines of the form *Attribute = Value*.
- If you define an attribute value multiple times within the same stanza, only the last definition is used unless the attribute is **machine_group**. If you specify multiple **machine_group** attributes, all are applied to the machine definition.
- If you use an invalid attribute keyword, then that attribute definition is ignored.
- Each line of the file can have only one header or attribute definition.
- Only one stanza may exist in a definition file for each machine hostname.
- If the stanza header entry is the keyword **default**, this specifies to use it for the purpose of defining default values.
- You can specify a default value for any machine attribute except the machine hostname. If you do not specify an attribute for a machine but define a default value, then the default value is used.
- You can specify and change default values at any location in the definition file. After a default value is set, it applies to all definitions following it.
- To turn off a default value for all following machine definitions, set the attribute value to **nothing** in a default stanza.
- To turn off a default value for a single machine definition, set the attribute value to **nothing** in the machine stanza.
- You can include comments in a client definition file. Comments begin with the pound (#) character.
- When parsing the definition file for header/attribute keywords and values, tab characters and spaces are ignored.

Client Definition File Keywords

The client definition file uses the following keywords to specify machine attributes:

Required Attributes

Item	Description
cable_type	Specifies the cable type of the machine. Required if network_type is ent .
gateway	Specifies the hostname or IP address of the default gateway used by the machine. If the machine does not use a gateway, then specify the value 0 (zero) for this attribute.
machine_type	Specifies the type of the machine: standalone , diskless , or dataless .
network_type	Specifies the type of the machine's network adapter: ent or tok .
ring_speed	Specifies the ring speed of the machine. Required if network_type is tok .
subnet_mask	Specifies the subnet mask used by the machine.

Optional Attributes

Item	Description
nim_name	Specifies the NIM name to use for a machine. Use this attribute if something other than the hostname is used for the NIM name. By default, the NIM name given to a machine is the hostname of the machine with any domain information stripped off. If you use non-unique hostnames in different domains, a conflict occurs because the same NIM name is used for both machines. In such an environment, define this attribute for the affected machine definitions.
platform	Specifies the machine hardware platform. If you do not specify this attribute, default is rs6k through AIX 5.1 only.
net_adptr_name	Specifies the name of the network adapter used by the machine (tok0 , ent0 , etc.).
netboot_kernel=NetbootKernelType	Specifies the type of kernel to use when booting the client over the network. The netboot_kernel values are up or mp .
ipl_rom_emulation	Specifies the device to use for IPL ROM emulation (/dev/fd0 , /dev/rmt0 , etc.).
primary_interface	Specifies the hostname used for the original machine definition. Use this attribute if the current stanza is only to define an additional interface to a machine that is defined in the NIM environment.
master_gateway	Specifies the gateway that the NIM master uses to reach this machine if this machine is on a different network. This attribute is not necessary if this machine is defined on a network that is already defined in the NIM environment, or if the NIM master network has a default gateway specified.
machine_group	Specifies the group or groups to add the machine to when it is defined.
comments	Specifies a comment to include in the machine definition. The comment string should be in double quotes (").

Client Definition File Stanza Errors

A definition stanza is incorrect under any of the following conditions:

- The hostname used in the stanza header for the definition is unresolvable.
- A required attribute is missing.
- You specify an invalid value for an attribute.
- An attribute mismatch occurs. For example, you can not specify **network_type=tok** and **cable_type=bnc** in the same stanza.
- A group-type mismatch occurs. For example, you can not specify a group for a machine if the group includes standalone machines and you specify **machine_type=diskless**.
- Machine definitions occur multiple times for the same hostname.
- A machine definition occurs for a machine that is already defined in the NIM environment.
- The **primary_interface** value in a machine definition does not match the hostname of any defined machine or stanza definition.
- The **primary_interface** value in a machine definition matches the hostname of another machine definition, but that definition is incorrect.

Sample Client Definition File

These default values are for AIX 5.1 and earlier.

```
# Set default values.
default:
  machine_type = standalone
  subnet_mask  = 255.255.240.0
  gateway      = gateway1
  network_type = tok
  ring_speed   = 16
  platform     = rs6k
  machine_group = all_machines

# Define the machine "lab1"
# Take all defaults.
lab1:

# Define the machine "lab2"
# Take all defaults and specify 2 additional attributes.
# The machine "lab2" uses IPL ROM emulation, and will be added to
# the machine groups "all_machines" and "lab_machines".
lab2:
  ipl_rom_emulation = /dev/fd0
  machine_group     = lab_machines

# Define the machine "lab3"
# Take all defaults, but do not add the machine to the default
# group.
lab3:
  machine_group=

# Define the machine "lab4"
# Take all defaults, but do not add "lab4" to the default group
# "all_machines".
# Instead add it to the groups "lab_machines" and "new_machines".
lab4:
  machine_group =
  machine_group = lab_machines
  machine_group = new_machines

# Change the default "platform" attribute.
default:
  platform = rspc

# define the machine "test1"
# Take all defaults and include a comment.
test1:
  comments = "This machine is a test machine."
```

Flags

Item	Description
-c	Generates commands from a client definition file. This flag processes the definition file and generates the commands to add the definitions. The commands are not invoked but displayed as a KSH script that you can redirect to a file and invoke at a later time.
-d	Defines machines from a client definition file. This flag processes the definition file and invokes the commands to add the definitions to the NIM environment.
-f <i>Name</i>	Specifies the name of the client definition file.

Item	Description
-p	<p>Displays a preview of the client definition file. This flag processes the definition file but does not add machines to the NIM environment. Displays the following:</p> <ul style="list-style-type: none"> All complete and valid NIM definition stanzas. All additional interfaces that will be defined for machines. All invalid definitions stanzas and the reason for failure. All new machine groups and the members to add. All existing machine groups and the members to add. All network definitions to add to the NIM environment. The commands to invoke to add each new machine. The commands to invoke to add each additional machine interface. The commands to invoke to create new machine groups and add their members. The commands to invoke to add new members to existing machine groups. <p>Note: We recommend that you specify the -p flag on a client definition file to verify that all stanzas are correct before using it for adding machines.</p>

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 the **nimdef** command.

Auditing Events: N/A

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 preview the client definition file **client.defs**, enter:

```
nimdef -p -f client.defs
```

2. To add the NIM clients described in the client definition file **client.defs**, enter:

```
nimdef -d -f client.defs
```

3. To create a kshell script called **client.add** to add the NIM clients described in the client definition file **client.defs**, enter:

```
nimdef -c -f client.defs > client.add
```

Files

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

niminit Command

Purpose

Configures the Network Installation Management (NIM) client package.

Syntax

To configure the NIM client package

```
niminit{-a name=Name -a pif_name=Pif -a master=Hostname} [ -a master_port=PortNumber ] [ -a registration_port=PortNumber ] [ -a cable_type=Type | -a ring_speed=Speed ] [ -a iplrom_emu=Device ] [ -a platform=PlatformType ] [ -a netboot_kernel=NetbootKernelType ] [ -a adpt_add=AdapterAddress ] [ -a is_alternate=yes | no ] [ -a connect=value ] [ -a vlan_tag=value ] [ -a vlan_pri=value ] [ -c ]
```

To rebuild the `/etc/niminfo` file

```
niminit {-a name=Name -a master=Hostname -a master_port=PortNumber}
```

Description

The **niminit** command configures the NIM client package. This must be done before the **nimclient** command can be used. When the required attributes are supplied to the **niminit** command, a new machine object will be created to represent the machine where the **niminit** command is being executed. When the **niminit** command completes successfully, the machine will be able to participate in the NIM environment.

After the NIM client package has been successfully configured, the **niminit** command can be run again to rebuild the `/etc/niminfo` on the client. The `/etc/niminfo` file is used by the **nimclient** command and must be rebuilt if it is accidentally removed by a user.

This command configures an `alternate_master` when the `is_alternate` attribute is set to `yes`. The `bos.sysmgt.nim.master` fileset must be installed prior to configuring an `alternate_master`. Once the configuration of an `alternate_master` is successful, the master that it registered with will be able to run `alternate_master` operations on this machine.

Flags

Item	Description	Attribute Description
-a	Specifies up to five different attributes for the niminit command. All of the following attribute=value pairs are preceded by the -a flag:	
name=<i>Name</i>		Specifies the name that NIM will use to identify the workstation. This value is required.
pif_name=<i>Pif</i>		Defines the name of the network interface for all NIM communications. This value is required.

Item	Description	Attribute Description
	master= <i>Hostname</i>	Specifies the hostname of the NIM master. The client must have the ability to resolve this hostname to an Internet Protocol (IP) address. This value is required.
	master_port= <i>PortNumber</i>	Specifies the port number of the nimesis daemon used for NIM communications.
	cable_type= <i>CableType</i>	Specifies the ethernet cable type. When the pif_name refers to an ethernet network, this value must be given. Acceptable values are: bnc , dix , and N/A .
	ring_speed= <i>Speed</i>	Speed in Mbps. When the pif_name refers to a token ring network, this value must be given. Acceptable values are: 4 and 16 .
	iplrom_emu= <i>Device</i>	Specifies a device that contains a ROM emulation image. This image is required for models that do not have internal support for booting via network interface.
	platform= <i>PlatformType</i>	Specifies the platform that corresponds to the client's machine type. If this attribute is not specified, the default, chrp , will be used. The supported platforms are: <ul style="list-style-type: none"> chrp PowerPC Common Hardware Reference Platform (CHRP) architecture-based machines rs6k Micro Channel-based, uniprocessor models for AIX 5.1 and earlier rs6ksmp Micro Channel-based, symmetric multiprocessor models for AIX 5.1 and earlier rspc PowerPC PCI bus-based, uniprocessor machines for AIX 5.1 and earlier rspcsmp PowerPC PCI bus-based, symmetric multiprocessor machines for AIX 5.1 and earlier
	adpt_add= <i>AdapterAddress</i>	Specifies the hardware address that corresponds to the network adapter.
	registration_port= <i>PortNumber</i>	Specifies the port number used for NIM client registration.
		Note: <ol style="list-style-type: none"> 1. If you do not specify port numbers on the command line, the port numbers in the /etc/services file for NIM is used. If the /etc/services file does not contain entries for the NIM ports nim and nimreg, the default values of 1058 for master_port and 1059 for registration_port are used. 2. The values used for master_port and registration_port should match the values used by the NIM master. To display the values used by the NIM master, run the command lsnim -l master on the NIM master.

Item	Description	Attribute Description
	netboot_kernel= <i>NetbootKernelType</i>	Specifies the type of kernel to use when booting the client over the network. The netboot_kernel values are: up Kernel for uniprocessor machines mp Kernel for multiprocessor machines The default is up .
	is_alterate=[yes no]	Set this to yes if this machine is to be configured as an alternate_master .
	connect=value	Specifies the communicating service used by the NIM client for remote execution of NIM commands. Value options are shell (for rsh) and nimsh . The default setting is connect=shell . This attribute is optional. If the is_alterate attribute is set to yes then nimsh is the default setting, and is the only valid value. Using the is_alterate attribute is optional.
	vlan_tag=value	Specifies the virtual logical area network (VLAN) identifier that is used for VLAN tagging. The ID is used to identify the VLAN to which the Ethernet frame must belong. The ID allows the network administrator to organize the client's communication logically rather than assigning the network to the subnet. The VLAN tagging value is used by NIM to perform a network boot of a client. The configuration of the VLAN tag communication must be handled outside of NIM before using the value. Valid values are 0 - 4094.
	vlan_pri=value	Specifies the virtual logical area network (VLAN) priority that is used for VLAN tagging. The priority value, along with the VLAN tag, is used to identify the VLAN to which the Ethernet frame must belong. The priority allows the network administrator to organize the client's communication logically rather than assigning the network to the subnet. The VLAN tagging value is used by NIM to perform a network boot of a client. The configuration of the VLAN tag communication must be handled outside of NIM before using the value. Valid values are 0 - 7.
> > -c	Enables secure sockets layer (SSL) authentication during push operations of NIM master. Note: You must configure Open Secure Sockets Layer (OpenSSL) certificates on the NIM master by using the nimconfig -c command.	NA <<

Security

Access Control: You must have root authority to run the **niminit** 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 configure the NIM client package on a machine that has a BOOTP-enabled IPL ROM such that it will be known as `scuba` in the NIM environment, using `en0` as its primary interface and an Ethernet cable type of `bnc`, and specifying that it communicates with the NIM master using the master's hostname of `manta` and the default NIM ports located in `/etc/services` for network install communications, type:

```
niminit -a name=scuba -a pif_name=en0 -a cable_type=bnc \  
-a master=manta
```

2. To rebuild the `/etc/niminfo` file when it has accidentally been removed by a user, using a hostname of `superman` for the master's hostname and a port number of `1058`, type:

```
niminit -a name=robin -a master=superman -a master_port=1058
```

3. To configure the NIM client package for AIX 5.1 and earlier on a machine that is a PowerPC PCI bus-based, uniprocessor system that has a BOOTP-enabled IPL ROM such that it will be known as `starfish` in the NIM environment, using `en0` as its primary interface and an Ethernet cable type of `dix`, and specifying that it communicates with the NIM master using the master's host name of `whale` and a port number of `1058`, type:

```
niminit -a name=starfish -a pif_name=en0 -a cable_type=dix \  
-a master=whale -a master_port=1058 -a platform=rspc
```

4. To configure the NIM client, on a machine to be known as `bluefish` in the NIM environment, using `at0` as its primary interface and specifying that it communicates with the NIM master using the master's host name `redfish` and a port number of `1058`, type:

```
niminit -a name=bluefish -a pif_name=at0 -a master=redfish \  
-a master_port=1058
```

Note: Because an interface to an ATM network does not currently support booting over the network, this operation will define a machine object on the NIM master if a Generic network object corresponding to the client's subnet is already defined.

5. To configure the NIM client for AIX 5.1 and earlier on a machine that is a PowerPC PCI bus-based, symmetric multiprocessor system that has a BOOTP-enabled IPL ROM such that it will be known as `jellyfish` in the NIM environment, using `en0` as its primary interface and an Ethernet cable type of `dix`, and specifying that it communicates with the NIM master using the master's host name of `whale` and a port number of `1058`, type:

```
niminit -a name=jellyfish -a pif_name=en0 -a cable_type=dix \  
-a master=whale -a master_port=1058 -a platform=rspcsmp
```

6. To configure the NIM client package on a machine that will use an IPL ROM emulation in device `/dev/fd0`, such that it will be known as `octopus` in the NIM environment and uses `tr0` as its primary interface and a ring speed of `16`, and communicates with the NIM master using the master's hostname of `dolphin` and a port number of `1700` for client communications and `1701` for client registration, type:

```
niminit -a iplrom_emu=/dev/fd0 -a name=octopus -a pif_name=tr0 \  
-a ring_speed=16 -a master=dolphin -a master_port=1700 \  
-a registration_port=1701
```

7. To configure this machine as an `alternate_master` with the NIM master `dolphin` and communicate over interface `en0`, type:

```
niminit -a is_alternate=yes -a name=octopus -a pif_name=en0 \  
-a cable_type=bnc -a master=dolphin
```

8. **>** To copy a Secure Sockets Layer (SSL) certificate and to enable SSL authentication during client initialization, enter the following command:

```
niminit -a name=octopus -a master=dolphin -a connect=nimsh -c -v
```

Note: For NIM communication, configuring SSL keys and certificates during NIM master initialization is supported only for AIX 7.3 with Technology Level 2, and later.

[|<](#)

Files

Item	Description
<code>/etc/niminfo</code>	Contains variables used by NIM.

niminv Command

Purpose

Allows system administrators to gather, conglomerate, compare, and download fixes based on installation inventory of NIM objects.

Syntax

To get installation inventory:

```
niminv -o invget -a targets=object1,object2,... [ -a location=path ] [ -a colonsep=yes|no ]
```

To conglomerate installation inventory:

```
niminv -o invcon -a targets=object1,object2,... [ -a base=highest|lowest ] [ -a location=path ] [ -a colonsep=yes|no ]
```

To compare installation inventory:

```
niminv -o invcmp -a targets=object1,object2,... [ -a base=object|any ] [ -a location=path ]
```

To get fixes based on conglomerate inventory:

```
niminv -o fixget -a targets=object1,object2,... [ -a download=yes|no ] [ -a lp_source=object ] [ -a location=path ] -a newlppname=name
```

Description

The `niminv` command (Network Install Manager Inventory) allows system administrators to accomplish the following tasks:

- Gather installation inventory of several NIM objects.
- Conglomerate installation inventory of several NIM objects.
- Compare installation inventory of several NIM objects.
- Download fixes base on the installation inventory of several NIM objects.

The `niminv` command can use any NIM object that contains installation information. Examples of such objects include standalone client objects, SPOT objects, `lp_source` objects and `mksysb` objects.

Using the `niminv` command has the following advantages:

- Hardware installation inventory is gathered alongside the software installation inventory.
- Data files are saved with a naming convention that is easily recognizable.
- All NIM objects that have installation inventory can be used.
- The command provides a holistic view of all managed NIM objects.

The information displayed by `niminv` can be limited by any of the following factors:

- Only software installation inventory is provided for objects that do not actually have physical devices (such as SPOT objects, lpp_source objects, and mksysb objects).
- Software and hardware installation inventory on client objects are limited to what commands on the remote system can provide.
- The recognition of fixes to download is based on the fix backend server. For more details, see **Using the Software Service Management menu (including SUMA)**.

Flags

Item	Description
-a <i>attribute=value</i>	Specifies the attribute and value. The supported attributes and values are based on the operation.
-o <i>operation</i>	<p>Specifies the operation. The following operations are currently supported:</p> <p>fixget Gathers the latest fixes based on the installation inventory. This operation supports the following attributes:</p> <p>targets (required) A comma-separated list of NIM objects to base the gathering of fixes.</p> <p>lpp_source (optional) The NIM lpp_source object to use as a filter for downloading fixes. If the location and newlppname attributes are not used, this lpp_source object will also be where any fixes are downloaded to.</p> <p>location (optional) A directory to store the fixes. Use this attribute only if the fixes should not be downloaded to the object supplied to the lpp_source attribute. This attribute can only be used with the newlppname attribute.</p> <p>newlppname (optional) The NIM object name of the lpp_source to create at location. This attribute can only be used with the location attribute. The value supplied must be distinct and currently unused in the NIM environment.</p> <p>download (optional) Instructs the command whether or not to download the fixes. If no lpp_source or location field is specified and the value of this attribute is yes, fixes will be downloaded to the default location through the suma command.</p> <p>Note: The suma command will increase the file system space according to the MaxFSSize field in the suma configuration.</p>

Item**Description**

-o *operation (Continued)*

invcmp

Compares installation inventory. This operation supports the following attributes:

targets

(required) A comma-separated list of NIM objects to compare installation inventory.

base

(required) The NIM object to use as the comparison base, or the keyword *any*. If the NIM object is supplied, the installation inventory in the object is the sole determinate of the data displayed, and only inventory in the base object is compared against inventory in the target objects. The keyword *any* forces the command to use any installation inventory of the targets.

location

(optional) A directory to store the data files. If this option is used, each inventory is saved with the format *conglomeratebase_object.target_object_list.timestamp*, where *base_object* is the NIM name of the base object used for comparison (or the keyword *any*), *target_object_list* is a colon-separated and sorted list of the NIM name of the objects, and *timestamp* is the time the command was run (*year month day hour minute second*). If the directory does not exist, it will be created. The default is to display the data to the screen.

Item	Description
-o operation <i>(Continued)</i>	<p>invcon Conglomerates installation inventory. This operation supports the following attributes:</p> <p>targets (required) A comma-separated list of NIM objects to conglomerate installation inventory.</p> <p>base (optional) Specifies whether the conglomerate inventory is based on the highest or lowest software levels.</p> <p>location (optional) A directory to store the data files. If this option is used, each inventory is saved with the format <i>base.target_object_list.timestamp</i>, where <i>base</i> indicates whether the conglomerate is based on the highest or lowest levels, <i>target_object_list</i> is a colon-separated and sorted list of the NIM name of the objects, and <i>timestamp</i> is the time that the command was run (<i>year month day hour minute second</i>). If the directory does not exist, it will be created. The default is to display the data to the screen.</p> <p>colonsep (optional) Instructs the command whether or not to produce colon-separated output. The default is no.</p> <p>invget Gathers installation inventory. This operation supports the following attributes:</p> <p>targets (required) A comma-separated list of NIM objects to gather installation inventory.</p> <p>location (optional) A directory to store the data files. If this option is used, each inventory is saved with the format <i>conglomerate.target_object_name.timestamp</i>, where <i>target_object_name</i> is the NIM name of the object, and <i>timestamp</i> is the time that the command was run (<i>year month day hour minute second</i>). If the directory does not exist, it will be created. The default is to display the data to the screen.</p> <p>colonsep (optional) Instructs the command whether or not to produce colon-separated output. The default is no.</p>

Exit Status

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

Security

Access Control: You must have root authority to run the `niminv` command.

Attention RBAC users: The **rbacqry** command grants execute (x) access to the root user. The **rbacqry** command is a privileged command that is used to run privilege operations. You must activate a role that has the authorization to run the command successfully.

Examples

1. To gather installation inventory of a two clients and save the output to /tmp/inventory, enter:

```
niminv -o invget -a targets=client1,client2 -a location=/tmp/inventory
```

Output similar to the following is displayed:

```
Installation Inventory for client1 saved to
/tmp/inventory/inventory.client1.060406140453.
Installation Inventory for client2 saved to
/tmp/inventory/inventory.client2.060406140453.
```

The information in the files is similar to the output of `ls1pp -L`

2. To conglomerate installation inventory of two clients and save the output to /tmp/inventory, enter:

```
niminv -o invcon -a targets=client1,client2 -a location=/tmp/inventory
```

Output similar to the following is displayed:

```
Installation Inventory for client1 saved to
/tmp/inventory/conglomerate.client1:client2.060406140500.
```

The information in the files is similar to the output of `ls1pp -L`.

3. To compare installation inventory of a mksysb, SPOT, and lpp_source to what's currently installed on the master, and save the output to /tmp/inventory, enter:

```
niminv -o invcon -a targets=mksysb1,spot1,lpp_source1 -a base=master -a \
location=/tmp/inventory
```

Output similar to the following is displayed:

```
Installation Inventory for client1 saved to
/tmp/inventory/comparison.master.mksysb1:spot1:lpp_source1.060406140610.
```

The information in the file is listed in column format. The comparison only includes installation inventory on the master.

4. To do the same comparison as in the preceding example but also include software on any of the objects, enter:

```
niminv -o invcon -a targets=mksysb1,spot1,lpp_source1,master -a base=any -a \
location=/tmp/inventory
```

Output similar to the following is displayed:

```
Installation Inventory for client1 saved to
/tmp/inventory/comparison.any.mksysb1:spot1:lpp_source1.060406140733.
```

The information in the file is listed in column format. The comparison includes any installation inventory in any of the target objects.

5. To see the fixes that can be downloaded based on the lowest installations in a mksysb, SPOT and lpp_source, enter:

```
niminv -o fixget -a targets=mksysb1,spot1,lpp_source1
```

Output similar to the following is displayed:

```
*****
Performing preview download.
*****
```

```
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/Java14.debug.1.4.1.0.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/Java14.debug.1.4.1.7.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.00100100.com.5.2.0.50.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.00100300.diag.5.2.0.75.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.00100f00.rte.5.2.0.85.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.13100560.rte.5.2.0.85.bff
```

```
Summary:
  6 downloaded
  0 failed
  0 skipped
```

6. To download the latest fixes based on the lowest installations in a mksysb, SPOT and lpp_source, enter:

```
niminv -o fixget -a targets=mksysb1,spot1,lpp_source1 -a download=yes
```

Output similar to the following is displayed:

```
Extending the /usr filesystem by 30 blocks.
File System size changed to 8126464
```

```
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/Java14.debug.1.4.1.0.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/Java14.debug.1.4.1.7.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.00100100.com.5.2.0.50.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.00100300.diag.5.2.0.75.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.00100f00.rte.5.2.0.85.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.13100560.rte.5.2.0.85.bff
```

```
Summary:
  6 downloaded
  0 failed
  0 skipped
```

Note: Any installations already contained in the default download path (as specified by the `suma` command) will not be downloaded again. The default download path in this example was `/usr/sys/inst.images`. Refer to the [suma](#) command for specifics on where the default download path will be.

7. To download the latest fixes based on the lowest installations in a mksysb, SPOT and lpp_source to an existing lpp_source, enter:

```
niminv -o fixget -a targets=mksysb1,spot1,lpp_source1 -a download=yes -a \
lpp_source=lpp_source2
```

Output similar to the following is displayed:

```
Download SUCCEEDED: /nim/lpps/lpp_source2/installp/ppc/Java14.debug.1.4.1.0.bff
Download SUCCEEDED: /nim/lpps/lpp_source2/installp/ppc/Java14.debug.1.4.1.7.bff
```

```
Summary:
  2 downloaded
  0 failed
  0 skipped
```

Note: Any installations already contained in `lpp_source2` will not be downloaded again. In this example, the `filesets` device already existed in the `lpp_source2`.

8. To download the latest fixes based on the lowest installations in a mksysb, SPOT and lpp_source to a new lpp_source while filtering filesets in an existing lpp_source, enter:

```
niminv -o fixget -a targets=mksysb1,spot1,lpp_source1 -a download=yes -a \
location=/nim/lpps/newlpp1 -a newlppname=newlpp1
```

Output similar to the following is displayed:

```
Download SUCCEEDED: /nim/lpps/newlpp1/installp/ppc/Java14.debug.1.4.1.0.bff
Download SUCCEEDED: /nim/lpps/newlpp1/installp/ppc/Java14.debug.1.4.1.7.bff
```

```
Summary:
  2 downloaded
  0 failed
  0 skipped
```

Note: Any installations already contained in `lpp_source2` will not be downloaded again. In this example, the `filesets` device already existed in the `lpp_source2`.

9. To download the latest fixes based on the lowest installations in a `mksysb`, `SPOT` and `lpp_source` to a new `lpp_source`, enter:

```
niminv -o fixget -a targets=mksysb1,spot1,lpp_source1 -a download=yes -a \
location=/nim/lpps/newlpp2 -a newlppname=newlpp2
```

Output similar to the following is displayed:

```
Download SUCCEEDED: /nim/lpps/newlpp2/installp/ppc/Java14.debug.1.4.1.0.bff
Download SUCCEEDED: /nim/lpps/newlpp2/installp/ppc/Java14.debug.1.4.1.7.bff
Download SUCCEEDED: /nim/lpps/newlpp2/installp/ppc/devices.pci.00100100.com.5.2.0.50.bff
Download SUCCEEDED: /nim/lpps/newlpp2/installp/ppc/devices.pci.00100300.diag.5.2.0.75.bff
Download SUCCEEDED: /nim/lpps/newlpp2/installp/ppc/devices.pci.00100f00.rte.5.2.0.85.bff
Download SUCCEEDED: /nim/lpps/newlpp2/installp/ppc/devices.pci.13100560.rte.5.2.0.85.bff

Summary:
  6 downloaded
  0 failed
  0 skipped
```

Location

`/usr/sbin/niminv`

nimol_backup Command

Purpose

Creates NIMOL install resources from an AIX client.

Note: This command is used only for Virtual I/O Server (VIOS) or Integrated Virtualization Management (IVM) from the Hardware Management Console (HMC).

Syntax

```
nimol_backup -c client_hostname [-t directory] [-m remote_access_method] [-L label] [-D]
```

Description

The **nimol_backup** command creates NIMOL install resources from a configured NIMOL client using the specified remote access method, which is `/usr/bin/rsh` by default, to call the `nimol_mk_resources` method on the client. When configuring a NIMOL server using the **nimol_config** command, the user can set the default remote access method to something other than `/usr/bin/rsh`, such as `/usr/bin/ssh`. A machine is considered a NIMOL client when it has been installed using the **nimol_install** command without the **-n** flag.

The command creates the target directory and label on the NIMOL server. The directory is then exported. The default label is `default`. For example, if the command is passed **-t /export/aix -L aix530**, then the command creates the `/export/aix/aix530` directory on the NIMOL server.

The command then uses the remote access method to run the **nimol_mk_resources** command. The **nimol_mk_resources** command creates the necessary install resources in the target directory.

Flags

Item	Description
-c <i>client_hostname</i>	Specifies the NIMOL client hostname on which to execute the geninstall command.

Item	Description
-D	Runs the command in debug mode.
-L label	Specifies the label or name to create for the created resources.
-m remote_access_method	Specifies the remote access method to use to run the geninstall command. The default /usr/bin/rsh . Another option is /usr/bin/ssh .
-t directory	Specifies the target directory where the AIX install resources will be created from the NIMOL client. The default directory is /export/aix .

Exit Status

Item	Description
0	The command completed successfully.
> 0	Error returned.

Security

To run the **nimol_backup** command on a NIMOL client, the client must provide remote access permissions to the NIMOL server. Using **/usr/bin/ssh** is a more secure remote access method than **/usr/bin/rsh**.

Examples

1. To create install resources from client **myclient** in the **/export/aix** directory and named **530**, type:

```
nimol_backup -c myclient -L 530 -t /export/aix
```

2. To execute **nimol_mk_resources** using **ssh**, type:

```
nimol_backup -c myclient -m ssh
```

Location

/usr/sbin/nimol_backup

Files

Item	Description
/etc/nimol.conf	Stores configuration information for the command.

nimol_config Command

Purpose

Configures a Linux server to network install a machine with AIX by configuring services and copying install resources.

Note: This command is used only for Virtual I/O Server (VIOS) or Integrated Virtualization Management (IVM) from the Hardware Management Console (HMC).

Syntax

nimol_config [-d *DirectoryContainingAIXResources*] [-t *TargetDirectoryToCopyResources*] [-L *InstallResourcesLabel*] [-s *NIMOLServerHostname*] [-m *RemoteAccessMethod*] [-C] [-e] [-l] [-r] [-S] [-U] [-D]

Description

The **nimol_config** command configures a Linux server to network install a machine with AIX. The command performs the following configuration.

1. First, the command obtains the hostname and IP address of the Linux server. If no hostname is specified with the **-s** flag, the command uses the hostname of the local machine and the IP address associated with the hostname. If a hostname and IP address are specified, then the pair is added to the **/etc/hosts** file, if it does not already exist.
2. The command then starts the portmap service and nfs server.
3. The command stores the remote access method in the **/etc/nimol.conf** file if specified with the **-m** flag. The default remote access command is **/usr/bin/rsh**, which is used to communicate with NIMOL clients that have been installed without specifying the **-n** flag to the **nimol_install** command.
4. Next, tftpboot is configured. The **/tftpboot** directory is created if it does not exist and the **/etc/xinetd.d/tftp** file is created if it does not exist. Then the command sets `disable` equal to `no` in the **/etc/xinetd.d/tftp** file and restarts xinetd so that the tftp server can handle incoming requests.
5. The **nimol_config** command also sets up syslog to accept incoming messages from other machines. Clients pass back status while installing to the syslog server. The **/etc/sysconfig/syslog** file is modified to include the **-r** flag in the `SYSLOGD_OPTIONS` or `SYSLOGD_PARAMS` variable. Then the command searches **/etc/syslog.conf** for the first available local log and sets it to write messages to **/var/log/nimol.log**. Clients write status to this log file, which can be monitored during a client installation. After the changes are made to the syslog configuration files, the service is restarted.
6. Next, the command sets up the DHCP server to receive bootp requests from AIX clients. The subnet of the NIMOL server is determined and added to the **dhcpd.conf** file. The options `allow bootp`, `not authoritative`, and `ddns-update-style none` are added if they do not already exist. Existing settings for these options will be overwritten.
7. Once the services have been configured, the **nimol_config** command attempts to copy AIX install resources locally, if the **-C** flag was not passed to the command. The command copies resources from the source directory specified with the **-d** flag (**/mnt/cdrom** by default) to the target directory (**/export/aix** by default). A directory is created (name that matches the LABEL name specified with the **-L** flag 'default' by default). The command looks in the source directory for the following resources:
 - a SPOT (Source Product Object Tree) directory named **/SPOT** and a SPOT directory named **ispot.tar.Z**
 - an `lpp_source` directory named **/lpp_source**
 - a `mksysb` named **mksysb** or **mksysb.bff**
 - a boot image named **booti.chrp.mp.ent**
 - a `bosinst.data` file named **bosinst.data**
 - an `image.data` file named **image.data**
 - a customization script named **cust.script**
 - a `resolv.conf` file named **resolv.conf**A SPOT, boot image, and either `mksysb` or `lpp_source` are required.
8. The target directory is then globally exported unless the **-e** flag is specified.
9. If a target directory and label are specified that contain resources, then these resources will be used and no resources will be copied. For example, if the command is passed **-t /export/aix -L aix530** and the directory **/export/aix/aix530** contains resource, then the command will not attempt to copy resources from the source directory.

10. After the NIMOL server has been configured, the **nimol_config** command will not attempt to reconfigure services on the NIMOL server when defining new resource labels.
11. The command also lists defined resource labels with the **-l** flag.
12. Resource labels can be removed by specifying the **-r** flag with a resource label. The command unexports the directory, if exported, and deletes the directory of the resource label.
13. When the **-U** flag is passed, the command attempts to undo any configuration that it has done, such as unconfiguring services.

Flags

Item	Description
-C	Specifies that the server should only configure services without copying install resources.
-d <i>directory</i>	Specifies the source directory that contains the AIX install resources. The default directory is /mnt/cdrom .
-D	Runs the command in debug mode.
-e	Instructs the command not to globally export the directory of newly created resource label.
-l	Lists the defined resource labels available to install a client.
-L <i>label</i>	Specifies the label or name to create for the copied resources.
-m <i>method</i>	Specifies the remote access method to use when running commands on clients that have been installed without specifying the -n flag to the nimol_install command.
-r	Instructs the command to remove the specified resource label.
-s <i>hostname</i>	The hostname to use for the NIMOL server. The default is to determine the hostname by running the hostname command.
-S	Instructs the command to not configure the syslog service. No status will be logged when clients are installing.
-t <i>directory</i>	Specifies the target directory where the AIX install resources will be copied from the source directory. The default directory is /export/aix .
-U	Instructs the command to unconfigure the NIMOL server. The command will attempt to undo any configuration that it performed.

Exit Status

Item	Description
0	The command completed successfully.
> 0	Error returned.

Security

Configuring the syslog service to accept messages from remote clients can be a security issue. Configure your firewall to only accept syslog messages from known clients.

Examples

1. To configure the NIMOL server without copying resources, type:

```
nimol_config -C
```


2. To configure the NIMOL server, copy resources from **/mnt/aix** to **/export/aix**, and label the resource aix530, type:

```
nimol_config -d /mnt/aix -t /export/aix -L aix530
```

3. To configure the NIMOL server and copy resources without configuring syslog and without globally exporting the resource label directory, type:

```
nimol_config -S -e
```

4. To list defined resource labels, type:

```
nimol_config -l
```

5. To remove the aix530 resource label, type:

```
nimol_config -L aix530 -r
```

Location

/usr/sbin/nimol_config

Files

Item	Description
/etc/nimol.conf	Stores configuration information for the command.

nimol_install Command

Purpose

Sets up a configured NIMOL server to install AIX to a specific client machine.

Note: This command is used only for Virtual I/O Server (VIOS) or Integrated Virtualization Management (IVM) from the Hardware Management Console (HMC).

Syntax

```
nimol_install -c client_hostname [ -g gateway ] [ -m mac_address ] [ -p ip_address ] [ -s subnet_mask ] [ -L label ] [ -n ] [ -r ] [ -D ]
```

Description

The **nimol_install** command sets up a configured NIMOL server to network install a machine with AIX. The command performs the following configuration.

1. The command determines the IP address of the client hostname if the client IP address isn't specified. If the client hostname isn't resolvable and a client IP address is specified, then the pair will be added to the **/etc/hosts** file if it does not exist.
2. The client is added to the **/etc/nimol.conf** file.
3. The directory of the resource label is exported to the client if it is not already globally exported.
4. A stanza for the client is added to the **/etc/dhcpd.conf** file. The client's subnet will also be added to the **/etc/dhcpd.conf** file if it does not exist. If the client or its subnet already exist in the **/etc/dhcpd.conf** file, an error is displayed.
5. A symbolic link to the boot image is created in the **/tftpboot** directory for the client.
6. A static arp entry is added if the client is on the same subnet as the NIMOL server.

7. The command will turn off the firewall rules to a client that is installing if the **iptables** command exists by running:

```
iptables -I INPUT 1 -s client_hostname -j ACCEPT
```

This allows the various services used by NIMOL to succeed. When a client is removed, the **nimol_install** command will run the following command to delete the rule: `iptables -D INPUT -s client_hostname`.

8. The command ensures that the required resources exist in the resource label's directory.
9. A `nim_script` is created in the scripts subdirectory of the resource label's directory if a **resolv.conf** or customization script was specified or if the client will remain a client of the NIMOL server after the installation. The **nimol_install** command will look for a general customization script in the resource label's directory named `cust.script` or a specific customization script for the client named `client_name.script`.
10. An information file is created in the **/tftpboot** directory that will be used during the installation of the operating system.
11. If the **-l** flag is specified, the command will list clients set up for an installation. A client will be removed if the **-r** flag is specified with a client name.
12. Once the client has been set up to install, the client must be told to perform a network install. If the client has AIX installed and is running, then use the **bootlist** command. For example, if the NIMOL server is 192.168.1.20 and the AIX client is 192.168.1.30, then to boot off `ent0` run:

```
bootlist -m normal -ent0 bserver=192.168.1.20 \<\  
gateway=0.0.0.0 client=192.168.1.30
```

then reboot by running:

```
shutdown -Fr
```

13. If the client is not running, then boot into the SMS menus and specify the network boot parameters and the network boot device. If the client is on the same subnet as the NIMOL server, then the client will be able to do a broadcast **bootp** install. A broadcast **bootp** does not require the IP parameters to be set; the `bserver`, `gateway` and `client` would be `0.0.0.0` on a broadcast `bootp` install.

Flags

Item	Description
-c <i>client_hostname</i>	Specifies the client hostname that will be set up for an install or will be removed.
-D	Runs the command in debug mode.
-g <i>gateway</i>	Specifies the gateway that will be configured after the client has installed AIX. This is required when installing a client.
-l	Lists the clients set up to install.
-L <i>label</i>	Specifies the label or name of resources with which to install the client. The default is <code>default</code> .
-m <i>mac_address</i>	Specifies the MAC address of the network interface the client will install over. This is required when installing a client. The MAC address must contain colons (for example <code>00:60:08:3F:E8:DF</code>).
-n	Specifies not to configure the machine to remain a client of the NIMOL server after the installation has completed. If this option is specified, the client will not have its network configured after the installation.
-p <i>ip_address</i>	Specifies the IP address of the client. Use this flag if the client's hostname is not resolvable.

Item	Description
-r	Removes the client. The client will not be able to install AIX until it is reconfigured. This flag requires a client hostname.
-s <i>subnet_mask</i>	Specifies the subnet mask of the client interface. This flag is required when installing a client.

Exit Status

Item	Description
0	The command completed successfully.
> 0	Error returned.

Security

If the machine remains a client of the NIMOL server (the **-n** flag is not specified), then it will give the NIMOL server **/usr/bin/rsh** permissions so it can run commands on the client.

Examples

1. To setup client `myclient` to install the `aix530` resource label with gateway `192.168.1.1`, MAC address `00:60:08:3F:E8:DF`, and subnet mask `255.255.255.0`, type:

```
nimol_install -c myclient -g 192.168.1.1 \\  
-m 00:60:08:3F:E8:DF -s 255.255.255.0 -L aix530
```

2. To setup client `myclient` and not have it remain a client to the NIMOL server after the installation, type:

```
nimol_install -n -c myclient -g 192.168.1.1 \\  
-m 00:60:08:3F:E8:DF -s 255.255.255.0 -L aix530
```

3. To list the clients configured to be installed, type:

```
nimol_install -l
```

4. To remove client `myclient`, type:

```
nimol_config -c myclient -r
```

Location

/usr/sbin/nimol_install

Files

Item	Description
/etc/nimol.conf	Stores configuration information for the command.

nimol_lslpp Command

Purpose

Runs the **lslpp** command on a NIMOL client.

Note: This command is used only for Virtual I/O Server (VIOS) or Integrated Virtualization Management (IVM) from the Hardware Management Console (HMC).

Syntax

```
nimol_lslpp -c client_hostname [ -m remote_access_method ] [ -f lslpp_flags ] [ -D ]
```

Description

The **nimol_lslpp** command executes the **lslpp** command on a configured NIMOL client using the specified remote access method, which is **/usr/bin/rsh** by default. When configuring a NIMOL server using the **nimol_config** command, the user can set the default remote access method to something other than **/usr/bin/rsh**, such as **/usr/bin/ssh**. A machine is considered a NIMOL client when it has been installed using the **nimol_install** command without the **-n** flag.

The command runs the **lslpp** command with **-L -c** as the default flags. The **lslpp** command flags can be specified with the **-f** flag.

Flags

Item	Description
-c <i>client_hostname</i>	Specifies the NIMOL client hostname on which to execute the lslpp command.
-D	Runs the command in debug mode.
-f <i>lslpp_flags</i>	Specifies the lslpp command flags to pass to the lslpp command.
-m <i>remote_access_method</i>	Specifies the remote access method to use to run the lslpp command. The default is /usr/bin/rsh . Another option is /usr/bin/ssh .

Exit Status

Item	Description
0	The command completed successfully.
> 0	Error returned.

Security

To run the **nimol_lslpp** command on a NIMOL client, the client must provide remote access permissions to the NIMOL server. Using **/usr/bin/ssh** is a more secure remote access method than **/usr/bin/rsh**.

Examples

1. To run the **lslpp** command on client **myclient**, with the default flags **-Lc**, type:

```
nimol_lslpp -c myclient
```

2. To run the **lslpp** command on client **myclient**, with the flags **-i bos.rte**, type:

```
nimol_lslpp -c myclient -f "-i bos.rte"
```

3. To run the **lslpp** command on client **myclient**, using **ssh** as the remote access method, type:

```
nimol_lslpp -c myclient -m ssh
```

Location

/usr/sbin/nimol_lslpp

Files

Item	Description
<code>/etc/nimol.conf</code>	Stores configuration information for the command.

nimol_update Command

Purpose

Runs `geninstall` on a NIMOL client to perform software maintenance.

Note: This command is used only for Virtual I/O Server (VIOS) or Integrated Virtualization Management (IVM) from the Hardware Management Console (HMC).

Syntax

```
nimol_update -c client_hostname [ -L label ] [ -f geninstall_flags ] [ -m remote_access_method ] [ -p package_list ] [ -D ]
```

Description

The `nimol_update` command executes the `geninstall` command on a configured NIMOL client using the specified remote access method, which is `/usr/bin/rsh` by default. When configuring a NIMOL server using the `nimol_config` command, the user can set the default remote access method to something other than `/usr/bin/rsh`, such as `/usr/bin/ssh`. A machine is considered a NIMOL client when it has been installed using the `nimol_install` command without the `-n` flag.

The command runs the `geninstall` command with `-acgX` as the default flags. Use the `-f` flag to specify `geninstall` command flags. The software packages to pass the `geninstall` command are specified with the `-p` flag.

When installing filesets using the `nimol_update` command, you must specify a resource label that has an `lpp_source`. Run `nimol_config -l -L label` to determine if a resource label contains an `lpp_source`. The command will export the resource label directory if it is not already globally exported. The client will mount the directory and use it as the source directory during an installation.

Flags

Item	Description
<code>-c client_hostname</code>	Specifies the NIMOL client hostname on which to execute the <code>geninstall</code> command.
<code>-D</code>	Runs the command in debug mode.
<code>-f geninstall_flags</code>	Specifies the flags to pass to the <code>geninstall</code> command. The default flags are <code>-acgX</code> .
<code>-L label</code>	Specifies the name of the resource label that will be used as the source for install images.
<code>-m remote_access_method</code>	Specifies the remote access method to use to run the <code>geninstall</code> command. The default is <code>/usr/bin/rsh</code> . Another option is <code>/usr/bin/ssh</code> .
<code>-p package_list</code>	Specifies the name of software packages to pass to the <code>geninstall</code> command. The default is <code>all</code> .

Exit Status

Item	Description
0	The command completed successfully.
> 0	Error returned.

Security

To run the **nimol_update** command on a NIMOL client, the client must provide remote access permissions to the NIMOL server. Using **/usr/bin/ssh** is a more secure remote access method than **/usr/bin/rsh**.

Examples

1. To install all packages in resource label 530 to client `myclient`, type:

```
nimol_update -c myclient -L 530
```

2. To apply an update for `bos.games` on client `myclient`, type:

```
nimol_update -c myclient -L 530 -f "-a" -p "bos.games"
```

3. To remove `bos.games` from client `myclient`, type:

```
nimol_update -c myclient -f "-u" -p "bos.games"
```

4. To execute the **geninstall** command using `ssh`, type:

```
nimol_update -c myclient -L 530 -m ssh
```

Location

/usr/sbin/nimol_update

Files

Item	Description
/etc/nimol.conf	Stores configuration information for the command.

nimquery Command

Purpose

Query a system in the Network installation management (NIM) environment for system information and create client objects in the environment.

Syntax

```
nimquery{ -a host=hostname [-a name=client obj [-d]] [-a hmc=obj name [-d]] [-a cec=obj name [-a bcmm=obj name [-a ivm=obj name]] [-p] [-q] [-v]}
```

Description

The `nimquery` command queries a machine for system information when using the `-a host` parameter. The information is used for defining a new client object in the NIM environment. System information is provided from systems that use the NIM service handler (`nimsh`).

The `nimquery` command can also be used to query logical partitions (LPARs), central electronics complex (CEC) and blades information when pointing to a Hardware Management Console (HMC), CEC, Integrated Virtualization Manager (IVM) or Blade Center Management Module (BCMM) object. To do so, the **openssh.base.client** must be installed on the NIM master.

Flags

Item	Description
-a	Assigns the following parameter attribute=value pairs.
-d	Defines a new client object (requires the name attribute when -a host is used).
-p	Enables print format.
-q	Shows an attribute list for <code>nimquery</code> command.
-v	Enables verbose debug output during command execution.

Parameters

Item	Description
<code>host=hostname</code>	Specifies the host name of the system to query. This attribute is required.
<code>name=client_obj</code>	Specifies the name to assign the client object when creating a new definition in the NIM database.
<code>hmc=objname</code>	Specifies the object name of the HMC system to query. This attribute is required.
<code>cec=objname</code>	Specifies the object name of the CEC system to query. This attribute is required.
<code>ivm=objname</code>	Specifies the object name of the IVM system to query. This attribute is required.
<code>bcmm=objname</code>	Specifies the object name of the BCMM system to query. This attribute is required.

Exit Status

- 0**
Returns zero on success.

Security

You must have root authority to run the `nimquery` command.

Examples

1. To query machine buckey for system information, type:

```
nimquery -a host=buckey
```

2. To query machine buckey for system information and to provide detailed output information, type:

```
nimquery -a host=buckey -p
```

3. To define machine buckey.austin.ibm.com using name client6 as the NIM object name, type:

```
nimquery -a name=client6 -a host=buckey -d
```

4. To query Management Module bcmm2 for blade system information, type:

```
nimquery -a bcmm=bcmm2
```

5. To define CEC objects managed by HMC hmc1, type:

```
nimquery -a hmc=hmc1 -d
```

6. To query LPARs attached to cec1 buckey for system information, type:

```
nimquery -a cec=cec1
```

Files

Item	Description
/usr/sbin/nimquery	The location of nimquery command.

nistoldif Command

Purpose

Exports user, group, name resolution, and rpc data to rfc 2307-compliant form.

Syntax

```
nistoldif -d Suffix [ -a BindDN -h Host -p Password [-n Port] ] [ -f Directory ] [ -y domain ] [ -S Schema ] [ -k KeyPath -w SSLPassword ] [ -s Maps ] [ -m ldap_mapname ]
```

Description

The **nistoldif** command converts the data from **passwd**, **group**, **hosts**, **services**, **protocols**, **rpc**, **networks**, **netgroup**, and **automount** into forms compliant with rfc2307. It will first attempt to read data from NIS, and if it cannot find a NIS map it will fall back to the flat files.

If the server information (the **-a**, **-h**, and **-p** flags) is given on the command line, data will be written directly to the server. If any data conflicts with an entry already on the server, either because the entry already exists, or because the **uid** or **gid** already exists, a warning will be printed. If the server information is not given, the data will be written to **stdout** in LDIF. In either case, **nistoldif** does not add an entry for the suffix itself; if that entry does not exist, attempts to add data to the server will fail. This entry will be added during server setup, usually by the **mksecdap** command.

Translation is not exact. Because of the limitations of the rfc2307 definitions, some attributes are defined in a case-insensitive way; for example, TCP, Tcp, and tcp are all the same protocol name to the LDAP server. Uids and gids greater than $2^{31}-1$ will be translated to their negative twos complement equivalent for storage.

The **nistoldif** command reads the **/etc/security/ldap/sectoldif.cfg** file to determine what to name the sub-trees that the **passwd**, **group**, **hosts**, **services**, **protocols**, **rpc**, **networks** and **netgroup** data will be exported to. The names specified in the file will be used to create sub-trees under the base DN specified with the **-d** flag. For more information, see the **/etc/security/ldap/sectoldif.cfg** file documentation.

Flags

Item	Description
-a	Specifies the administrative bind DN used to connect to the LDAP server. If this flag is used, -h and -p must also be used, and data will be written directly to the LDAP server.
-d	Specifies the suffix that the data should be added under.

Item	Description
-f	Specifies the directory to look for flat files in, or the name of the automount map file. If this flag is not used, nistoldif will look for files in /etc . This flag is required for automount maps.
-h	Specifies the host name which is running the LDAP server. If this flag is used, -a and -p must also be used, and data will be written directly to the LDAP server. This flag will be ignored for automount data.
-k	Specifies the SSL key path. If this flag is used, -w must also be used.
-m	Specifies the automount map on the LDAP server.
-n	Specifies the port to connect to the LDAP server on. If this flag is used, -a , -h and -p must also be used; if it is not used, the default LDAP port is used.
-p	Specifies the password used to connect to the LDAP server. If this flag is used, -a and -h must also be used, and data will be written directly to the LDAP server.
-s	Specifies a set of maps to be written to the server. This flag should be followed by a list of letters representing the maps that should be migrated. If this flag is not used, all maps will be migrated. The letters are: a for automount, e for netgroup, g for group, h for hosts, n for networks, p for protocols, r for rpc, s for services, and u for passwd.
-S	Specifies the LDAP schema to use for users and groups. This can be either RFC2307 or RFC2307AIX; RFC2307AIX gives extended AIX schema support. If this flag is not used, RFC2307 is the default.
-w	Specifies the SSL password. If this flag is used, -k must also be used.
-y	Specifies the NIS domain to read maps from. If this flag is not used, the default domain will be used.

Exit Status

This command returns the following exit values:

0

No errors occurred. Note that failure to find a map is not considered an error.

>0

An error occurred.

Security

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

Examples

1. To export the NIS maps from the domain **austin.ibm.com** (falling back to the flat files in **/tmp/etc**) to LDIF under the suffix **cn=aixdata**, type:

```
nistoldif -d cn=aixdata -y austin.ibm.com -f /tmp/etc > ldif.out
```

2. To export the hosts and services maps from the default domain (falling back to the flat files in **/etc**) to the LDAP server **ldap.austin.ibm.com** with administrator bind DN **cn=root** and password **secret** under the suffix **cn=aixdata**, type:

```
nistoldif -d cn=aixdata -h ldap.austin.ibm.com -a cn=root -p secret -s hs
```

3. To convert the **/etc/auto_master** automount map file into LDIF, type:

```
nistoldif -s a -f /etc/auto_master > ldif.out
```

4. In order to remove automount data, the LDIF file must be created manually. For example, suppose the user `user1` was erroneously added to the `auto_home` automount map in the `dc=austin,dc=ibm,dc=com` suffix, and needs to be deleted. Create the following LDIF:

```
# cat /tmp/del_user1.ldif
dn: automountKey=user1,automountMapName=auto_home,dc=austin,dc=ibm,dc=com
changetype: delete
```

Then run the following command:

```
ldapmodify -f /tmp/del_user1.ldif
```

5. In order to edit automount data, the LDIF file must be created manually. For example, suppose the user `user2` was given the wrong mount point in the `auto_home` automount map in the `dc=austin,dc=ibm,dc=com` suffix, and needs to be changed to the correct location of `/home/user2`. Create the following LDIF:

```
# cat /tmp/ch_user2.ldif
dn: automountKey=user2,automountMapName=auto_home,dc=austin,dc=ibm,dc=com
changetype: modify
replace: automountInformation
automountInformation: /home/user2
```

The run the following command:

```
ldapmodify -f /tmp/ch_user2.ldif
```

Files

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

nl Command

Purpose

Numbers lines in a file.

Syntax

```
nl [ -b Type ] [ -f Type ] [ -h Type ] [ -l Number ] [ -d Delimiter ] [ -i Number ] [ -n Format ] [ -v Number ] [ -w Number ] [ -p ] [ -s Separator ] [ File ]
```

Description

The `nl` command reads the *File* parameter (standard input by default), numbers the lines in the input, and writes the numbered lines to standard output. In the output, the `nl` command numbers the lines on the left according to the flags you specify on the command line.

The input text must be written in logical pages. Each logical page has a header, a body, and a footer section (you can have empty sections). Unless you use the `-p` flag, the `nl` command resets the line numbers at the start of each logical page. You can set line-numbering flags independently for the header, body, and footer sections (for example, the header and footer lines can be numbered while the text lines are not).

Signal the start of logical-page sections with lines in the file that contain only the following delimiter characters:

Line Contents	Start Of
<code>\:\:</code>	Header

Line Contents	Start Of
\:\ \\:	Body
\\:	Footer

You can name only one file on the command line. You can list the flags and the file name in any order.

Flags

All the parameters are set by default. Use the following flags to change these default settings. Except for the **-s** flag, enter a **-n** flag without a variable to see its default value.

Item	Description
-b <i>Type</i>	Chooses which body section lines to number. Recognized values for the <i>Type</i> variable are: a Numbers all lines t Does not number lines that are blank or lines that contain any non-graphic character such as a tab within them. (default) n Does not number any lines pPattern Numbers only those lines specified by the <i>Pattern</i> variable.
-d <i>Delimiter</i>	Uses the two characters specified by the <i>Delimiter</i> variable as the delimiters for the start of a logical page section. The default characters are \: (backslash, colon). You may specify two ASCII characters, two 1-byte extended characters, or one extended character. If you enter only one 1-byte character after the -d flag, the second character remains the default (a colon). If you want to use a backslash as a delimiter, enter two backslashes (\\).
-f <i>Type</i>	Chooses which logical-page footer lines to number. The possible values for the <i>Type</i> variable are the same as the -b flag. The default value of the <i>Type</i> variable is n (no lines numbered).
-h <i>Type</i>	Chooses which logical-page header lines to number. The possible values for the <i>Type</i> variable are the same as the -b flag. The default value of the <i>Type</i> variables n (no lines numbered).
-i <i>Number</i>	Increments logical-page line numbers by the number specified in the <i>Number</i> variable. The default value of the <i>Number</i> variable is 1. The range of the <i>Number</i> variable is from 1 to 250.
-l <i>Number</i>	(Lowercase L) Uses the value specified in the <i>Number</i> parameter as the number of blank lines to count as one. For example, -13 numbers every third blank line in a series. The default value of the <i>Number</i> variable is 1. This flag works when the -ha , -ba , or -fa option is set. The range of the <i>Number</i> variable is from 1 to 250.
-n <i>Format</i>	Uses the value of the <i>Format</i> variable as the line numbering format. Recognized formats are: ln Left-justified, leading zeros suppressed rn Right-justified, leading zeros suppressed (default) rz Right-justified, leading zeros kept

Item	Description
-p	Does not restart numbering at logical page delimiters.
-s Separator	Separates the text from its line number with the character specified in the <i>Separator</i> variable. The default value of the <i>Separator</i> variable is a tab character.
-v Number	Sets the initial logical-page line number to the value specified by the <i>Number</i> variable. The default value of the <i>Number</i> variable is 1. The range of the <i>Number</i> variable is from 0 to 32767.
-w Number	Uses the value specified by the <i>Number</i> variable as the number of characters in the line number. The default value of the <i>Number</i> variable is 6. The range of the <i>Number</i> variable is from 1 to 20.

Exit Status

This command returns the following exit values:

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

Examples

1. To number only the non-blank lines, enter:

```
n1 chap1
```

This displays a numbered listing of chap1, numbering only the non-blank lines in the body sections. If chap1 contains no \:\:\+, or \: delimiters, then the entire file is considered the body.

2. To number all lines:

```
n1 -ba chap1
```

This numbers all the lines in the body sections, including blank lines. This form of the **nl** command is adequate for most uses.

3. To specify a different line number format, enter:

```
n1 -i10 -nrz -s:: -v10 -w4 chap1
```

This numbers the lines of chap1 starting with ten (-v10) and counting by tens (-i10). It displays four digits for each number (-w4), including leading zeros (-nrz). The line numbers are separated from the text by two colons (-s : :).

For example, if chap1 contains the text:

```
A not-so-important
note to remember:

You can't kill time
without injuring eternity.
```

then the numbered listing is:

```
0010::A not-so-important
0020::note to remember

0030::You can't kill time
0040::without injuring eternity.
```

Note that the blank line was not numbered. To do this, use the **-ba** flag as shown in example 2.

Files

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

nlsrc Command

Purpose

Gets the status of a subsystem or a group of subsystems in canonical form.

Syntax

`nlsrc [-h host] -a`

`nlsrc [-h host] -g group_name`

`nlsrc [-h host] [-l] [-c] -s subsystem_name`

`nlsrc [-h host] [-l] [-c] -p subsystem_pid`

The syntax for the first two usages of `nlsrc` will generate the exact same output as `lsrc`. The syntax for the last two usages will generate the output in the canonical form as `lsrc`.

Description

Use the `nlsrc` command to get the status of a subsystem or a group of subsystems in canonical form. For the AIX platform, use the `nlsrc -c` command to get language-independent output for supported subsystems from the `lsrc` command. The status is displayed in English regardless of the installed language locale. If the `-c` flag is not present, the `nlsrc` command will invoke the `lsrc` command that uses the daemon's locale.

Flags

Item	Description
<code>-a</code>	Lists the current status of all defined subsystems
<code>-c</code>	Requests the canonical <code>lsrc</code> output of the supported subsystems.
<code>-g <i>group_name</i></code>	Specifies a group of subsystems to get status for. The command is unsuccessful if the <i>group_name</i> parameter is not contained in the subsystem object class.
<code>-h <i>host</i></code>	Specifies the foreign host on which this status action is requested. The local user must be running as root. The remote system must be configured to accept remote System Resource Controller (SRC) requests. That is, the <code>srcmstr</code> daemon (see <code>/etc/inittab</code>) must be started with the <code>-r</code> flag and the <code>/etc/hosts.equiv</code> file or the <code>.rhosts</code> file must be configured to allow remote requests.

Item	Description
-l	Requests that a subsystem send its current status in long form. Long status requires that a status request be sent to the subsystem; it is the responsibility of the subsystem to return the status.
-p <i>subsystem_pid</i>	Specifies a particular instance of the <i>subsystem_pid</i> parameter to get status for, or a particular instance of the subsystem to which the status subserver request is to be taken.
-s <i>subsystem_name</i>	Specifies a subsystem to get status for. The <i>subsystem_name</i> parameter can be the actual subsystem name or the synonym name for the subsystem. The command is unsuccessful if the <i>subsystem_name</i> parameter is not contained in the subsystem object class.

Security

You do *not* need root authority to run this command.

Exit Status

- 0** Command has run successfully.
- 1** Command was not successful.

Restrictions

This command applies to the cthags and cthats subsystems only.

Standard Output

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

Standard Error

Error messages are written to standard error (and to the `ctsnap.host_name.nnnnnnnn.log` file).

Examples

- To get nlsrc output in English from a subsystem called ctsubsys, enter:

```
nlsrc -c -ls ctsubsys
```

- The following example shows the same information in different formats:

```
nlsrc -ls ctsubsys (locale-dependent)
Subsystem Group      PID  Status
ctsubsys  ctsubsys  6334 active
2 locally-connected clients. Their PIDs:
15614 23248
HA Subsystem domain information:
Domain established by node 5
Number of groups known locally: 1
Group Name      Number of      Number of local
ha_filesys      providers      providers/subscribers
7                1                0
```

```
nlssrc -ls ctsubsys -c (canonical form)
```

```
Number of local clients: 2
PIDs: 15614 23248
HA Subsystem domain information:
Domain established by node 5.
Number of known local groups: 1
Group Name: ha_filesys
  Providers: 7
  Local Providers: 1
  Local Subscribers: 0
```

Location

/opt/rsct/bin/nlssrc

Contains the `nlssrc` command

Files

/tmp/ctsupt

Location of the default directory that contains the output files.

/tmp/ctsupt/ctsnap.*host_name*.nnnnnnnn.log

Location of the log file of the command execution, where *nnnnnnnn* is a timestamp and *host_name* is the name of the host on which the command is running.

tmp/ctsupt/ctsnap.*host_name*.nnnnnnnn.tar.Z

Location of the compressed tar file that contains the collected data, where *nnnnnnnn* is a timestamp and *host_name* is the name of the host on which the command is running.

nm Command

Purpose

Displays information about symbols in object files, executable files, and object-file libraries.

Syntax

```
nm [ -A ] [ -C ] [ -X { 32|64|32_64|d64|any } ] [ -f ] [ -h ] [ -l ] [ -p ] [ -r ] [ -T ] [ -v ] [ -B | -P ] [ -e | -g | -u ] [ -d ] [ -o | -x | -t Format ] File ...
```

Description

The **nm** command displays information about symbols in the specified *File*, which can be an object file, an executable file, or an object-file library. If the file contains no symbol information, the **nm** command reports the fact, but does not interpret it as an error condition. The **nm** command reports numerical values in decimal notation by default.

The **nm** command writes the following symbol information to standard output:

- **Library** or **Object Name**

The **nm** command reports either the library or the object name associated with the file only if you specify the **-A** option.

- **Symbol Name**

- **Symbol Type**

The **nm** command represents the file's symbol type with one of the following characters (with weak symbols represented by the same characters as global symbols):

Item	Description
A	Global absolute symbol.
a	Local absolute symbol.
B	Global bss symbol.
b	Local bss symbol.
D	Global data symbol.
d	Local data symbol.
f	Source file name symbol.
L	Global thread-local symbol (TLS).
l	Static thread-local symbol (TLS).
T	Global text symbol.
t	Local text symbol.
U	Undefined symbol.

- **Value**

- **Size**

The **nm** command reports the size associated with the symbol, if applicable.

Flags

Item	Description
-A	Displays either the full path name or library name of an object on each line.
-B	Displays output in the Berkeley Software Distribution (BSD) format: <pre>value type name</pre>
-C	Suppresses the demangling of C++ names. The default is to demangle all C++ symbol names. Note: Symbols from C++ object files have their names demangled before they are used.
-d	Displays a symbol's value and size as a decimal. This is the default.
-e	Displays only static and external (global) symbols.
-f	Displays full output, including redundant .text, .data, and .bss symbols, which are normally suppressed.
-g	Displays only external (global) symbols.
-h	Suppresses the display of output header data.

Item	Description
-l	<p>Distinguishes between WEAK and GLOBAL symbols by appending a * to the key letter for WEAK symbols. If used with the -P option, the symbol type for weak symbols is represented as follows:</p> <p>V Weak Data Symbol</p> <p>W Weak Text Symbol</p> <p>w Weak Undefined Symbol</p> <p>Z Weak bss Symbol</p>
-o	Displays a symbol's value and size as an octal rather than a decimal number.
-P	<p>Displays information in a standard portable output format:</p> <pre style="background-color: #f0f0f0; padding: 5px;">library/object name name type value size</pre> <p>This format displays numerical values in hexadecimal notation, unless you specify a different format with the -t, -d, or -o flags.</p> <p>The -P flag displays the library/object name field only if you specify the -A flag. Also, the -P flag displays the size field only for symbols for which size is applicable.</p>
-p	Does not sort. The output is printed in symbol-table order.
-r	Sorts in reverse order.
-t <i>Format</i>	<p>Displays numerical values in the specified format, where the <i>Format</i> parameter is one of the following notations:</p> <p>d Decimal notation. This is the default format for the nm command.</p> <p>o Octal notation.</p> <p>x Hexadecimal notation.</p>
-T	Truncates every name that would otherwise overflow its column, making the last character displayed in the name an asterisk. By default, nm displays the entire name of the symbols listed, and a name that is longer than the width of the column set aside for it causes every column after the name to be misaligned.
-u	Displays only undefined symbols.
-v	Sorts output by value instead of alphabetically.
-x	Displays a symbol's value and size as a hexadecimal rather than a decimal number.

Item	Description
-X mode	Specifies the type of object file nm should examine. The <i>mode</i> must be one of the following: <ul style="list-style-type: none"> 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. <p>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 nm to process any 64-bit objects and ignore 32-bit objects. The -X flag overrides the OBJECT_MODE variable.</p>

Note: The **nm** command supports the **—** (double hyphen) flag. This flag distinguishes a *File* operand if the file name can be misinterpreted as an option. For example, to specify a file name that begins with a hyphen, use the **—** flag.

Exit Status

This command returns the following exit values:

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

Examples

1. To list the static and external symbols of the object file `a.out`, enter:

```
nm -e a.out
```

2. To display symbol sizes and values as hexadecimal and sort the symbols by value, enter:

```
nm -xv a.out
```

3. To display symbol of all 64-bit objects in `libc.a`, ignoring all 32-bit objects:

```
nm -X64 /usr/lib/libc.a
```

Files

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

nmon Command

Purpose

Displays local system statistics in interactive mode and records system statistics in recording mode.

Syntax

Interactive mode

nmon [-h]

nmon [-s < seconds >] [-c < count >] [-b] [-B] [-g < filename >] [-k < disklist >] [-C < process1:process2:...:processN >] [-i]

Recording mode

nmon [-f | -F filename | -x | -X | -z] [-r < runname >] [-t | -T | -Y] [-s seconds] [-c number] [-w number] [-l dpl] [-d] [-g filename] [-k disklist] [-C < process1:process2:...:processN >] [-G] [-K] [-o outputpath] [-D] [-E] [-J] [-V] [-P] [-M] [-N] [-W] [-S] [-^] [-O] [-L] [-I percent] [-A] [-m < dir >] [-Z priority] [-i] [-y options]

Note: In recording mode, specify only one of the -f, -F, -z, -x, or -X flags as the first argument.

Description

The **nmon** command displays and records local system information. The command can run either in interactive or recording mode. If you specify any of the -F, -f, -X, -x, and -Z flags, the **nmon** command is in recording mode. Otherwise, the **nmon** command is in interactive mode.

The **nmon** command provides the following views in interactive mode:

- [Adapter I/O statistics](#) (pressing the **a** key)
- [AIO processes view](#) (pressing the **A** key)
- [Detailed Page Statistics](#) (pressing the **M** key)
- [Disk busy map](#) (pressing the **o** key)
- [Disk groups](#) (pressing the **g** key)
- [Disk statistics](#) (pressing the **D** key)
- [Disk statistics with graph](#) (pressing the **d** key)
- [ESS vpath statistics view](#) (pressing the **e** key)
- [Fibre Channel adapter statistics](#) (pressing the **^** key)
- [JFS view](#) (pressing the **j** key)
- [Kernel statistics](#) (pressing the **k** key)
- [Long term processor averages view](#) (pressing the **l** key)
- [Large page analysis](#) (pressing the **L** key)
- [Memory and paging statistics](#) (pressing the **m** key)
- [Network interface view](#) (pressing the **n** key)
- [NFS panel](#) (pressing the **N** key)
- [Paging space](#) (pressing the **P** key)
- [Process view](#) (pressing the **t** and **u** keys)
- [Processor usage small view](#) (pressing the **c** key)
- [Processor usage large view](#) (pressing the **C** key)
- [Shared Ethernet adapter statistics](#) (pressing the **O** key)

- Shared-processor logical partition view (pressing the **p** key)
- System resource view (pressing the **r** key)
- Thread level statistics (pressing the **i** key)
- Verbose checks OK/Warn/Danger view (pressing the **v** key)
- Volume group statistics (pressing the **V** key)
- WLM view (pressing the **W** key)

In the recording mode, the command generates the **nmon** files. You can view these files directly by opening them or with post-processing tools such as nmon analyzer. The **nmon** tool disconnects from the shell during the recording so that the command continues running even if you log out.

If you use the same set of keys every time you start the **nmon** command, you can place the keys in the NMON shell variable. For example, you can run the following command:

```
export NMON=mcd
```

Then run the **nmon** command.

To stop the **nmon** command from the command line, use the **kill -USR2** with the nmon process ID.

To print the background process IDs of the nmon recording, run the **nmon** command with the **-p** flag.

To limit the processes that the **nmon** command lists (online and to a file), you can utilize the following options:

- Set the program names in environment variables from **NMONCMD0** to **NMONCMD63**
- Use the **-C** flag with *cmd:cmd:cmd* parameter. For example, you can enter the following command:

```
nmon -C ksh:vi:syncd
```

To limit the disks that the **nmon** lists to a maximum of 64 (online only), use the **-k** flag with the *diskname* parameter. For example, you can enter the following command:

```
nmon -k hdisk2,hdisk0,hdisk3
```

The **nmon** tool disconnects from the shell during the recording, ensuring that the command continues running even if you log out. This function is not true in the case of recordings triggered using the on-demand recording facility.

Recording or monitoring journaled file system (JFS) statistics in nmon can prevent unloading a file system because the file system is in use while collecting statistics.

Inside workload partitions (WPAR), the **nmon** command shows global values for processors and memory statistics. The rest of the values are WPAR specific. The following statistics cannot be retrieved inside a WPAR, and the **nmon** screen does not support them inside a WPAR:

- Disks, disk I/O graphs, disk busy map, disk groups
- Disk adapters
- Paging space
- Volume group
- ESS/vpaths
- Fibre Channel adapters
- VIOS Shared Ethernet adapters

Note: The dynamic configuration changes applied to the system are not reflected in the current **nmon** recording. The nmon tool must be restarted for the new configuration changes to take effect.

Flags in Interactive Mode

You can use the following flags in the interactive mode.

Item	Description
-s < <i>seconds</i> >	Time interval between refreshing the screen. The default value is 2 seconds.
-c < <i>count</i> >	Number of times the screen must be refreshed.
-g < <i>filename</i> >	A file that contains user-defined disk groups that can be specified using the <i>filename</i> parameter. Each line in the file begins with a group name. The list of hard disks follows the group name and is separated by spaces. The file can contain a maximum of 64 disk groups. A hard disk can belong to various disk groups.
-b	Displays the view in black and white mode.
-B	Does not include boxes in the view. By default, the command displays boxes.
-h	Displays help information.
-k < <i>disklist</i> >	Reports only the disks in the disk list.
-i	Reports top thread level CPU utilization.

Flags in Recording Mode

Item	Description
-A	Includes the Asynchronous I/O section in the view.
-c	Specifies the number snapshots that must be taken by the command. The default value is 10000000.
-d	Includes the Disk Service Time section in the view.
-D	Skips the Disk Configuration section.
-E	Skips the ESS Configuration section.
-f	Specifies that the output is in spreadsheet format. By default, the command takes 288 snapshots of system data with an interval of 300 seconds between each snapshot. The name of the output file is in the format of <i>hostname_YYMMDD_HHMM.nmon</i> .
-F	Specifies that the output is in spreadsheet format and the name of the output file is <i>filename</i> . The <i>filename</i> parameter specifies the name of the output file.
-g	Specifies the file that contains the user-defined disk groups, using the <i>filename</i> parameter. Each line in the file begins with a group name. The list of disks follows the group name and is separated with spaces. The file can contain a maximum of 64 disk groups. A disk can belong to various disk groups.
-G	Uses Greenwich mean time (GMT) instead of local time. This method is helpful when you compare nmon files from many LPARof 1 system for processor view but the LPARare in different time zones.
-i	Reports thread level statistics.
-I	Specifies the percentage of process threshold at which the command ignores the TOP processes statistics. The default percentage is 0.1. The command does not save the TOP processes statistics if the process is using less processor power than the specified percentage.
-J	Skips the JFS section.
-k	Specifies a list of disks to be recorded.

Item	Description
-K	Includes the RAW Kernel section and the LPAR section in the recording file. The -K flag dumps the raw numbers of the corresponding data structure. The memory dump is readable and can be used when the command is recording the data.
-l	Specifies the number of disks to be listed on each line. By default, 150 disks are listed per line. For EMC disks, specify a value of 64.
-L	Includes the large page analysis section.
-m	Changes the directory before the command saves the data to a file.
-M	Includes the MEMPAGES section in the recording file. The MEMPAGES section displays detailed memory statistics per page size.
-N	Includes the NFS section in the recording file. To collect the NFSv4 statistics, specify -NN .
-o	Specifies the file name or directory to which the recorded file is to be stored.
-O	Includes the Shared Ethernet adapter (SEA) VIOS sections in the recording file.
-P	Includes the Paging Space section in the recording file.
-r	Specifies the value for the <i>runname</i> field written to the spreadsheet file. By default, the value is the hostname.
-s	Specifies the interval in seconds between 2 consecutive recording snapshots.
-S	Includes WLM sections with subclasses in the recording file.
-t	Includes the top processes in the output. You cannot specify the -t , -T , or -Y flags with each other.
-T	Includes the top processes in the output and saves the command-line arguments into the UARG section. You cannot specify the -t , -T , or -Y flags with each other.
-V	Includes disk volume group section.
-w	Specifies the size of timestamp (Tnnnn) to be recorded. The timestamp is recorded in the .csv file. The value of the <i>number</i> parameter ranges from 4 through 16. For NMON analyzer, use the values 4 or 8.
-W	Includes the WLM sections into the recording file.
-x	Specifies the sensible spreadsheet recording for duration of 1 day for capacity planning. By default, the recording is done every 900 seconds for 96 times. This flag is equivalent to -ft -s 900 -c 96 .
-X	Specifies the sensible spreadsheet recording for duration of 1 hour for capacity planning. By default, the recording is done every 30 seconds for 120 times. This flag is equivalent to -ft -s 30 -c 120 .

Item	Description
-y options	<p>Controls the nmon recording sections. The values of the options parameter must be separated by commas.</p> <p>The following values are valid for the options parameter:</p> <p>sub=sea Records the SEA Bridged adapters statistics.</p> <p>sub=ssp Records the shared storage pool (SSP) statistics.</p> <p>PCPU=[on off] Enables or disables recording of Physical CPU (PCPU) sections, which are nothing but metrics that start with PCPU. These metrics are based on Processor Utilization of Resources Register (PURR). The default value is off.</p> <p>You can specify the following values for the PCPU section:</p> <p>on Enables recording of the PCPU section.</p> <p>off Disables recording of the PCPU section.</p> <p>SCPU=[on off] Enables or disables recording of Scaled CPU (SCPU) sections, which are metrics that start with SCPU. These metrics are based on Scaled Processor Utilization of Resources Register (SPURR). Default value is off.</p> <p>You can specify the following values for the SCPU section:</p> <p>on Enables recording of the SCPU section.</p> <p>off Disables recording of the SCPU section.</p> <p>dfreq=[on off] Enables or disables frequency sections, which are metrics that start with CPUMHZ. These metrics are based on the POWER options that are set. Default value is off.</p> <p>You can specify the following values for the dfreq section:</p> <p>on Enables recording of the CPUMHZ section.</p> <p>off Disables recording of the CPUMHZ section.</p> <p>Note:</p> <p>The value reported in the currentMHz column in CPUMHZ can vary slightly from the actual processor speed depending on conditions such as system load.</p> <p>The latest values of the options parameter override the previous values if the same value is used more than once in the command line.</p> <p>Example: If you run the command "nmon -f -y PCPU=on -y PCPU=off", the value <i>off</i> is used for the PCPU option.</p>
-Y	Includes the top process in the recording with all of the commands of the same name added and recorded. You cannot specify the -t , -T , or -Y flags together.
-z	Specifies the sensible spreadsheet recording for duration of 1day for capacity planning. By default, the recording is done every 900 seconds for 96 times. This flag is equivalent to -f -s 900 -c 96 .

Item	Description
-Z	Specifies the priority of the nmon command that is running. A value of -20 means important. A value of 20 means not important. Only root user can specify negative value.
-^	Includes the Fibre Channel (FC) sections.

Parameters

Item	Description
<i>disklist</i>	Specifies a list of disks.
<i>dir</i>	Specifies a directory.
<i>dpl</i>	Specifies the number of disks to list on each line.
<i>filename</i>	Specifies a file that contains the disk group you select.
<i>number</i>	Specifies the number of refreshes.
<i>count</i>	Specified the number of times to record.
<i>percent</i>	Specifies the percentage of processor usage.
<i>priority</i>	Specifies the priority of processes to be run.
<i>runname</i>	Specifies the value for the <i>runname</i> field in the spreadsheet file to be run.
<i>seconds</i>	Specifies the interval, in seconds, of refreshing the snapshot.
<i>outputpath</i>	Specifies the path for the output file.

Subcommands

Item	Description
space	Refreshes the screen immediately.
.	Displays only busy disks and processes.
~	Switches to the topas screen.
^	Displays the Fibre Channel adapter statistics
+	Doubles the screen refresh time.
-	Decreases the screen refresh time by half.
0	Resets the peak values of statistics (displayed on the screen) to zero. Applicable only for panels that display peak values.
a	Displays the I/O statistics of the adapters.
A	Summarizes the Async I/O (AIO server) processes.
b	Displays the view in black and white mode.
c	Displays processor statistics with bar graphs.
C	Displays processor statistics. It is useful for comparison when the number of processors ranges from 15 to 128.
d	Displays the I/O information of disks. To display specific disks only, specify the -k flag.
D	Displays the I/O statistics of disks. To get additional statistics of the disks, press the D key more than once.

Item	Description
e	Displays the I/O statistics of the ESS virtual path logical disks.
g	Displays the I/O statistics of the Disk Group. You must specify the -g flag with this key.
h	Displays the online help information.
j	Displays the JFS statistics.
k	Displays the internal statistics of the kernel.
l	Displays the processor statistics in long format. More than 75 snapshots are displayed with bar graphs.
m	Displays the memory and paging statistics.
M	Displays multiple page size statistics in pages. If you press the M key twice, the statistics are displayed in megabytes.
n	Displays the network statistics.
N	Displays the statistics of the NFS Network file system. If you press the N key twice, you see the NFSv4 statistics.
o	Displays the map of Disk I/O.
O	Displays only the Shared Ethernet adapter VIOS.
p	Displays the statistics of the partitions.
P	Displays the statistics of the paging space.
q	Quits. You can also use the x , or Ctrl+C key sequence.
r	Displays the resource type, system name, cache details, AIX version, and the LPAR information.
S	Displays the WLM with subclasses.
t	Displays the statistics of top processes. You can press the following keys with this subcommand: <ul style="list-style-type: none"> • 1: Displays basic details. • 2: Displays accumulated process information. • 3: Sorts the view by processor. • 4: Sorts the view by size. • 5: Sorts the view by I/O information.
u	Displays the top processes with the command arguments. To refresh the arguments for new processes, press the u key twice.
U	Displays the top processes with the command arguments, and the workload class or workload partitionworkload partition information.
v	Highlights status of pre-defined system resources and categorizes them as either danger, warnings, or normal.
V	Displays the statistics of the Disk Volume Group.
w	Displays the wait processes when used with the top processes.
W	Displays the statistics of the Workload Manager (WLM).
[Triggers a custom on-demand recording. The recording initiated exits along with the interactive nmon if not stopped earlier.
]	Stops a custom recording triggered by [.

Output Details

This section provides explanations to the metrics that are displayed on nmon screen.

System resources view

This view provides general information about the system resources. To display this view, press the **r** key. It contains information about the following resources:

- The number of processors in the system.
- The number of online processors that are active in the system.
- The frequency of the processors.
- The version of AIX and its technical level.
- The type of the running kernel.
- The logical partition.
- The power savings mode of the logical partition.
- The model of the hardware.
- The processor architecture of the system.
- The type of the platform bus.
- The cache information of processors.
- The number of active events.
- The old serial number. This number is the system ID of the partition before the dynamic configuration event.
- The current serial number. This number is the current system ID or the system ID of the partition after the dynamic configuration event.
- The local time of the last dynamic reconfiguration event. This information is labeled with the "When" keyword.
- The sub processor mode of the logical partition.

AIO Processes View

The AIO processes view provides information about the asynchronous I/O (AIO) processes. To display this view, press the **A** key. The following columns are displayed on the screen:

Item	Description
Total AIO Processes	The total number of AIO processes.
Actually in use	The number of AIO processes that uses more than 0.1% of the processor.
CPU Used	The percentage of the processor that is used by all of the kernel processes.
All time peak	The maximum number of kernel processes that are running since the system starts.
Recent peak	The recent maximum number of kernel processes that use more than 0.1% of the processor.
Peak	The maximum percentage of the processor that is used by all of the kernel processes.

Process View

The **Process View** provides details of the processes in the system. To display this view, press the **t** key or the **v** key. It contains the following columns are displayed on the screen:

Item	Description
pid	The ID of the process.

Item	Description
ppid	The ID of the parent process.
User	The user ID of the process.
Proc Group	The ID of the process group.
Nice	The initial priority of a process. This value is set by the nice command.
Priority	The base schedule priority of a process.
Status	The status of a program.
Proc_Flag	The flag of a process.
Thrds	The number of threads.
Files	The maximum file index that is in use.
Foreground	Foreground process or background process.
Command	The name of the command.
Time Start	The time when the command started.
CPU-Total	The total time that the process takes since it starts.
Child Total	The total time that the child process takes since it starts.
Delta-Total	The total time taken by the process in the interval.
%CPU Used	The percentage of the processor that is used in the last interval.
Size KB	The size of the pages in kilobytes.
Res Size	The sum of real-memory data (resident set) and real-memory (resident set) text size of the process.
Res Set	The sum of real-memory data (resident set) and real-memory (resident set) text size of the process.
Res Text	The real-memory text size of the process.
Res Data	The real-memory data size of the process.
Char I/O	The number of I/O characters per second from the last interval.
RAM Use	The percentage of the RAM that is used.
Paging I/O	The I/O page faults per second in the last interval.
Paging Other	The non-I/O page faults per second in the last interval.
Paging Repages	The number of repage faults per second in the last interval.
Class	The Workload Manager class name of the process.

Processor Usage Small View

The Processor Usage Small View provides a brief summary of the user, system, idle, and wait time of logical processors, the corresponding entitlement, and the virtual processor used. You can generate the Processor Usage Small View by pressing the **c** key.

Processor Usage Large View

The Processor Usage Large View displays the use of logical processor in a graph. To display this view, you can press the **C** key.

The following labels are used to identify time that is spent in different modes:

- **s**: Labels the percentage of time that is spent in system mode
- **u**: Labels the percentage of time that is spent in user mode

Shared-Processor Logical Partition View

The Shared-Processor Logical Partition View includes flags that indicate the following information of a partition:

- Whether the partition is an LPAR or not
- Whether the partition can be an LPAR or not
- Whether the partition is shared or dedicated
- Whether migration is enabled on this server. Refer the Hardware Management Console (HMC) to determine whether migration has been disabled for the logical partition.
- Whether the SMT is turned on or off
- Whether the shared-partition is capped or uncapped
- Whether LPAR is SMT bound or enabled
- Whether the LPAR flags are set, and whether they are set to display a value greater than `AVG=1p`

If the flags are set, the **nmon+C** graph contains information about the `Cpu_user` and the `Avg_user`, respectively. You can view the graph in the right column.

To display this view, you can press the **p** key.

Processors:

The following metrics of the processor status are displayed in this view:

Item	Description
Max Phys in Sys	Maximum number of physical processors in the system
Phys CPU in system	Number of physical processors in the system
Virtual Online	Number of online virtual processors
Logical online	Number of online logical processors
Physical pool	Number of shared physical processors in the shared pool ID that this partition is assigned to
SMT threads/CPU	Number of SMT threads per processor

Capacity:

The following information displays the processor capacity:

Item	Description
Cap. Processor Min	Minimum number of processing units that are defined for this LPAR
Cap. Processor Max	Maximum number of processing units that are defined for this LPAR
Cap. Increment	Granularity at which changes to the entitled capacity can be made
Cap. Unallocated	Sum of the number of processor units that are unallocated from shared LPAR in an LPAR group
Cap. Entitled	Entitled capacity
MinReqVirtualCPU	Minimum required virtual processors for the LPAR

ID Memory:

The following metrics of the ID memory are displayed:

Item	Description
LPAR ID Group:Pool	ID of an LPAR group and its pool ID

Item	Description
Memory (MB/GB) Min:Max	Minimum and maximum memory that is defined for this LPAR in megabytes or gigabytes
Memory(MB/GB) Online	Online real memory in megabytes or gigabytes
Memory Region LMB	Size in bytes of one logical memory block (LMB)

Time (in seconds):

Item	Description
Time Dispatch Wheel	Interval during which each virtual processor receives its entitlement
MaxDispatch Latency	Maximum latency in seconds between the dispatch of the LPAR on the physical processors
Time Pool Idle	Time in seconds that the shared processor pool is idle
Time Total Dispatch	Total time in seconds that the LPAR dispatches

Minimum and Maximum Values of Processors

The following minimum and maximum values of processors are displayed:

Item	Description
Virtual CPU (Min - Max)	Minimum number and maximum number of virtual processors in the LPAR definition
Logical CPU (Min - Max)	Minimum number and maximum number of logical processors

Weight

The following information about the weight of the processor is displayed:

Item	Description
Weight Variable	Variable weight of the processor capacity
Weight Unallocated	Unallocated variable weight available for this partition

NFS Panel

The NFS Panel provides information about the Network File System (NFS). To display this view, press the **N** key. The following metrics are included in the view:

Item	Description
Root	NFS V2 server and client root requests
Wrcache	NFS server and client write cache requests
Null	NFS server and client write cache requests
Getattr	NFS server and client get attributes requests
Setattr	NFS server and client set attributes requests
Lookup	NFS server and client filename lookup requests
Readlink	NFS server and client read link requests
Read	NFS server and client read requests
Write	NFS server and client write requests
Create	NFS server and client file creation requests
Mkdir	NFS server and client directory creation requests

Item	Description
Symlink	NFS server and client symbolic link creation requests
Remove	NFS server and client file removal requests
Rmdir	NFS server and client directory removal requests
Rename	NFS server and client file renaming requests
Link	NFS server and client link creation requests
Readdir	NFS server and client read-directory requests
Fsstat	NFS server and client file-status requests
Access	NFS V3 server and client access requests
Mknod	NFS V3 server and client mknod creation requests
readdir+	NFS V3 server and client read-directory plus requests
Fsinfo	NFS V3 server and client file information requests
Pathconf	NFS V3 server and client path configuration requests
Commit	NFS server and client commit requests
Bad calls	NFS server and client failed calls
Calls	NFS server and client requests

The following NFS V4 client/server statistics are printed when you press the **N** key twice.

Item	Description
Access	NFS V4 server and client access requests
acl_read	NFS V4 client reading access control list (ACL)
acl_stat_l	NFS V4 client that is retrieving long ACL information
acl_write	NFS V4 client write access control list (ACL)
Clntconfirm	NFS V4 client confirm operations
Close	NFS V4 client closing files
Commit	NFS V4 server and client committed
Compound	NFS V4 server compound calls
Create	NFS V4 server and client that is creating a non-regular object
Deletpurge	NFS V4 server purge delegations that is awaiting recovery
Delegreturn	NFS V4 server and client that is returning delegation
Finfo	NFS V4 client that is obtaining file information
getattr	NFS V4 server and client retrieving attributes
getfh	NFS V4 server retrieving file handles
Link	NFS V4 server and client that is linking operations
Lock	NFS V4 server and client that is locking operations
lockt/test	NFS V4 server that is testing the specified lock or NFS V4 client lock test
locku/unlock	NFS V4 server or NFS V4 client unlock operations
lookup	NFS V4 server and client that is looking up filenames
lookupp	NFS V4 server that is looking up parent directories

Item	Description
mkdir	NFS V4 client that is creating directories
mknod	NFS V4 client that is creating special files
Null	NFS V4 server null calls or NFS V4 client null calls
nverify	NFS V4 server verifying difference in attributes
openattr	NFS V4 server opening named attribute directories
openconfirm	NFS V4 server and client that is confirming the open for usage
opendowngrade	NFS V4 server and client that is downgrading the access for a specified file
Open	NFS V4 server and client open operations
operations	NFS V4 server and client operations
pcl_read	NFS V4 client extracting numeric data from printer control language (PCL) files
pcl_readstat_l	NFS V4 client pcl_stat long operations
pcl_stat	NFS V4 client pcl_stat operations
pcl_write	NFS V4 client pcl_write operations
putfh	NFS V4 server setting current file handles
putpubfh	NFS V4 server setting public file handles
putrootfh	NFS V4 server setting root file handles
readdir	NFS V4 server and client reading directories
readlink	NFS V4 server and client reading symbolic links
Read	NFS V4 server and client reading data from files
release	NFS V4 server and client release_lock operations
remove	NFS V4 server and client removing file system object
rename	NFS V4 server and client renaming object names
renew	NFS V4 server and client renewing leases
replicate	NFS V4 client replicate operations
restorefh	NFS V4 server restoring file handles
rmdir	NFS V4 client removing directories
savefh	NFS V4 server saving file handles
secinfo	NFS V4 server and client obtaining security information
setattr	NFS V4 server and client setting object attributes
setclient	NFS V4 server and client setclient operations
stats	NFS V4 client file statistics requests
symlink	NFS V4 client symbolic link operations
verify	NFS V4 client verifying same attributes
write	NFS V4 server and client writing to files

Network Interface View

The Network Interface View shows the statistics errors for the network. You can view this information by pressing the **n** key.

If the screen is updated three times with no network errors, the Network Interface View does not contain the network error statistics.

The following metrics are displayed in this view:

Item	Description
I/F Name	Interface name
Recv-KB/s	Data that are received in kilobytes per second in the interval
Trans-KB/s	Data that are transmitted in kilobytes per second in the interval
Packin	Number of packets that are received in the interval
Packout	Number of packets that are sent in the interval
Insize	Average size of packet that is received in the interval
Outsize	Average size of packet that is sent in last interval
Peak->Recv	Peak value of received data in kilobytes per second
Peak->Trans	Peak value of sent data in kilobytes per second
Total Recv	Total received data in megabytes per second
Total Sent	Total sent data in megabytes per second
MTU	Maximum size of transport unit in bytes
Ierror	Number of input errors
Oerror	Number of output errors
Collision	Number of collisions
Mbits/s	Adapter bit rate in megabits (Mbits) per second. If the network adapter is larger than 10Gb, the adapter bit rate is shown as 10240 Mbits per second.
Description	Description of the interface

WLM View

The WLM View shows the information about workload management. You can open this view by pressing the **W** key. To turn on the subclasses section, press the **S** key from WLM View. To turn off the subclasses section, press the **S** key again.

The following metrics are displayed in this view:

Item	Description
CPU	Percentage of processor use of the class.
MEM	Percentage of physical memory use of the class.
BIO	Percentage of disk I/O bandwidth use for the class.
Process (Procs)	Number of processes in the class.
Tier (T)	Tier number. The value ranges from zero through nine.
Inheritance (I)	Values of the inheritance attribute. A value of zero means no. A value of one means yes.
Location	Values of location. A value of one means avoiding transfer of segments to shared classes. Otherwise, a value of zero is displayed.

Disk Busy Map

The Disk Busy Map shows the use statistics of disks. To display this map, press the **o** key. A maximum of 100 disks is shown per screen. Only the disks with the names that range from hdisk0 through hdisk100 are displayed. The following table shows the symbols for the ranges of names.

Symbols	Names
-	Less than 5
.	Less than 10
-	Less than 20
+	Less than 30
o	Less than 40
0	Less than 50
O	Less than 60
8	Less than 70
X	Less than 80
#	Less than 90
@	Less than 100 and equal to 100

Disk Groups

Multiple disks can be monitored by placing them in groups. To display this view, press the **g** key.

You must create a group configuration file that contains the lines as shown in the following example:

```
<Group_name1> <disk_name1> <disk_name2> ....
<Group_name2> <disk_nameA> <disk_nameB> ...
```

In the example, <Group_name1> is the name of the first disk in the group; <disk_name1> and <disk_name2> are the first and second disks in the group.

To see the Disk Group I/O, run the **nmon** command with the **-g** flag and a group file, and then press the **g** key. The following metrics are shown in this view:

Item	Description
Name	Disk Group name. You can specify a maximum of 64 groups. A disk can be in multiple groups.
Disks	Number of disks in the group.
Read/Write-KB/s	Data transfer rate of read and written data in kilobytes per second in the interval.
TotalMB/s	Sum of read and written data in megabytes per second in the interval.
Xfers/s	Number of read and written data transfers per second in the interval.
BlockSizeKB	Block size in kilobytes read or written per transfer operation.

ESS Vpath Statistics View

This view provides the ESS Vpath Statistics. To display this view, press the **e** key. The following metrics are included in this view:

Item	Description
Name	Name of the virtual path.
Size	Size of the ESS path.
AvgBusy	Average busy use of the disk.

Item	Description
Write-KB/s	Transfer rate of written data in kilobytes per second in the interval.
Read-KB/s	Transfer rate of read data in kilobytes per second in the interval.
Xfers/s	Number of read and write transfers per second.
Total vpaths	Number of virtual paths.

JFS View

This view provides the Journaled File System (JFS) statistics. To display this view, press the **j** key. The following statistics are recorded in this view:

Item	Description
FileSystem	Name of the file system.
Size (MB)	Size in megabytes for the file system.
Free (MB)	Available free space in megabytes in the file system.
%Used	Percent of file system used.
%Inodes	Percent of file system that is used by i-nodes.
Mount point	Local mount point.

Kernel Statistics

This view contains the statistics of the kernel. To display this view, press the **k** key. The following statistics are displayed in this view:

Item	Description
runqueue	Average number of threads that are ready to run but are waiting for an available processor.
pswitch	Number of processor switches per second in the interval.
fork	Number of forks per second in the interval.
exec	Number of execs per second in the interval.
msg	Number of interprocess communication (IPC) messages that are sent and received per second in the interval.
sem	Number of semaphore operation system calls per second in the interval.
hw intrp	Number of device interrupts per second in the interval.
sw intrp	Number of off-level handlers that are called per second in the interval.
Swapin	Number of processes in swap queue per second in the interval.
Syscall	Number of system calls per second in the interval.
read	Number of read calls per second in the interval.
write	Number of write calls per second in the interval.
readch	Number of characters that are transferred through read system call per second in the interval.
Writech	Number of characters that are transferred through write system call per second in the interval.
R + W (MB/s)	Number of read and write characters in megabytes per second in the interval.
Uptime	Time duration for which the system is up.

Item	Description
iget	Number of inode lookups per second in the interval.
dirblk	Number of 512-byte block reads by the directory search routine to locate an entry for a file per second in the interval.
namei	Number of vnode lookup from a path name per second in the interval.
ksched	Number of kernel processes that are created per second in the interval.
koverf	Number of kernel process creation attempts where the user forked to the maximum limit or the configuration limit of processes that are reached per second in the interval.
kexit	Number of kernel processes that become zombies per second in the interval.

Long Term Processor Averages View

This view provides information about the instantaneous system. To display this view, press the **l** key. You can use the following labels to identify the time that is spent in different modes:

- **s**: Labels the percentage of the time that is spent in system mode.
- **u**: Labels the percentage of the time that is spent in user mode.
- **w**: Labels the percentage of the time that is spent in wait mode.

The following metrics are displayed on this view:

Item	Description
EntitledCPU	Entitled capacity of the partition.
UsedCPU	Number of physical processors that are used by the partition.

Large Page Analysis

This view provides analysis of the large page. To display this view, press the **L** key. The following information is displayed:

Item	Description
Count	Number of large pages and their total size.
Free	Percentage of free large pages and their size.
In Use	Percentage of large pages in use and their size.
Size	Size of a large page.
High water mark	Large page high watermark.

Paging Space

This view prints the paging-space statistics. To display this view, press the **p** key. The following metrics are displayed in the view:

Item	Description
PagingSpace	Number of paging spaces.
Volume-Group	Number of volume groups.
Type	Type of logical volumes. The types can be NFS or LV.
LPs	Size of logical partitions.
MB	Size in megabytes.
Used	Percentage of use for volume groups.

Item	Description
IOpending	Number of pending I/O in the paging space.
Active/Inactive	Active or inactive paging space.
Auto/NotAuto	Indicates whether the paging space is auto that is loaded or not.

Volume Group Statistics

This view provides statistics for the volume group. To display this view, press the **V** key. The following information is displayed in the view:

Item	Description
Name	Volume group name.
Disks	Number of disks in the group.
AvgBusy	Average busy of the disks in the volume group.
Read/Write-KB/s	Data transfer rate of read and written data in kilobytes per second in the interval.
TotalMB/s	Sum of read and written data in megabytes per second in the interval.
Xfers/s	Number of read and written transfers per second in the interval.
BlockSizeKB	Block size that is read or written per transfer in kilobytes per second in the interval.

Disk Statistics

This view provides statistics for disks. To display this view, press the **D** key. You can press the **D** key for the following times to view various metrics:

- Once: Shows disk numbers
- Twice: Shows disk descriptions
- Three times: Shows service times
- Four times: Shows disk statistics with graphs similar to the graph shown on pressing the **d** key

Disk Numbers (pressing the **D** key once)

The following metrics are shown in this view:

Item	Description
Name	Name of the disks.
Busy	Average busy of the disks.
Read-KB/s	Data transfer rate of read data in kilobytes per second in the interval.
Write-KB/s	Data transfer rate of written data in kilobytes per second in the interval.
Transfers/sec	Number of read and written transfer per second in the interval.
SizeKB	Block size that is read or written per transfer in kilobytes per second in the interval.
Peak	Peak percentage of average busy.
Peak KB/s	Peak that is read and written data in kilobytes per second.
qDepth	Number of requests that are sent to disk and are not completed.
Totals Size (GB)	Total size of disks in gigabytes.
Totals Free (GB)	Total free space that is left in disks in gigabytes.

Item	Description
Totals Read (MB/s)	Total data transfer rate of read data from all disks in megabytes per second.
Totals Write (MB/s)	Total data transfer rate of written data to all disks in megabytes per second.

Disk Descriptions (Pressing the **D** key twice)

The following metrics are shown in this view:

Item	Description
Name	Disk names.
Size (GB)	Size of disks in gigabytes.
Free (GB)	Free space that is left in disk in gigabytes.
Disk Paths	Number of paths that are defined to the disk.
Disk Adapter	Name of disk adapters.
Volume Group	Volume group that the disk belongs to.
Disk Description	Description of the disk.
Totals Size (GB)	Total size of disks in gigabytes.
Totals Free (GB)	Total free space that is left in disks in gigabytes.
Totals Read (MB/s)	Total data transfer rate of read data from all disks in megabytes per second.
Totals Write (MB/s)	Total data transfer rate of written data to all disks in megabytes per second.

Service Times (Pressing the **D** key three times)

The following metrics are displayed in the view:

Item	Description
Disk	Name of the disk.
Service (in msec)	Average service time per request in milliseconds.
Wait (in msec)	Average waiting time per request in milliseconds.
ServQ size	Average number of requests in service queue.
WaitQ size	Average number of requests that is waiting to be accomplished.
ServQ Full	Number of times the disk is not accepting any coming requests.
Totals Size (GB)	Total size of disks in gigabytes.
Totals Free (GB)	Total free space that is left in disks in gigabytes.
Totals Read (MB/s)	Total data transfer rate of read data from all disks in megabytes per second.
Totals Write (MB/s)	Total data transfer rate of written data to all disks in megabytes per second.

Disk Statistics With Graphs (Pressing the **D** key four times)

This view displays disk statistics with graphs. To display this view, press the **d** key. The following metrics are displayed in this view:

Item	Description
Name	Name of the disk.
Busy	Average percentage of busy for the disk.
Read-KB/s	Data transfer rate of read data in kilobytes per second.

Item	Description
Write-KB/s	Data transfer rate of written data in kilobytes per second.

Memory and Paging Statistics

The view provides information about the memory and paging statistics. To display this view, press the **m** key. The following metrics are included in this view:

Item	Description
%Used	Percentage of used space in physical memory and paging space.
%Free	Percentage of free space in physical memory and paging space.
MB Used	Physical memory and paging space that are used in megabytes.
MB Free	Physical memory and paging space that are free in megabytes.
Pages/sec to Paging Space	Number of I/O pages that are transferred to or from the paging space per second.
Pages/sec to file system	Number of I/O pages that are transferred to or from the file system per second.
Page Scans	Number of page scans by clock.
Page Faults	Number of page faults.
Page Cycles	Number of page replacement cycles.
Page Steals	Number of pages steals.
Numperm	Number of frames that are used for files (in 4-KB pages).
Process	Percentage of real memory that is used by process segments.
System	Percentage of real memory that is used by system segments.
Free	Percentage of real memory that is free.
Total	Percentage of total real memory used.
Min/Maxperm	The minperm and maxperm values for page steals.
Min/Maxfree	The minfree and maxfree pages free list.
Min/Maxpgahead	Minimum and maximum number of page ahead pages.
Total Virtual	Total virtual memory.
Accessed Virtual	Active virtual memory.
Numclient	Number of client frames.
Maxclient	Maximum number of client frames.
User	Real memory that is used by non-system segments.
Pinned	Real memory that is pinned.

The AMS statistics are displayed in the **topas_nmon** memory panel. To display this view, press the **m** key. The following metrics are included in this view:

Item	Description
Pool	AMS pool ID of the pool that the logical partition (LPAR) belongs to.
Weight	Weight of the variable memory.
pMem	Physical memory currently backing up the logical memory partition (in MB).
hpi	Number of hypervisor page-ins.

Item	Description
hpit	Time that is spent in hypervisor page-ins (in seconds).

Logical unit information:

Item	Description
Size (MB)	Total size that is allocated for the logical unit.
Lu Udid	Logical unit identifier.

Adapter I/O Statistics View

This view provides the adapter I/O statistics. To display this view, press the **a** key. The following metrics are displayed in this view:

Item	Description
Adapter	Name of the adapter.
Busy%	Bandwidth use of the adapter. This is the aggregate Busy% of the disks connected to this adapter. The value might exceed 100% if more than one disk is connected to the adapter.
Read-KB/s	Data transfer rate of read data in kilobytes per second.
Write-KB/s	Data transfer rate of written data in kilobytes per second.
Transfers	Number of read and write transfers.
Disks	Number of disks.
Adapter-Type	Type of the adapter.

Shared Ethernet adapter

This view provides shared Ethernet adapter statistics in a Virtual I/O Server (VIOS). To display this view, press the **O** key. The following metrics are displayed in this view:

Item	Description
Number	Serial number.
Name	Name of the shared Ethernet adapter.
Recv-KB/s	Data transfer rate of received data in kilobytes per second.
Trans-KB/s	Data transfer rate of sent data in kilobytes per second.
Packin	Number of packets that are received per second in the interval.
Packout	Number of packets that are sent per second in the interval.
Insize	Average size per second for received packet in the interval.
Outsize	Average size per second for outgoing packet in the interval.

Verbose Checks OK/Warn/Danger

This view prints the statistics for processor, memory, and disks. It also prints the status message, such as OK, Warn, or Danger, which is based on the system metrics that exceed the pre-defined threshold values. To display this view, press the **v** key.

Detailed Page Statistics

This view provides page statistics. To display this view, press the **M** key.

If you press the **M** key once, the view contains the statistics in pages. If you press the **M** key twice, the page statistics are shown in megabytes.

The following metrics are shown in this view:

Item	Description
Numframes	Number of real memory frames of the page size.
Numfrb	Number of pages on free list.
Numclient	Number of client frames.
Numcompress	Number of frames in compressed segments.
Numperm	Number of frames in non-working segments.
Numvpages	Number of accessed virtual pages.
Minfree	Minimum free list.
Maxfree	Maximum free list.
Numpout	Number of page-outs.
Numremote	Number of remote page-outs.
Numwseguse	Number of pages in use for working segments.
Numpseguse	Number of pages in use for persistent segments.
Numclseguse	Number of pages in use for client segments.
Numwsegpin	Number of pages that are pinned for working segments.
Numpsegpin	Number of pages that are pinned for persistent segments.
Numclsegpin	Number of pages that are pinned for client segments.
numpgsp_pgs	Number of allocated page spaces.
numralloc	Number of remote allocations.
pfrsvdblks	Number of system reserved blocks.
Pfavail	Number of pages available for pinning.
Pfpinavail	Application level number pages available for pinning.
system_pgs	Number of pages on segment control blocks (SCB) that are marked with V_SYSTEM .
nonsys_pgs	Number of pages on SCBs not marked with V_SYSTEM .
Numpermio	Number of pageouts in non-working storage.
Pgext	Number of page faults.
Pgrclm	Number of pages reclaims.
Pageins	Number of paged-in pages.
Pageouts	Number of paged-out pages.
Pgspgins	Number of paged-in pages from page space.
Pgspgouts	Number of paged-out pages from page space.
Numsios	Number of I/O started.
Numiodone	Number of I/O completed.
Zerofills	Number of zero-filled pages.
Exfills	Number of exec-filled pages.
Scans	Number of page scans by clock.
Cycles	Number of clock hand cycles.

Item	Description
pgsteals	Number of pages steals.

Fibre Channel Adapter Statistics

This view contains information about the Fibre Channel adapter. You can see this view by pressing the caret (-^) key. The following metrics are included in this view:

Note: If the N_Port Virtualization (NPIV) is configured on the VIOS, use the -^ option in the **nmon** command to record the NPIV related statistics.

Item	Description
Number	Serial number.
Name	Name of the Fibre Channel adapter.
Receive-KB/s	Data transfer rate of received data in kilobytes per second.
Transmit-KB/s	Data transfer rate of sent data in kilobytes per second.
Requests In	Number of requests that are received per second in the interval.
Requests Out	Number of requests that are sent per second in the interval.
Outsize	Average outgoing packet size per second in the interval.

Thread level statistics

This view contains information about thread level statistics. To display this view, press the **-i** key. The following metrics are included in this view:

Item	Description
PID	Process ID to which the thread belongs.
TID	Top thread ID that uses higher CPU. Sorting is based on CPU utilization in descending order.
%CPU	Percentage of CPU used by the specific thread.
BOUND CPU ID	Bounded CPU ID if the thread has been bound to any processor.

Environment Variables

Environment variables **NMON_START**, **NMON_END**, **NMON_SNAP**, and **NMON_ONE_IN** are used for collecting external data while recording in **nmon** format.

Item	Description
NMONCMD0, NMONCMD1, ..., NMONCMD63	You can monitor only the processes that are set in these variables when these environment variables are set. Alternatively, you can use the -C flag to restrict the commands in the process listing of the nmon command. For example, you can run the nmon -C db2:nmon:topas command.
NMON	Contains the set of key strokes corresponding to the initial set of panels to be displayed when the nmon command is started.

Item	Description
NMON_TIMESTAMP	<p>You can specify the NMON_TIMESTAMP variable to the following values:</p> <p>NMON_TIMESTAMP = 0 The recorded lines contain the <code>nmon Tnnnn</code> timestamps at the beginning of the line and work with the <code>nmon</code> data file.</p> <p>NMON_TIMESTAMP = 1 The lines contains timestamps that have the hours, minute, seconds, day, month, and year. This value can be used if you do not want to merge the data with the <code>nmon</code> file for analysis.</p>
NMON_START	External command to be started when the nmon recording begins.
NMON_END	External command to be started when the nmon recording ends.
NMON_SNAP	External command to be started periodically to record metrics.
NMON_ONE_IN	<p>You can specify the NMON_ONE_IN variable to the following values:</p> <p>NMON_ONE_IN=1 Runs the snap command every time the recording is done.</p> <p>NMON_ONE_IN=n Runs the snap command after the number of recordings that are specified by the <i>n</i> parameter is done.</p>

Examples

1. To generate the **nmon** recording in the current directory for two hours, capturing data every 30 seconds, enter the following command:

```
nmon -f -s 30 -c 240
```

2. To display the memory and processor statistics immediately after the **nmon** command is started, do the following steps:

- a. Enter the following command:

```
export NMON=mc
```

- b. Run the **nmon** command.

3. To run the **nmon** command for 20 seconds with the screen that is refreshed at 10 seconds, enter the following command:

```
nmon -c 10 -s 2
```

4. To run `nmon` in black and white mode, enter the following command:

```
nmon -b
```

5. To view the process information, do the following steps:

- a. Run the **nmon** command.

- b. Press the **t** key.
6. To view the list of views that **nmon** command provides, press the key **h**.
7. The following sample explains the steps to collect external data. In the sample, the `mystart` file, the `mysnap` file, and the `myend` file are executable and are in the path that the `$PATH` defines.
 - a. Set the environment variables as indicated in the following example:

```
$export NMON_TIMESTAMP=0
$export NMON_START="mystart"
$export NMON_SNAP="mysnap"
$export NMON_END="myend"
$export NMON_ONE_IN=1
```

In the previous example, the value of one is the default value for the `NMON_ONE_IN` environment variable. It generates one set of external recorded data for every snapshot of `nmon` recording.

- b. Modify the content of the `mystart` file as the following:

```
ps -ef >start_ps.txt
echo "PROCCOUNT,Process Count, Procs" >ps.csv
```

- c. Modify the content of the `mysnap` file as the following:

```
echo PROCCOUNT,$1,`ps -ef | wc -l` >>ps.csv
```

- d. Modify the content of the `myend` file as the following:

```
echo PROCCOUNT,$1,`ps -ef | wc -l` >>ps.csv
```

- e. Run the **nmon** command as follows:

```
nmon -f -s 2 -c 10
```

The recording finishes in 20 seconds.

The output of the `ps.csv` file is similar to the following sample:

```
PROCCOUNT,Process Count, Procs
PROCCOUNT,T0001, 43
PROCCOUNT,T0002, 43
PROCCOUNT,T0003, 43
PROCCOUNT,T0004, 43
PROCCOUNT,T0005, 43
PROCCOUNT,T0006, 43
PROCCOUNT,T0007, 43
PROCCOUNT,T0008, 43
PROCCOUNT,T0009, 44
PROCCOUNT,T0010, 44
PROCCOUNT,T0010, 44
```

To concatenate the generated `nmon` file with the `ps.csv` file that is generated by external recording, enter the following command:

```
cat filename.nmon ps.csv > c.csv
```

To get the graph, open the `c.csv` file in **nmon** analyzer.

8. To view the `hdisk` details, enter the **nmon** command with `-k` flag:

```
nmon -k hdisk1,hdisk2
```

The previous command shows the disk details for `hdisk1` and `hdisk2`. For `hdiskpower` devices, enter the following command:

```
nmon -k hdiskpower or
nmon -k power
```

Note: The `nmon -k hdisk` matches all the `hdisk` devices on the LPAR and does not match the `hdiskpower` devices.

All hdiskpower devices display as power in interactive and recording modes. For example, `nmon -k hdiskpower1` matches the device `hdiskpower1` and `nmon -k hdiskpower` matches all `hdiskpower` devices on the LPAR.

Note: The output of the `lsconf` and `lspv` commands in the `nmon` recording file is not affected by the changes to the `nmon-k` command.

Location

`/usr/bin/nmon`

`/usr/bin/topasrec`

no Command

Purpose

Manages the tuning parameters of the network.

Syntax

```
no [ -p | -x [-K] ] { -o Tunable[=NewValue] }
```

```
no [ -p | -x [-K] ] { -d Tunable }
```

```
no [ -p | -x [-K] ] { -D }
```

```
no [ -p | -x [-K] ] [-F] -a
```

```
no -h [Tunable]
```

```
no [-F] -L [Tunable]
```

```
no [-F] -x [Tunable]
```

Note: Multiple flags `-o`, `-d`, `-x`, and `-L` are allowed.

Description

Use the `no` command to configure parameters that used to tune the network. The `no` command sets or displays current or next system boot values for network tuning parameters. This command can also make permanent changes or defer changes until the next system reboot. Whether the command sets or displays a parameter, is determined by the accompanying flag. The `-o` flag does both these actions. It can either display the value of a parameter or set a new value for a parameter. When the `no` command is used to modify a network option, it logs a message to the syslog by using the LOG_KERN facility.

Note: Be careful when you use this command. If used incorrectly, the `no` command can cause your system to become inoperable.

Before you modify any tunable parameter, you must read about all its characteristics in the Tunable Parameters section, and follow the Refer To pointer instructions to understand the purpose. Ensure that the Diagnosis and Tuning sections for this parameter apply to the situation, and changing the value of this parameter helps to improve the performance of your system.

If the Diagnosis and Tuning sections both contain N/A, you must not change this parameter unless directed by AIX development.

Flags

Item	Description
-a	Displays current, reboot (when used with -r), Live Update value (when used with the -K option), or permanent (when used with -p) value for all tunable parameters, one per line in pairs <i>Tunable = Value</i> . For the permanent options, a value displays for a parameter if its reboot and current values are equal. Otherwise, NONE displayed as the value.
-d <i>Tunable</i>	Resets <i>Tunable</i> to its default value. The tunable parameter is not changed, but a warning message is displayed if -r flag is not used in combination and the tunable parameter that must be changed is set to one of the following values: <ul style="list-style-type: none">• The tunable is not set to its default value and it is of type Bosboot or Reboot.• The tunable is of type Incremental and must be changed from its default value.• The tunable parameter is of type Reboot and is supported throughout the Live Update operation.
-D	Resets all tunable parameters to their default value. The tunable parameter is not changed, but a warning message is displayed if either the -p or -r flags are used in combination and the tunable parameter that must be changed is of one of the following types: <ul style="list-style-type: none">• Bosboot or Reboot type.• Incremental type and is changed from its default value.• Reboot type and is supported throughout the Live Update operation.
-F	Forces restricted tunable parameters to be displayed when the options -a , -L , or -x are specified 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 <i>Tunable</i> parameter if one is specified. Otherwise, the -h flag displays the no command usage statement.
> > <-k	Sets the reboot value to its default 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 reboot value is set to its default value only in the <code>/etc/tunables/nextboot</code> file. The <code>/etc/tunables/nextliveupdate</code> file is updated with the default reboot value only if a nondefault value was set for the tunable parameter for the next Live Update operation.

|<<

Item**-L** [*Tunable*]**Description**

Lists the characteristics of one or all *Tunables*, one per line, by using the following format:

```

NAME          CUR   DEF   BOOT   LVUP
MIN   MAX   UNIT   TYPE
DEPENDENCIES
-----
General Network
Parameters
-----
sockthresh    85   85   85     0
100   %_of_thewall  D
-----
fasttimo      200  200  200    50
200   millisecond  D
-----
inet_stack_size 16   16   16
1     kbyte         R
-----

```

...
where:

```

CUR = current value
DEF = default value
BOOT = reboot value
LVUP = Live Update value
MIN = minimal value
MAX = maximum value
UNIT = tunable unit of measure
TYPE = parameter type the following command: 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

```

-o *Tunable* [=*NewValue*]

Displays the value or sets the *Tunable* to *NewValue*. The tunable parameter is not changed, but a warning message is displayed if the **-r** flag is not used with the **-o** flag and the tunable parameter that must be changed is of one of the following types:

- Bosboot or Reboot.
- Incremental and its current value is more than the specified value.
- Reboot and is supported throughout the Live Update operation.

When **-r** is used with this flag without a new value, the nextboot 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 **-p** is used in combination without a new value, a value displays only if the current and next boot values for tunable are the same. Otherwise, the value is displayed as NONE.

Item	Description
-p	Changes are applied to both current and reboot values when used in with -o , -d or -D that is turns on updating of the <code>/etc/tunables/nextboot</code> file in addition to updating of the current value. These combinations cannot be used on Reboot and Bosboot type parameters because their current value cannot be changed. When used with -a or -o without specifying a new value, the values are displayed when the current and next boot values for a parameter are the same. Otherwise, the value is displayed as NONE.
-r	Changes are applied to reboot values when used in with -o , -d , or -D flags. The -r flag turns on updating the <code>/etc/tunables/nextboot</code> file. If any parameter of type Bosboot is changed, the user is prompted to run bosboot. When used with -a or -o without specifying a new value, next boot values for tunables display instead of the 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 tunables, one per line, by using the following (spreadsheet) format:

```
tunable,current,default,reboot,liveupdate,min,max,unit,type,{dtunable }
```

where:

```
current = current value
default = default value
reboot = reboot value
liveupdate = Live Update value
min = minimal value
max = maximum value
unit = tunable unit of measure
TYPE = parameter type the following command: 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
```

If you change by using the **-o**, **-d**, or **-D** flag to a restricted tunable parameter, it results in a warning message that a tunable parameter of the restricted-use type is modified. If you also specify the **-r** or **-p** options on the command line, you are prompted for confirmation of the change. During system reboot, the presence of restricted tunables in the `/etc/tunables/nextboot` file that were modified to a value different from their default value by using a command line and by specifying the **-r** or **-p** options, results in an error log entry that identifies the list of these modified tunables.

If you change by using the **-o**, **-d**, or **-D** flag to a parameter of type Mount, it results in a warning message that the change is effective for future mountings.

If you change to a parameter of type Connect by using the **-o**, **-d**, or **-D** flag, it results in starting the **inetd** daemon and displays a warning message that the change is effective for future socket connections.

If you change to a parameter of type Bosboot or Reboot by using the **-o**, **-d**, or **-D** flag and without using the **-r** flag, it results in an error message.

If you change the current value of a parameter of type Incremental with a new value that is smaller than the current value by using the **-o**, **-d**, or **-D** flag and without using the **-r** flag, it results in an error message.

Tunable Parameters Type

All the tunable parameters that are manipulated by the tuning commands such as **no**, **nfso**, **vmo**, **ioo**, **schedo**, and **raso** commands are classified into the following categories:

Item	Description
Dynamic	Specifies whether the parameter can be changed at any time.
Static	Specifies whether the parameter can never be changed.
Reboot	Specifies whether the parameter can be changed during reboot.
Bosboot	Specifies whether the parameter can be changed by running bosboot command and rebooting the machine.
Mount	Specifies whether changes to the parameter are only effective for future file systems or directory mounts.
Incremental	Specifies whether the parameter can be incremented, except at boot time.
Connect	Specifies whether changes to the parameter are only effective for future socket connections.
Deprecated	Specifies whether this parameter cannot be changed and is no longer supported by the current release of AIX.

For parameters of type Bosboot, whenever there is a change, the tuning commands automatically prompt the user to ask if they want to run the **bosboot** command. For parameters of type Connect, the tuning commands automatically restart the **inetd** daemon if **pre520tune** is disabled.

Note: The current set of parameters that are managed by the **no** command includes Reboot, Static, Dynamic, Incremental, and Connect types.

Tunable Parameters

For default values and range of values for tunable parameters, refer the **no** command help (**-h <tunable_parameter_name>**).

Item	Description
arpqsize	<p>Purpose Specifies the maximum number of packets to queue while waiting for Address Resolution Protocol (ARP) responses.</p> <p>Tuning This attribute is supported by Ethernet, 802.3, Token Ring and FDDI interfaces.</p>
arpt_killc	<p>Purpose Specifies the time in minutes before a complete ARP entry is deleted.</p> <p>Tuning To reduce ARP activity in a stable network, you can increase arpt_killc.</p>

Item	Description
arptab_bsiz	<p>Purpose Specifies Address Resolution Protocol (ARP) table bucket size.</p> <p>Tuning The netstat -p arp command shows the number of ARP packets that are sent and the number of ARP entries that are purged from the ARP table. If large number of entries are purged, the ARP table size must be increased. Use arp -a to show the ARP table-hashing distribution.</p>
arptab_nb	<p>Purpose Specifies the number of ARP table buckets.</p> <p>Tuning The netstat -p arp command shows the number of ARP packets that are sent and the number of ARP entries that are purged from the ARP table. If large number of entries are purged, the ARP table size must be increased. Use arp -a to show the ARP table-hashing distribution. Increase this value for systems that have a large number of clients or servers. The default provides for $149 \times 7 = 1043$ ARP entries, but assumes an even hash distribution.</p>
bcastping	<p>Purpose Allows response to ICMP echo packets to the broadcast address.</p> <p>Tuning A value of 0 disables it and a value on 1 enables it. The default is to not respond to echo packets to a broadcast address. The default value prevents so called broadcast storms on the network that can result when multiple machines respond to a broadcast address.</p>
clean_partial_conns	<p>Purpose Specifies whether or not SYN attacks are avoided. If nonzero, clean_partial_conns specifies how many partial connections to be removed randomly to make room for new nonattack connections.</p> <p>Tuning A value of 0 disables this option. This option must be turned on for servers that need to protect against network attacks.</p>

Item	Description
delayack	<p>Purpose Delays ACKs for certain TCP packets and attempts to piggyback them with the next packet sent instead.</p> <p>Tuning This action is performed only for connections whose destination port is specified in the list of the delayackports attribute. This action can be used to increase the performance when communicating with an HTTP server by reducing the total number of packets sent. The parameter can have one of following four values:</p> <ul style="list-style-type: none"> 0 No delays, normal operation 1 Delays the ACK for the server's SYN 2 Delays the ACK for the server's FIN 3 Delay both the ACKs for the SYN and FIN.
delayackports	<p>Purpose Specifies the list of destination ports for which the operation that is defined by the <code>delayack</code> port option is performed.</p> <p>Tuning The attribute takes a maximum of 10 ports, which are separated by commas and enclosed in curly braces. For example:</p> <pre style="background-color: #f0f0f0; padding: 5px;">no -o delayackports={80,30080}.</pre> <p>To clear the list, set the option <code>delayackports={}</code>.</p>
dgd_flush_cached_routes	<p>Purpose Flushes the cached routes of sockets when Dead Gateway Detection detects a previous dead gateway back online. The connections are forced to reacquire the route before the data is sent.</p> <p>Tuning A value of 1 enables the DGD to flush the cached routes. A value of 0 disables it.</p>
dgd_packets_lost	<p>Purpose Specifies how many consecutive packets must be lost before Dead Gateway Detection decides that a gateway is down.</p>
dgd_ping_time	<p>Purpose Specifies the seconds that must pass between pings of a gateway by Active Dead Gateway Detection.</p>
dgd_retry_time	<p>Purpose Specifies the minutes a route's cost must remain raised when it is raised by Passive Dead Gateway Detection. After this many minutes pass, the route's cost is restored to its user-configured value. The unit that is specified is in numeric.</p>

Item	Description
directed_broadcast	<p>Purpose Specifies whether a directed broadcast to a gateway must be allowed or not.</p> <p>Tuning The value of 1 allows packets to be directed to a gateway that must be broadcast on a network on the other side of the gateway.</p>
fasttimo	<p>Purpose Allows you to set the millisecond delay for the TCP fast timeout timer. This timeout controls how often the system scans the TCP control blocks to send delayed acknowledgments.</p> <p>Tuning Reducing this timer value can improve performance with some non-IBM systems. However, this parameter can result in slightly increased system utilization.</p>
hstcp	<p>Purpose Enables the HighSpeed TCP as specified in RFC 3649. This parameter modifies the congestion control mechanism for use with TCP connections with large congestion windows to improve the average throughput.</p> <p>Tuning A value of 1 enables the HighSpeed TCP enhancements on a system-wide scale. A value of 0 disables it.</p>
icmp6_errmsg_rate	<p>Purpose Specifies the higher limit for the number of ICMP v6 error messages that can be sent per second. This parameter prevents excessive bandwidth from being used by ICMP v6 error messages.</p>
icmpaddressmask	<p>Purpose Specifies whether the system responds to an ICMP address mask request.</p> <p>Tuning If the value 0 is set, the network silently ignores any ICMP address mask request that it receives.</p>
icmptimestamp	<p>Purpose Specifies whether the system responds to an ICMP timestamp request.</p> <p>Tuning If the value of 0 is set, the network ignores any ICMP timestamp request that it receives.</p>
ie5_old_multicast_map ping	<p>Purpose Specifies IP multicasts on token ring that must be mapped to the broadcast address rather than a functional address when value 1 is used.</p>

Item	Description
ifstat32	<p>Purpose Enables or disables the 32-bit statistics. By default ifstat32 is disabled, when enabled it updates 32-bit statistics. Some applications might use 32-bit interface counters.</p> <p>Tuning A value of 0 disables it and a value of 1 enables it.</p>
ifsize	<p>Purpose Specifies the maximum number of network interface structures per interface of a single type. This limit does not apply to Ethernet interface structures for which the infrastructure expands dynamically to handle any number of Ethernet interface structures.</p> <p>Tuning The ifsize parameter must be large on systems that supports hotplug adapters and on DLPAR configurations because adapters can be added as required. The static interface tables must be large enough to accept the large number of adapters that is added for this system or partition. If the system detects at the start, that more adapters of a type are present than that is allowed by the current value of ifsize, it automatically increases the value to support the number of adapters present.</p>
ip6_defttl	<p>Purpose Specifies the default hop count that is used for IP version 6 packets if no other hop count is specified.</p>
ip6_prune	<p>Purpose Specifies how often to check the IP version 6 routing table for expired routes, in seconds.</p>
ip6forwarding	<p>Purpose Specifies whether the kernel must forward the IP version 6 packets.</p> <p>Tuning The default value of 0 prevents forwarding of ipv6 packets when they are not for the local systems. A value of 1 enables forwarding.</p>
ip6srcrouteforward	<p>Purpose Specifies whether the system forwards source-routed IP version 6 packets.</p> <p>Tuning A value of 1 allows the forwarding of source-routed packets. A value of 0 causes all source-routed packets that are not at their destinations to be discarded.</p>
ip_ifdelete_notify	<p>Purpose Specifies when an interface address is deleted. All the existing TCP connections that were bound locally to the interface address and were deleted must be notified with error ENETDOWN.</p> <p>Tuning Existing FTP/Telnet connections are disconnected when the ENETDOWN error is returned.</p>

Item	Description
ip_ifdelete_no_retrans	<p>Purpose Specifies that when an interface address is deleted, the existing TCP connections that were bound locally to the interface address must not retransmit data.</p> <p>Tuning No further retransmission of data occurs over the existing SSH connections.</p>
ip_nfrag	<p>Purpose Specifies the maximum number of fragments of an IP packet that can be kept on IP reassembly queue at a time.</p>
ipforwarding	<p>Purpose Specifies whether the kernel must forward packets.</p> <p>Tuning Set this parameter to 1, if the system is acting as an IP router.</p>
ipfragttl	<p>Purpose Specifies the time to live for IP fragments in half-seconds.</p> <p>Tuning Check for fragments that dropped after timeout (netstat -p ip). If the value of IP, that is, the fragments that are dropped after timeout, is nonzero, it can increase the ipfragttl parameter and reduce retransmissions.</p>
ipignoreredirects	<p>Purpose Specifies whether to process redirects that are received.</p> <p>Tuning A value of 0 processes redirects as usual. A value of 1 ignores redirects.</p>
ipqmaxlen	<p>Purpose Specifies the number of received packets that can be queued on the IP protocol input queue.</p> <p>Tuning Examine if ipintrq overflows (netstat -s) or use crash to access IP input queue overflow counter. Increase size if system is using many loopback sessions. Most operating system network drivers call IP directly and do not use the IP queue. Increasing the ipqmaxlen parameter on these devices has no effect.</p>
ipoutqueues	<p>Purpose Specifies whether to queue User Datagram Protocol (UDP) packets that are sent over IPv4. These UDP packets are handled by a separate kernel thread.</p> <p>Tuning The default value is 0 and it specifies the UDP to transmit the packet immediately without queuing. A nonzero value specifies the number of queues to be created and used. For example, to create a single queue that is used by the UDP, enter the following command:</p>

```
no -o ipoutqueues=1
```

Item	Description
ipsendredirects	<p>Purpose Specifies whether the kernel must send redirect signals.</p> <p>Tuning This parameter is a configuration decision with performance consequences.</p>
ipsrouteforward	<p>Purpose Specifies whether the system forwards source-routed packets.</p> <p>Tuning The default value of 1 allows the forwarding of source-routed packets. A value of 0 causes all source-routed packets that are not at their destinations to be discarded.</p>
ipsrouterecv	<p>Purpose Specifies whether the system accepts source-routed packets.</p> <p>Tuning The default value of 0 causes all source-routed packets that are destined for this system to be discarded. A value of 1 allows source-routed packets to be received.</p>
ipsroutesend	<p>Purpose Specifies whether applications can send source-routed packets.</p> <p>Tuning The default value of 1 allows source-routed packets to be sent. A value of 0 causes setsockopt() to return an error if an application attempts to set the source routing option, and removes any source routing options from the outgoing packets.</p>
limited_ss	<p>Purpose Enables the Limited SlowStart as specified in RFC 3742. This option limits the number of segments by which the congestion window is increased for one window during slow-start. This enhancement improves the performance for TCP connections with large congestion windows.</p> <p>Tuning A value from 1 to 100 enables the Limited SlowStart enhancements on a system-wide scale and sets it as the number of segments to the value of the maximum SlowStart threshold. A value of 0 disables it. The default value is 0.</p>
llsleep_timeout	<p>Purpose Specifies timeout value in seconds for link local timeouts (used when multi_homed=1).</p>
lo_perf	<p>Purpose Specifies whether you want to use a separate queue per CPU to improve loopback performance.</p> <p>Tuning A value of 1 enables a separate queue per CPU. A value of 0 disables this option.</p>

Item	Description
lowthresh	<p>Purpose Specifies the maximum number of bytes that can be allocated by using the allocb call for the BPRI_LO priority.</p> <p>Tuning When the total amount of memory that is allocated by the net_malloc call reaches this threshold, then the allocb request for the BPRI_LO priority returns 0. The lowthresh attribute represents a percentage of the thewall attribute and you can set its value from 0 to 100.</p>
main_if6	<p>Purpose Specifies the interface to use for link local addresses.</p>
main_site6	<p>Purpose Specifies the interface to use for site local address routing.</p>
maxnip6q	<p>Purpose Specifies the maximum number of IP version 6 packet reassembly queues.</p>
maxttl	<p>Purpose Specifies the time to live (in seconds) for RIP packets.</p>
medthresh	<p>Purpose Specifies the maximum number of bytes that can be allocated by using the allocb call for the BPRI_MED priority.</p> <p>Tuning When the total amount of memory that is allocated by the net_malloc call reaches this threshold, then the allocb request for the BPRI_MED priority returns 0. The medthresh attribute represents a percentage of the thewall attribute. A typical setting of 95 represents 95% of thewall attribute.</p>

Item	Description
mpr_policy	<p>Purpose Specifies the policy to be used for Multipath Routing.</p> <p>Tuning The following are the available routing policies:</p> <p>Weighted Round-Robin (1) Based on user-configured weights that are assigned to the multiple routes (through the route command) round-robin is applied. If no weights are configured, then it behaves identical to plain round-robin.</p> <p>Random (2) Chooses a route at random.</p> <p>Weighted Random (3) Chooses a route that is based on user-configured weights and a randomization routine. The policy adds up the weights of all the routes and picks a random number between 0 and total weight. Each of the individual weights is removed from the total weight until this number is zero. This policy picks a route in the range of the total number of routes available.</p> <p>Lowest Utilization (4) Chooses a route with the minimum number of current connections that goes through it.</p> <p>Hash-based (5) Hash-based algorithm chooses a route by hashing based on the destination IP address.</p>
multi_homed	<p>Purpose Specifies the level of multi-homed IP version 6 host support.</p> <p>Tuning Tuning is performed for connections whose destination port is specified in the list of the <code>delayackports</code> parameter. This parameter can be used to increase performance when communicating with an HTTP server. The parameter can have one of four values:</p> <p>0 Indicates the original functionality in AIX 4.3.</p> <p>1 Indicates that link local addresses is resolved by querying each interface for the link local address.</p> <p>2 Indicates that link local addresses is examined for the interface that is defined by <code>main_if6</code>.</p> <p>3 Indicates that link local addresses is examined for the interface that is defined by <code>main_if6</code> and site local addresses are routed to the <code>main_site6</code> interface.</p>

Item	Description
nbc_limit	<p>Purpose Specifies the total maximum amount of memory that can be used for the Network Buffer Cache.</p> <p>Tuning This attribute is in number of Kilobytes. When the cache grows to this limit, the rarely used cache objects are flushed out of the cache to make room for the new ones.</p>
nbc_max_cache	<p>Purpose Specifies the maximum size of the cache object that is allowed in the Network Buffer Cache without using the private segments.</p> <p>Tuning This parameter is in number of bytes. A data object bigger than this size is either cached in a private segment or is not cached at all.</p>
nbc_min_cache	<p>Purpose Specifies the minimum size of the cache object that is allowed in the Network Buffer Cache.</p> <p>Tuning This attribute is in number of bytes. A data object smaller than this size is not put into the NBC. This attribute applies for send_file() API and some web servers that use the get engine in the kernel.</p>
nbc_ofile_hashsz	<p>Purpose Specifies the size of the hash table that is used for hashing cache objects in the Network Buffer Cache.</p> <p>Tuning This hash table size applies to only opened file entries that is, entries that cache files from the file system. Since this attribute resizes the hash table size and affects the hashing of all existing entries, this attribute can be modified when the Network Buffer Cache is empty.</p>
nbc_pseg	<p>Purpose Specifies the maximum number of private segments that can be created for the Network Buffer Cache.</p> <p>Tuning When this option is set at nonzero, a data object between the size that is specified in nbc_max_cache and the segment size (256 MB) is cached in a private segment. A data object bigger than the segment size is not cached. When the maximum number of private segments exist, cache data in private segments can be flushed for new cache data so that the number of private segments do not exceed the limit. When nbc_pseg is set to 0, all cache in private segments is flushed.</p>

Item	Description
nbc_pseg_limit	<p>Purpose Specifies the maximum amount of cached data size that is allowed in private segments in the Network Buffer Cache.</p> <p>Tuning This value is expressed in Kilobytes. Since data that is cached in private segments is pinned by the Network Buffer Cache, nbc_pseg_limit controls the amount of pinned memory that is used for the Network Buffer Cache in addition to the network buffers in global segments. When the amount of cached data reaches this limit, cache data in private segments can be flushed for new cache data so that the total pinned memory size does not exceed the limit. When nbc_pseg_limit is set to 0, all cache in private segments is flushed.</p>
ndd_event_name	<p>Purpose Specifies the list of interface names for ns_alloc and ns_free events to be captured, when the trace of ns_alloc/ns_free events is enabled by setting the ndd_event_tracing option.</p>
ndd_event_tracing	<p>Purpose Specifies the size of the ns_alloc/ns_free trace buffer.</p> <p>Tuning If the value of this option is nonzero all ns_alloc and ns_free events are traced in a kernel buffer. A value of zero disables this event tracing. If the values of ndd_event_tracing are larger than 1024, it allocates as many items in the kernel buffer for tracing.</p>
ndp_mmaxtries	<p>Purpose Specifies the maximum number of Multicast NDP Neighbor Discovery Protocol (NDP) packets to send.</p>
ndp_umaxtries	<p>Purpose Specifies the maximum number of Unicast Neighbor Discovery Protocol (NDP) packets to send.</p>
ndpqsize	<p>Purpose Specifies the number of packets to hold waiting on completion of a Neighbor Discovery Protocol (NDP) entry that is used by IP version 6.</p>
ndpt_down	<p>Purpose Specifies the time, in half seconds, to hold down an NDP entry.</p>
ndpt_keep	<p>Purpose Specifies the time, in half seconds, to keep a Neighbor Discovery Protocol (NDP) entry.</p>
ndpt_probe	<p>Purpose Specifies the time in half seconds to delay before the first Neighbor Discovery Protocol (NDP) probe is sent.</p>
ndpt_reachable	<p>Purpose Specifies the time, in half seconds, to test if a Neighbor Discovery Protocol (NDP) entry is still valid.</p>

Item	Description
ndpt_retrans	<p>Purpose Specifies the time, in half seconds, to wait before an NDP request is retransmitted.</p>
net_buf_size	<p>Purpose Specifies a list of buffer sizes for <code>net_malloc</code> and <code>net_free</code> events to be captured.</p> <p>Tuning The <code>net_buf_size</code> strings represent a list of sizes. If this attribute is not of value <code>all</code>, only <code>net_malloc</code> and <code>net_free</code> events of those sizes are captured. A value of <code>all</code> means that the events of any size are captured.</p>
net_buf_type	<p>Purpose Specifies a list of buffer types for <code>net_malloc</code> and <code>net_free</code> events to be captured.</p> <p>Tuning The <code>net_buf_type</code> string represents a list of types. If the string is not empty and different from <code>all</code>, only <code>net_malloc</code> and <code>net_free</code> events of that type is captured.</p>
net_malloc_frag_mask	<p>Purpose It is used as boolean attribute for mask with each bucket that requests similar fragments to be promoted to full pages.</p> <p>Tuning Allows promotion of allocations smaller than 1 page to full pages for better detection of memory overwrite problems. It is a mask for each bucket size that requests such fragments to be promoted to full pages. Enabling this option for memory fragments results in low performance.</p>
netm_page_promote	<p>Purpose Specifies whether to allow promotion of a fragment to page size.</p> <p>Tuning This option allows promotion of fragment sizes that are specified in <code>net_malloc_frag_mask</code> to page size. Setting this option to 0, disables the page promotion irrespective of the sizes that are set in <code>net_malloc_frag_mask</code>.</p>
nonlocsrcroute	<p>Purpose Tells the Internet Protocol that strictly source-routed packets can be addressed to hosts outside the local network.</p> <p>Tuning A value of 0 disallows addressing to outside hosts. A value of 1 allows packets to be addressed to outside hosts. Loosely source-routed packets are not affected by this attribute.</p>
nstrpush	<p>Purpose Specifies the maximum number of modules that you can push onto a single stream. The minimum value is 8.</p> <p>Tuning This parameter is read-only. This attribute can be set when loading the operating system in the <code>/etc/pse_tune.conf</code> file.</p>

Item	Description
passive_dgd	<p>Purpose Specifies whether Passive Dead Gateway Detection is enabled.</p> <p>Tuning A value of 0 disables passive_dgd, and a value of 1 enables it for all gateways in use.</p>
pmtu_default_age	<p>Purpose This option is now unused because UDP applications are now required to always set IP_DONTFRAG socket option to be able to detect decreases in Path MTU.</p> <p>Tuning A value of zero allows no aging. The default value is 10 minutes. The pmtu_default_age value can be overridden by UDP applications. pmtu_default_age is a runtime attribute.</p>
pmtu_expire	<p>Purpose Specifies the default amount of time (in minutes) before which the path MTU entries with reference count of zero are deleted.</p> <p>Tuning A value of 0 suggests that the pmtu entries do not expire.</p>
pmtu_rediscover_interval	<p>Purpose Specifies the default amount of time (in minutes) before the path MTU value for UDP and TCP paths are checked for a higher value.</p> <p>Tuning A value of 0 allows no path MTU rediscovery.</p>
psebufcalls	<p>Purpose Specifies the maximum number of bufcalls to allocate by Streams.</p> <p>Tuning The Stream subsystem allocates certain number of bufcall structures at initialization, so that when the allocb call fails, the user can register their requests for the bufcall. You are not allowed to decrease this value until the system is restarted. During restart, the parameter returns to its default value.</p>
psecache	<p>Purpose Controls the number of stream buffers.</p>
psetimers	<p>Purpose Specifies the maximum number of timers to allocate by Streams.</p> <p>Tuning The Stream subsystem allocates a certain number of timer structures at initialization so that the STREAMS driver or module can register their timeout calls. You are not allowed to decrease this value until the system is restarted. During restart, the parameter returns to its default value.</p>

Item	Description
>> ras_art	<p>Purpose Specifies the maximum number of socket reliability, availability, and serviceability (RAS) artifacts that are maintained.</p> <p>Tuning The valid value of this tunable parameter is in the range 0 - 10000. The default value is 200. A value of 0 disables this tunable parameter.</p>
rfc1122addrchk	<p>Purpose Performs address validation as specified by RFC1122, Requirements for Internet Hosts-Communication Layers.</p> <p>Tuning A value of 0 does not perform address validation. A value of 1 performs address validation.</p>
rfc1323	<p>Purpose Enables TCP enhancements as specified by RFC 1323, TCP Extensions for High Performance.</p> <p>Tuning A value of 0 disables the RFC enhancements on a system-wide scale. A value of 1 specifies that all TCP connections attempt to negotiate the RFC enhancements. The SOCKETS application can override the default behavior on individual TCP connections, by using the setsockopt subroutine. The rfc1323 network option can also be set on a per interface basis through the ifconfig command.</p>
rfc2414	<p>Purpose Enables the increasing of TCPs initial window as described in RFC 2414.</p> <p>Tuning When it is on, the initial window depends on setting the tcp_init_window tunable.</p>
roce_v2_mode	<p>Purpose Enables RDMA over Converged Ethernet (RoCE) protocol mode. This tunable enables RoCE version 1 (RoCE v1) or RoCE version 2 in Open Fabrics enterprise Distribution (OFED) and device driver.</p> <p>Tuning The default value of 0 enables RoCE v1 protocol mode, and a value of 1 enables RoCE v2 protocol mode.</p>
route_expire	<p>Purpose Specifies whether the route expires.</p> <p>Tuning A value of 0 allows no route expiration. Negative values are not allowed for this option.</p>

Item	Description
routerevalidate	<p>Purpose Specifies that each cached route of a connection must be validated when a new route is added to the routing table.</p> <p>Tuning This option ensures that applications that keep the same connection open for long periods of time (for example NFS) uses the correct route after routing table changes occur. A value of 0 does not revalidate the cached routes. Turning on this option can cause some performance degradation.</p>
rto_high	<p>Purpose Specifies the TCP Retransmit Time out high value that is used in calculating factors, and the allowable maximum retransmits that is used in TCP data segment retransmits.</p> <p>Tuning rto_high is the high factor.</p>
rto_length	<p>Purpose Specifies the TCP Retransmit Time Out length value that is used in calculating factors, and the allowable maximum retransmits that is used in TCP data segment retransmits.</p> <p>Tuning rto_length is the total number of time segments.</p>
rto_limit	<p>Purpose Specifies the TCP Retransmit Time out limit value that is used in calculating factors, and the allowable maximum retransmits that is used in TCP data segment retransmits.</p> <p>Tuning rto_limit is the number of time segments from rto_low to rto_high.</p>
rto_low	<p>Purpose Specifies the TCP Retransmit Time Out low value that is used in calculating factors, and the allowable maximum retransmits that is used in TCP data segment retransmits.</p> <p>Tuning rto_low is the low factor.</p>
sack	<p>Purpose Enables TCP Selective Acknowledgment as described in RFC 2018.</p> <p>Tuning A value of 1 makes all TCP connections negotiate sack. Default is zero, which disables the negotiation. The sack feature needs support from the peer TCP. The negotiation phase during connection initiation determines that. When out of order segments are received, Selective Acknowledgment from the receiver informs the sender of the data that is received so that the sender can retransmit only the missing segments. This action results in less unnecessary retransmitted segments. Sack is useful for recovering fast from multiple packets drop in a window of data.</p>

Item	Description
sb_max	<p>Purpose Specifies the maximum buffer size that is allowed for a TCP and UDP socket. Limits setsockopt, udp_sendspace, udp_recvspace, tcp_sendspace, and tcp_recvspace.</p> <p>Tuning Increase size, preferably to multiple of 4096. Must be approximately two to four times the largest socket buffer limit.</p>
send_file_duration	<p>Purpose Specifies the cache validation duration for all the file objects that system call <code>send_file</code> accessed in the network buffer cache.</p> <p>Tuning This attribute is in number of seconds. A value of 0 means that the cache is validated for every access.</p>
site6_index	<p>Purpose Specifies the maximum interface number for site local routing.</p>
sockthresh	<p>Purpose Specifies the maximum amount of network memory that can be allocated for sockets. Used to prevent new sockets or TCP connections from exhausting all MBUF memory and reserve the remaining memory for the existing sockets or TCP connections.</p> <p>Tuning When the total amount of memory that is allocated by the net_malloc subroutine reaches the sockthresh threshold, the socket and socketpair system calls fail with an error of ENOBUFS. Incoming connection requests are silently discarded. Existing sockets can continue to use more memory. The sockthresh attribute represents a percentage of the thewall attribute.</p>
sodebug	<p>Purpose Specifies whether the newly created sockets have SO_DEBUG flag on.</p>
sodebug_env	<p>Purpose Specifies whether SODEBUG process environment variable is checked for the newly created sockets; if so, these sockets have the SO_DEBUG flag on.</p>
somaxconn	<p>Purpose Specifies the maximum listen backlog.</p> <p>Tuning Increase this parameter on busy web servers to handle peak connection rates.</p>

Item	Description
<p>>> lsoreuseport_lb</p>	<p>Purpose Specifies whether the SO_REUSEPORT socket option is enabled or disabled for load balancing.</p> <p>Tuning This tunable parameter can have the following values:</p> <ul style="list-style-type: none"> • 1 - Enables the SO_REUSEPORT socket option. • 0 - Disables the SO_REUSEPORT socket option. <p><<<</p>
<p>strctlsz</p>	<p>Purpose Specifies the maximum number of bytes of information that a single system call can pass to a Stream to place into the control part of a message (in an M_PROTO or M_PCPRTO block).</p> <p>Tuning The putmsg call with a control part that exceeds this size fails with ERANGE.</p>
<p>strmsgsz</p>	<p>Purpose Specifies the maximum number of bytes of information that a single system call can pass to a Stream to place into the data part of a message (in M_DATA blocks).</p> <p>Tuning Any write call that exceeds this size is broken into multiple messages. The putmsg call with a data part that exceeds this size fails with ERANGE.</p>
<p>strthresh</p>	<p>Purpose Specifies the maximum number of bytes Streams are normally allowed to allocate.</p> <p>Tuning When the threshold is passed, strthresh does not allow users without the appropriate privilege to open Streams, push modules, or write to Stream devices, and returns ENOSR. The threshold applies to the output and does not affect the data that comes into the system (for example, console continues to work properly). A value of zero means that there is no threshold. The strthresh attribute represents a percentage of the thewall attribute. The thewall attribute indicates the maximum number of bytes that can be allocated by Streams and Sockets by using the net_malloc call.</p>

Item	Description
strturncnt	<p>Purpose Specifies the maximum number of requests that are handled by the current running thread for Module or Elsewhere level Streams synchronization.</p> <p>Tuning The Module level synchronization works in a way that only one thread can run in the module at any time and all other threads, which try to acquire the same module enqueues their requests and leave. After the current running thread completes its work, it dequeues all the previously enqueued requests one by one and runs them. If there are many requests that are enqueued in the list, then the current running thread has to serve everyone and is always busy serving others and starves itself. To avoid this situation, the current running thread serves only the strturncnt number of threads after that a separate kernel thread activates and runs all the pending requests.</p>
subnetsarelocal	<p>Purpose Specifies whether all subnets that match the subnet mask are to be considered local for purposes of establishing, for example, the TCP maximum segment size.</p> <p>Tuning This parameter is used by the in_localaddress subroutine. The default value, 1 specifies that addresses that match the local network mask are local. If the value is 0, addresses that match the local subnetwork are local. This option is a configuration decision with performance consequences. If all the subnets do not have the same MTU, fragmentation at bridges can degrade performance. If the subnets do have the same MTU, and subnetsarelocal is 0, TCP sessions can use a small MSS.</p>
tcp_bad_port_limit	<p>Purpose Specifies the number of TCP segments to a port, which does not have a socket connection, within the time duration of half a second. TCP stops sending TCP reset segments in response after this time.</p> <p>Tuning If the value is set to 0, TCP indicates a bad port number error by sending TCP reset segments. A value greater than 0 indicates the number of TCP segments that are received by a port, which does not have a socket connection, within the time duration of half a second before TCP stops sending TCP reset segments.</p>
tcp_cubic	<p>Purpose Enables CUBIC, as specified in RFC 8312, by modifying the congestion control mechanism for TCP connections between networks that have large bandwidth to improve the average throughput.</p> <p>Tuning A value of 1 enables the CUBIC TCP enhancements for all systems in the network. A value of 0 disables it. The default value is 0.</p>

Item	Description
tcp_cwnd_modified	<p>Purpose Allows the TCP IP applications with specific socket options to adjust the network congestion window. This parameter might be used only in a specific wide area network (WAN) environment.</p> <p>Tuning Default value is 0, which disables the tuning parameter. Tuning it to a value of 1 allows you to adjust the network congestion window.</p>
tcp_ecn	<p>Purpose Enables TCP level support for Explicit Congestion Notification as described in RFC 2481.</p> <p>Tuning Default is off (0). Turning it on (1) makes all connections negotiate ECN capability with the peer. For this feature to work, you need support from the peer TCP and also IP level ECN support from the routers in the path.</p>
tcp_ephemeral_high	<p>Purpose Specifies the largest port number to allocate for TCP ephemeral ports.</p> <p>Tuning The number of ephemeral sockets is determined by tcp_ephemeral_high minus tcp_ephemeral_low. For maximum number of ephemeral sockets, set tcp_ephemeral_high to 65535 and tcp_ephemeral_low to 1024.</p>
tcp_ephemeral_low	<p>Purpose Specifies the smallest port number to allocate for TCP ephemeral ports.</p> <p>Tuning The number of ephemeral sockets is determined by tcp_ephemeral_high minus tcp_ephemeral_low. For maximum number of ephemeral sockets, set tcp_ephemeral_high to 65535 and tcp_ephemeral_low to 1024.</p>
tcp_fastlo	<p>Purpose Allows the TCP loopback traffic to cutoff the entire TCP/IP stack protocol and interface to achieve better performance.</p> <p>Tuning A value of 1 enables the TCP loopback traffic to cutoff the entire TCP/IP stack. A value of 0 disables this option.</p>
tcp_finwait2	<p>Purpose Specifies the length of time to wait in the FIN_WAIT2 state before closing the connection, which is measured in half seconds.</p>

Item	Description
tcp_icmpsecure	<p>Purpose Specifies whether or not ICMP (Internet Control Message Protocol) attacks on TCP are avoided.</p> <p>Tuning This option must be turned on to protect TCP connections against ICMP attacks. The ICMP attacks might be of the form of ICMP source quench attacks and PMTUD (Path MTU Discovery) attacks. If this network option is turned on, the system does not react to ICMP source quench messages. This option protects against ICMP source quench attacks. Also, if this network option is enabled, the payload of the ICMP message is tested to determine whether the sequence number of the TCP header portion of the payload is within the range of acceptable sequence numbers. This option mitigates PMTUD attacks to a large extent.</p>
tcp_init_window	<p>Purpose This value is used only when rfc2414 is turned on (ignored otherwise).</p> <p>Tuning If rfc2414 is on and this value is zero, then the initial window computation is done according to rfc2414. If this value is nonzero, the initial (congestion) window is initialized a number of maximum sized segments equal to tcp_init_window. Changing the tcp_init_window option allows you to tune the TCP slow start to control the number of TCP segments (packets) outstanding before an ACK is received. For example, setting this value to 6 would allow 6 packets to be sent initially, instead of the normal 2 or 3 packets, thus speeding up the initial packet rate.</p>
tcp_inpcb_hashtab_siz	<p>Purpose Specifies the size of the inpcb hash table for TCP connections.</p> <p>Tuning This table holds the inpcb required for connection management and is implemented as a table of hash chains. A larger table means that the linked hash chains are smaller and less traversal time on the average but the memory footprint is larger. This value must be a prime number. This option impacts performance and must be used with extreme caution. Consult a performance analyst in case that it is felt that the value needs to be changed. The execution environment could have an influence on the value. It is encouraged to maintain the system-defined defaults as they tend to execute optimally in most environments.</p>
tcp_keeppcnt	<p>Purpose tcp_keeppcnt represents the number of keepalive probes that could be sent before terminating the connection.</p>
tcp_keepidle	<p>Purpose Specifies the length of time to keep the connection active, measured in half seconds.</p>
tcp_keepinit	<p>Purpose Sets the initial timeout value for a TCP connection, which is measured in half seconds.</p>

Item	Description
tcp_keepintvl	<p>Purpose Specifies the interval, which is measured in half seconds, between packets that are sent to validate the connection.</p> <p>Tuning For example, 150 half seconds results in 75 seconds between validation probes. This option allows TCP to know that a connection is still valid and keep the connection open when it is otherwise idle. This option is a configuration decision with minimal performance consequences. No change is recommended. If the interval are shortened significantly, processing and bandwidth costs might become significant.</p>
tcp_limited_transmit	<p>Purpose Enables the feature that enhances TCP's loss recovery as described in the RFC 3042.</p> <p>Tuning A value of 1 enables this option and zero disables the option.</p>
tcp_low_rto	<p>Purpose Specifies the TCP retransmit timeout (RTO) in milliseconds for connections that are experiencing packet drops.</p> <p>Tuning A tick is 10 ms (one 100th of a second). The option timer_wheel_tick must be set to nonzero value before setting the tcp_low_rto option. Also, tcp_low_rto can be equal to zero or a multiple of ten times the timer_wheel_tick value. This tunable allows TCP to use smaller timeout values for packet timeout and retransmit on high-speed networks. Normal TCP retransmit timeout is 1.5 seconds.</p>
tcp_maxburst	<p>Purpose Specifies the number of back-to-back packets that TCP can send before pausing to allow those packets to be forwarded to their destination.</p> <p>Tuning This option can be useful if routers are unable to handle large bursts of TCP packets and are dropping some of them. A value of 0 means no limitation for back-to-back packets before pausing.</p>
tcp_maxqueuelen	<p>Purpose Specifies the maximum number of TCP segments that can be processed in the reassembly queue.</p> <p>Tuning Values for this tunable parameter are in the range 0 - 32767. A value of 0 means unlimited queue length. The default value is 1000.</p>

Item	Description
tcp_mssdfmt	<p>Purpose Default maximum segment size that is used in communicating with remote networks.</p> <p>Tuning tcp_mssdfmt is only used if path MTU discovery is not enabled or path MTU discovery fails to discover a path MTU. The tcp_mssdfmt network option can also be set on a per interface basis (see the documentation for ISNO options). Limiting data to (MTU - 40) bytes ensures that, where possible, only full packets are sent.</p>
tcp_nagle_limit	<p>Purpose This parameter is the Nagle algorithm threshold in bytes, which can be used to disable Nagle.</p> <p>Tuning The default is Nagle turned on. To disable Nagle, set this value to 0 or 1. TCP disables Nagle for data segments larger than or equal to this threshold value.</p>
tcp_nagleoverride	<p>Purpose Setting the option tcp_nagle_limit turns off the Nagle algorithm system wide and setting tcp_nodelay option for a socket turns off the Nagle algorithm for that specific connection whereas setting tcp_nagleoverride disables the Nagle algorithm only for certain situations during the connection.</p> <p>Tuning The value of 1 disables Nagle algorithm only for certain TCP packets in a connection.</p>
tcp_ndebug	<p>Purpose Specifies the number of tcp_debug structures.</p>
tcp_newreno	<p>Purpose Enables the modification to TCP's Fast Recovery algorithm as described in RFC 2582.</p> <p>Tuning This option fixes the limitation of TCP's Fast Retransmit algorithm to recover fast from dropped packets when multiple packets in a window are dropped. The sack also achieves the same thing but sack needs support from both ends of the TCP connection. The NewReno modification is only on the sender side.</p>
tcp_nodelayack	<p>Purpose Turning this parameter on causes TCP to send immediate acknowledgment (Ack) packets to the sender. When tcp_nodelayack is disabled, TCP delays sending Ack packets by up to 200 ms. This option allows the Ack to be piggy-backed onto a response and minimizes system overload.</p> <p>Tuning This option can be used to overcome bugs in other implementations of the TCP Nagle algorithm. Setting this option to 1 causes slightly more system overload, but can result in higher performance for network transfers if the sender is waiting on the receiver's acknowledgment.</p>

Item	Description
tcp_pmtu_discover	<p>Purpose Enables or disables path MTU discovery for TCP applications.</p> <p>Tuning A value of 0 disables path MTU discovery for TCP applications, while a value of 1 enables it.</p>
tcp_recvspace	<p>Purpose Specifies the system default socket buffer size for receiving data. This option affects the window size that is used by TCP.</p> <p>Tuning The optimum buffer size is the product of the media bandwidth and the average round-trip time of a packet. The tcp_recvspace network option can also be set on a per interface basis (reference documentation on Interface Specific Network Options (ISNO)). Most interfaces now have this tunable set in the ISNO defaults. The tcp_recvspace attribute must specify a socket buffer size less than or equal to the setting of the sb_max attribute.</p>
tcp_sendspace	<p>Purpose Specifies the system default socket buffer size for sending data.</p> <p>Tuning The optimum buffer size is the product of the media bandwidth and the average round-trip time of a packet: $\text{optimum_window} = \text{bandwidth} * \text{average_round_trip_time}$. The tcp_sendspace network option can also be set on a per interface basis (reference documentation on Interface Specific Network Options (ISNO)). Most interfaces now have this tunable set in the ISNO defaults. The tcp_sendspace attribute must specify a socket buffer size less than or equal to the setting of the sb_max attribute.</p>
tcp_syn_rto	<p>Purpose Specifies the TCP retransmission timeout (RTO) value, in interval of half-seconds, for a connection that experiences packet drops before the connection is established.</p> <p>Tuning The value of the tcp_syn_rto tunable parameter is set as the initial retransmission timeout value for retransmissions that occur before the connection is established. The values are in the range 0 - 32767. The default value is 0.</p>

Item	Description
tcp_tcpsecure	<p>Purpose Specifies whether connection reset attacks and data corruption attacks on TCP are avoided.</p> <p>Tuning This option is used to protect TCP connections from one or more of the following three vulnerabilities. The first vulnerability involves sending of a fake SYN to an established connection to abort the connection. A tcp_tcpsecure value of 1 provides protection from this vulnerability. The second vulnerability involves the sending of a fake RST to an established connection to abort the connection. A tcp_tcpsecure value of 2 provides protection from this vulnerability. The third vulnerability involves injecting fake data in an established TCP connection. A tcp_tcpsecure value of 4 provides protection from this vulnerability. Values for tcp_tcpsecure can range from a minimum of 0 (this value is the default value and provides no protection from these vulnerabilities) to a maximum value of 7. Values of 3, 5, 6, or 7 protects the connection from combinations of these three vulnerabilities.</p>
tcp_timewait	<p>Purpose The tcp_timewait option is used to configure how long connections are kept in the <code>timewait</code> state.</p> <p>Tuning It is given in 15-second intervals. Increasing this value degrades performance of web servers or applications that open and close many TCP connections.</p>
tcp_ttl	<p>Purpose Specifies the time to live for TCP packets, expressed in ticks.</p> <p>Tuning A tick is 0.6 seconds (there are 100 ticks per minutes).</p>
tcprexmtthresh	<p>Purpose Specifies the number of consecutive duplicate acknowledgments, which cause TCP to goto fast retransmit phase.</p> <p>Tuning Increase this parameter if TCP performance is low due to an increased number of duplicate acknowledgments but the network is not congested. Setting a high value for this option can cause TCP to time out and retransmit.</p>
tcptr_enable	<p>Purpose Enables TCP traffic regulation that is defined by policies that created by using the tcptr command. A value of 0 means disabled. Any nonzero value means that traffic regulation is enabled.</p> <p>Tuning A value of 0 disables this option. This option must be turned on for servers that must protect against network attacks.</p>

Item	Description
thewall	<p>Purpose Specifies the maximum amount of memory, in kilobytes that is allocated to the memory pool.</p> <p>Tuning Cannot be set anymore.</p>
timer_wheel_tick	<p>Purpose Specifies the slot interval of the timer wheel, in ticks, where a tick=1000/HZ=10 ms.</p> <p>Tuning This attribute is used with tcp_low_rto attribute to reduce the TCP timeout values to smaller units.</p>
udp_bad_port_limit	<p>Purpose Specifies the number of UDP packets to a port with no socket that can be received in a 500-millisecond period before UDP stops sending ICMP errors in response to such packets.</p> <p>Tuning If set to 0, ICMP errors are always sent when UDP packets are received for a bad port number. If greater than 0, it specifies the number of packets to be received before UDP stops sending ICMP errors.</p>
udp_ephemeral_high	<p>Purpose Specifies the largest port number to allocate for UDP ephemeral ports.</p>
udp_ephemeral_low	<p>Purpose Specifies the smallest port number to allocate for UDP ephemeral ports.</p>
udp_inpcb_hashtab_siz	<p>Purpose Specifies the size of the inpcb hash table for UDP connections. This table holds the inpcbs that is required for connection management and is implemented as a table of hash chains. A larger table means that the linked hash chains are smaller and less traversal time on the average but the memory footprint is larger.</p> <p>Tuning This value must be a prime number. This option impacts performance and must be used with extreme caution. Consult a performance analyst in case that it is felt that the value must be changed. The execution environment can have an influence on the value. It is encouraged to maintain the system-defined defaults as they tend to run optimally in most environments.</p>
udp_pmtu_discover	<p>Purpose Enables or disables path MTU discovery for UDP applications.</p> <p>Tuning UDP applications must be written to use path MTU discovery. A value of 0 disables the feature, while a value of 1 enables it.</p>

Item	Description
udp_recvspace	<p>Purpose Specifies the system default socket buffer size for receiving UDP data.</p> <p>Tuning Change when nonzero n in netstat -s report of udp: n socket buffer overflows. The udp_recvspace parameter must specify a socket buffer size less than or equal to the setting of the sb_max parameter. Increase size, preferably to multiple of 4096.</p>
udp_send_perf	<p>Purpose Improves the UDP Transmit performance by caching address information and Memory Buffers (mbufs) that are used to transmit packets over a network.</p> <p>Tuning The default value is 0 and it disables caching. To enable caching, specify a value of 1. For example, to enable caching, enter the following command:</p> <pre style="background-color: #f0f0f0; padding: 5px;">no -o udp_send_perf=1</pre>
udp_sendspace	<p>Purpose Specifies the system default socket buffer size (in bytes) for sending UDP data.</p> <p>Tuning The udp_sendspace attribute must specify a socket buffer size less than or equal to the setting of the sb_max attribute. udp_sendspace must be at least as large as the largest datagram size that the application sends. Increase size, preferably to multiple of 4096.</p>
udp_ttl	<p>Purpose Specifies the time to live (in seconds) for UDP packets.</p>
udpcksum	<p>Purpose Allows UDP checksum to be turned on or off.</p> <p>Tuning A value of 0 turns it off and a value of 1 turns it on.</p>
> unix_sock_buf <	<p>> </p> <p>Purpose Specifies the default system-socket buffer size (in bytes) for UNIX domain socket buffers.</p> <p>Tuning The unix_sock_buf attribute must specify a system-socket buffer size that is less than or equal to buffer size that is specified by using the sb_max attribute. By default, the value of unix_sock_buf attribute is set to 16 KB (16384 bytes). You can increase the buffer size to a multiple of 4096.</p> <p> <</p>

Item	Description
use_sndbufpool	<p data-bbox="617 168 1481 210">Purpose Enables caching of mbuf clusters to improve performance.</p> <p data-bbox="617 210 1481 252">Tuning If this value is disabled, then to allocate a mbuf cluster, AIX allocates a cluster buffer and also a mbuf buffer to point to it, thus requiring two buffer allocation operations. Likewise, to free the cluster, two buffer free operations are required. With this option enabled, AIX maintains a cache of clusters for each cluster size that is being used. This option improves performance by reducing overload to allocate and free mbuf clusters. The default value of 1 enables this option on a system-wide scale. The mbuf cluster cache can be displayed by using the netstat -M command.</p>

Compatibility Mode

When running in pre 5.2 compatibility mode that is controlled by the **pre520tune** attribute of `sys0`, see AIX 5.2 compatibility mode. The reboot values for parameters, except those of type `Bosboot`, are not applicable because in the pre 5.2 compatibility mode they are not applied during boot.

In pre 5.2 compatibility mode, setting reboot values to tuning parameters continues to be achieved by embedding calls to tuning commands in scripts that are called during the boot sequence. Therefore, 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 5L Version 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. See Kernel Tuning in the *Performance Tools Guide and Reference* for details.

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 maximum size of the mbuf pool, type the following command:

```
no -o thewall
```

2. To reset the time to live for UDP packets its default size, type the following command:

```
no -d udp_ttl
```

3. To change the default socket buffer sizes on your system, type the following command:

```
no -r -o tcp_sendspace=32768
no -r -o udp_recvspace=32768
```

4. To use a system as an internet work router over Internet Protocol networks, type the following command:

```
no -o ipforwarding=1
```

5. To list the current and reboot value, range, unit, type and dependencies of all tunable parameters that are managed by the **no** command, type the following command:

```
no -L
```

6. To display the help information about the `udp_ephemeral_high` option, type the following command:

```
no -h udp_ephemeral_high
```

7. To permanently turn off the `ip6srcrouteforward` option, type the following command:

```
no -p -o ip6srcrouteforward=0
```

8. To list the reboot values for all Network tuning parameters, type the following command:

```
no -r -a
```

9. To list (spreadsheet format) the current and reboot value, range, unit, type and dependencies of all tunable parameters that are managed by the **no** command, type the following command:

```
no -x
```

10. To log all allocations and frees of type `mbuf` or `socket` that are size 256 or 4096, type the following command:

```
no -o net_buf_type={mbuf:socket} -o net_buf_size={256:4096} -o net_malloc_police=1
```

11. To log all allocations and frees of type `mbuf`, type the following command:

```
no -o net_buf_type={mbuf} -o net_buf_size={all} -o net_malloc_police=1
```

12. To log all `ns_alloc` and `ns_free` for `en0` or `en3` by using a 2000 events buffer size, type the following command:

```
no -o ndd_event_name={en0:en3} -o ndd_event_tracing=2000
```

13. To log all `ns_alloc` and `ns_free` for all `en` adapters by using a 2000 events buffer size, type the following command:

```
no -o ndd_event_name={en} -o ndd_event_tracing=2000
```

14. To log all `ns_alloc` and `ns_free` for all adapters, type the following command:

```
no -o ndd_event_name={all} -o ndd_event_tracing=1
```

nohup Command

Purpose

Runs a command without hangups.

Syntax

```
nohup { -p pid | Command [ Arg ... ] [ & ] }
```

Description

The **nohup** command runs the command specified by the *Command* parameter and any related *Arg* parameters, ignoring all hang up signals (SIGHUP) or modifies the process specified with the **-p** option to ignore all SIGHUP signals. SIGHUP is a signal that is sent to a process when the controlling terminal of the process is closed.

The **nohup** command can also be used to run programs in the background after logging off. To run a nohup command in the background, add an & (ampersand) to the end of the command.

If the standard error is displayed on the terminal and if the standard output is neither displayed on the terminal, nor sent to the output file specified by the user (the default output file is nohup.out), both the ./nohup.out and \$HOME/nohup.out files are not created or opened for appending the error message. The **nohup** command does not execute the parameter utility that is specified with the **nohup** command and exits with exit status 127.

Note: The **-p pid** and *Command* options can not be specified together.

When the **-p pid** flag is used, the output of the specified process will not be re-directed to the nohup.out file.

Flags

Item	Description
-p pid	pid is the process-id of a running process. The nohup command modifies the specified process, to ignore all hangup (SIGHUP) signals.

Exit Status

This command returns the following exit values:

Item	Description
126	The command specified by the <i>Command</i> parameter was found but could not be invoked.
127	An error occurred in the nohup command or the command specified by the <i>Command</i> parameter could not be found.

Otherwise, the exit status of the nohup command is that of the command specified by the *Command* parameter.

Examples

1. To run a command in the background after you log off, enter:

```
$ nohup find / -print &
```

After you enter this command, the following is displayed:

```
670
$ Sending output to nohup.out
```

The process ID number changes to that of the background process started by & (ampersand). The message `Sending output to nohup.out` informs you that the output from the **find / -print** command is in the **nohup.out** file. You can log off after you see these messages, even if the `find` command is still running.

2. To run a command in the background and redirect the standard output to a different file, enter:

```
$ nohup find / -print >filenames &
```

This example runs the **find / -print** command and stores its output in a file named `filenames`. Now only the process ID and prompt are displayed:

```
677
$
```

Wait before logging off because the nohup command takes a moment to start the command specified by the *Command* parameter. If you log off too quickly, the command specified by the *Command*

parameter may not run at all. Once the command specified by the *Command* parameter starts, logging off does not affect it.

3. To run more than one command, use a shell procedure. For example, if you write the shell procedure:

```
neqn math1 | nroff > fmath1
```

and name it the `nnfmath1` file, you can run the `nohup` command for all of the commands in the `nnfmath1` file with the command:

```
nohup sh nnfmath1
```

4. If you assign execute permission to the `nnfmath1` file, you get the same results by issuing the command:

```
nohup nnfmath1
```

5. To run the `nnfmath1` file in the background, enter:

```
nohup nnfmath1  
&
```

6. To run the `nnfmath1` file in the Korn shell, enter:

```
nohup ksh nnfmath1
```

7. To make a running process ignore all hangup signals, enter:

```
nohup -p 161792
```

enotifyevent Command, notifyevent Command

Purpose

Mails event information generated by the event response resource manager (ERRM) to a specified user ID.

Syntax

```
enotifyevent [-h] [user-ID]
```

```
notifyevent [-h] [user-ID]
```

Description

The `enotifyevent` script always return messages in English. The language in which the messages of the `notifyevent` script are returned depend on the locale settings.

These scripts capture event information that is posted by the event response resource manager (ERRM) in environment variables that are generated by the ERRM when an event occurs. These scripts can be used as actions that are run by an event response resource. They can also be used as templates to create other user-defined actions.

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.

In AIX, these scripts use the `mail` command to send event information to the specified user ID. When a user ID is specified, it is assumed to be valid, and it is used without verifying it. If a user ID is not specified, the user who is running the command is used as the default.

user-ID is the optional ID of the user to whom the event information will be mailed. If *user-ID* is not specified, the user who is running the command is used as the default.

Flags

-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.

For AIX, 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.

For other platforms, the size of the *log_file* is not limited, and it will not overwrite itself. The file size will increase indefinitely unless the administrator periodically removes 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

Command has run successfully.

Restrictions

1. These scripts must be run on the node where the ERRM is running.
2. The mail command is used to read the file.

Standard Output

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

Examples

1. You can use the **mail** command to read the contents of the event information. The following example shows how a warning event for the /var file system (a file system resource) is formatted and logged:

```
=====  
Event reported at Sun Mar 26 16:38:03 2002  
  
Condition Name:                /var space used  
Severity:                      Warning  
Event Type:                    Event  
Expression:                    PercentTotUsed>90  
  
Resource Name:                 /var  
Resource Class Name:          IBM.FileSystem  
Data Type:                    CT_UINT32  
Data Value:                   91
```

Location

/opt/rsct/bin/enotifyevent

Contains the enotifyevent script

/opt/rsct/bin/notifyevent

Contains the notifyevent script

nroff Command

Purpose

Formats text for printing on typewriter-like devices and line printers.

Syntax

```
nroff [ -e ] [ -h ] [ -i ] [ -q ] [ -z ] [ -o List ] [ -n Number ] [ -s Number ] [ -r ANumber ] [ -u Number ] [ -T Name ] [ -man ] [ -me ] [ -mm ] [ -mptx ] [ -ms ] [ File ... | - ]
```

Description

The **nroff** command reads one or more files for printing on typewriter-like devices and line printers. If no file is specified or the - (minus sign) flag is specified as the last parameter, standard input is read by default. The *File* variable specifies files to be printed on a typewriter-like device by the **nroff** command. The default is standard input.

The **col** command may be required to postprocess **nroff** command output in certain cases.

Flags

Item	Description
-e	Produces equally spaced words in adjusted lines, using the full resolution of a particular terminal.
-h	Uses output tabs during horizontal spacing to speed output and reduce the output character count. Tab settings are assumed to be every eight nominal character widths.
-i	Reads standard input after reading all specified files.
-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.
-n <i>Number</i>	Assigns the specified number to the first printed page.
-o <i>List</i>	Prints only those pages specified by the <i>List</i> variable, which consists of a comma-separated list of page numbers and ranges, as follows: <ul style="list-style-type: none">• 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 the beginning through page 3, pages 6 through 8, page 10, and page 12 to the end. <p>Note: When the -oList flag is used in a pipeline (as with one or more of the eqn or tbl commands) you may receive a broken pipe message if the last page in the document is not specified in the <i>List</i> parameter. This broken pipe message is not an indication of any problem and can be ignored.</p>

Item	Description
-q	Calls the simultaneous input/output mode of the .rd request.
-r ANumber	Sets register A to the specified number. The value specified by the A variable must have a one-character ASCII name.
-s Number	Stops every specified number of pages (the default is 1). The nroff command halts every specified number of pages to allow paper loading or changing, then resumes upon receipt of a linefeed or new-line character. This flag does not work in pipelines (for example, with the mm command). When the nroff command halts between pages, an ASCII BEL character is sent to the workstation.
-T Name	<p>Prepares the output for the specified printing device. Typewriter-like devices and line printers use the following <i>Name</i> variables for international extended character sets, as well as English-language character sets, digits, and symbols:</p> <p>hplj Hewlett-Packard LaserJet II and other models in the same series of printers.</p> <p>ibm3812 3812 Pageprinter II.</p> <p>ibm3816 3816 Pageprinter.</p> <p>ibm4019 4019 LaserPrinter.</p> <p>Note: The 4019 and the HP Laser Jet II printer both have nonprintable areas at the top and bottom of a page. If a file is targeted for these printers, be sure to define top and bottom margins (for example, by formatting with the -mm flag) so that all output can be positioned within the printable page.</p> <p>37 Teletype Model 37 terminal (default) for terminal viewing only. This device does not support extended characters that are inputted by the \[N] form. Inputting Extended Single-Byte Characters provides more information.</p> <p>lp Generic name for printers that can underline and tab. All text sent to the lp value using reverse linefeeds (for example, text that includes tables) must be processed with the col command. This device does not support extended characters that are inputted by the \[N] form. Inputting Extended Single-Byte Characters provides more information.</p> <p>ppds Generic name for printers that support the personal printer data streams such as the Quietwriter III, Quickwriter, and Proprinters.</p> <p>ibm5575 5575 Kanji Printer.</p> <p>ibm5577 5577 Kanji Printer.</p> <p>Note: For completeness of the text formatting system, the following devices are shipped <i>as is</i> from the AT&T Distribution center. No support is provided for these tables.</p>

Item	Description
-T Name (Continued)	
2631	Hewlett-Packard 2631 printer in regular mode.
2631-c	Hewlett-Packard 2631 printer in compressed mode.
2631-e	Hewlett-Packard 2631 printer in expanded mode.
300	DASI-300 printer.
300-12	DASI-300 terminal set to 12 characters per inch.
382	DTC-382.
4000a	Trendata 4000a terminal (4000A).
450	DASI-450 (Diablo Hyterm) printer.
450-12	DASI-450 terminal set to 12 characters per inch.
832	Anderson Jacobson 832 terminal.
8510	C.ITOH printer.
tn300	GE Terminet 300 terminal.
X	Printers equipped with a TX print train.
300s	DASI-300s printer (300S).
300s-12	DASI-300s printer set to 12 characters per inch (300S-12).
-u Number	Sets the bold factor (number of character overstrokes) for the third font position (bold) to the specified number, or to 0 if the <i>Number</i> variable is missing.
-z	Prints only messages generated by .tm (workstation message) requests. Note: See the Macro Packages for Formatting Tools in the troff command for information about the macros.
-	Forces input to be read from standard input.

Files

Item	Description
/usr/share/lib/tmac/tmac.*	Contains pointers to standard macro files.
/usr/share/lib/macros/*	Contains standard macro files.
/usr/share/lib/nterm/*	Contains the terminal driving tables for the nroff command.
/usr/share/lib/pub/terminals	Contains a list of supported terminals.

nslookup Command

Purpose

Queries internet domain name servers interactively.

Syntax

```
nslookup [ - option ] [ name | - ] [ server ]
```

Description

The **nslookup** command queries internet domain name servers in two modes. Interactive mode allows you to query name servers for information about various hosts and domains, or to print a list of the hosts in a domain. In noninteractive mode, the names and requested information are printed for a specified host or domain.

The **nslookup** command enters interactive mode when no arguments are given, or when the first argument is a - (minus sign) and the second argument is the host name or internet address of a name server. When no arguments are given, the command queries the default name server. The **nslookup** command enters non-interactive mode when you give the name or internet address of the host to be looked up as the first argument. The optional second argument specifies the host name or address of a name server. You can specify options on the command line if they precede the arguments and are prefixed with a hyphen. For example, to change the default query type to host information, and the initial timeout to 10 seconds, enter the following command:

```
nslookup -query=hinfo -timeout=10
```

Interactive commands

Item	Description
host [<i>server</i>]	Looks up information for the host using the current default server or using <i>server</i> , if specified. If the host is an Internet address and the query type is A or PTR , the nslookup command returns the name of the host. If the host is a name and does not have a trailing period, the search list is used to qualify the name. To look up a host not in the current domain, append a period to the name.
server <i>Domain</i> lserver <i>Domain</i>	Changes the default server to the value specified by the <i>Domain</i> parameter. The lserver subcommand uses the initial server to look up information about the domain. The server subcommand uses the current default server. If an authoritative answer cannot be found, the names of any additional servers that might have the answer are returned.
exit	Exits the program.

Item	Description
set <i>Keyword</i> [= <i>Value</i>]	Changes state information that affects lookups. You can specify the following keywords:
all	Prints the current values of the frequently used options to set . Information about the current default server and host is also printed.
class= <i>value</i>	Changes the query class to one of the following value. The class specifies the protocol group of the information. The default is IN .
IN	The Internet class.
CH	The Chaos class.
HESIOD	The Hesiod class.
ANY	Wildcard (any of the above).
[no]debug	Turns debugging mode on. The default is nodebug .
[no]d2	Turns comprehensive debugging on. The default is nod2 .
domain= <i>name</i>	Changes the default domain name to the name specified by the <i>name</i> parameter.
[no]search	Appends the domain names in the domain search list to the request until an answer is received, if the lookup request contains a period other than a trailing period. The default is search .
port= <i>value</i>	Changes the default TCP/UDP name server port to the number specified by the <i>value</i> parameter. The default value is 53.
querytype= <i>valuetype=</i> <i>value</i>	Changes the type of the information query to the type specified by the <i>value</i> parameter. The default value is A.
[no]recurse	Tells the name server to query other servers if it does not have the information. The default is recurse .
ndots= <i>number</i>	Sets the number of dots (label separators) in a domain that disables the search function. Absolute names always stop the search function.
retry= <i>number</i>	Sets the number of retries to the number specified by the <i>number</i> parameter.
timeout= <i>number</i>	Changes the initial timeout interval for waiting for a reply to the seconds specified by the <i>number</i> parameter.
[no]vc	Always uses a virtual circuit when sending requests to the server. The default is novc .
[no]fail	Tries the next name server if a name server responds with SERVFAIL or a referral (nofail) or terminate query (fail) on such a response. The default is nofail .

Return values

The **nslookup** command returns with an exit status of 1 if a query fails and returns 0 otherwise.

IDN support

If the **nslookup** command is built with IDN (internationalized domain name) support, it can accept and display non-ASCII domain names. The **nslookup** command appropriately converts character encoding of a domain name before sending a request to a DNS server or displaying a reply from the server. To turn off IDN support, define the IDN_DISABLE environment variable. IDN support is disabled if the variable is set when the **nslookup** command runs, or when the standard output is not a tty.

Files

Item	Description
/etc/resolv.conf	Contains the initial domain name and name server addresses.

nsupdate Command

Purpose

Updates a DNS server dynamically.

Syntax

```
nsupdate [-d] [-D] [-i] [-L level] [ [-g] | [-o] | [-1] | [-y [hmac:]keyname:secret] | [-k keyfile] ] [-t timeout] [-u udptimeout] [-r udpretries] [-v] [-T] [-P] [-V] [ [-4] | [-6] ] [filename]
```

Description

The **nsupdate** command submits Dynamic DNS Update requests, as defined in RFC 2136, to a name server. This allows resource records to be added or removed from a zone without manually editing the zone file. A single update request can contain requests to add or remove more than one resource record.

Zones that are dynamically controlled by using the **nsupdate** command or a DHCP server must not be edited manually. Manual edits can conflict with dynamic updates and cause data loss.

The resource records that are dynamically added or removed by using the **nsupdate** command must be in the same zone. Requests are sent to the zone's primary server, which is identified by the MNAME field of the zone's start of authority (SOA) record.

Transaction signatures (TSIG) can be used to authenticate the Dynamic DNS updates. The transaction signatures use the TSIG resource record type as described in RFC 2845, the SIG(0) record as described in RFC 2535 and RFC 2931, or GSS-TSIG as described in RFC 3645.

TSIG relies on a shared secret that must be known only to the **nsupdate** command and the name server. For instance, suitable key and server statements are added to the `/etc/named.conf` file so that the name server can associate the appropriate secret key and algorithm with the IP address of the client application that is using TSIG authentication. The **ddns-confgen** command can generate suitable configuration fragments. The **nsupdate** command uses the **-y** or **-k** options to specify the TSIG shared secret. These options are mutually exclusive.

The SIG(0) key uses public key cryptography. To use a SIG(0) key, the public key must be stored in a KEY record in a zone that is served by the name server. Generic Security Service Algorithm - Transaction Signature (GSS-TSIG) uses Kerberos credentials. Standard GSS-TSIG mode is enabled by using the **-g** flag.

Flags

-4

Indicates that only IPv4 must be used.

-6

Indicates that only IPv6 must be used.

-d

Sets debug mode that provides tracing information about the update requests and the replies that are received from the name server.

-D

Sets extra debug mode.

-i

Forces interactive mode, even when standard input is not a terminal.

-k *keyfile*

Indicates the file that contains the TSIG authentication key. Key files can be in the following formats: a single file containing a `named.conf`-format key statement, which can be generated automatically by the **ddns-confgen** command; or a pair of files whose names are of the format `K{name}.+157.+[random].key` and `K{name}.+157.+[random].private`, which can be generated by using the **dnssec-**

keygen command. You can also use the **-k** flag to specify a SIG(0) key that is used to authenticate dynamic DNS update requests. In this case, the specified key is not an HMAC-MD5 key.

-l

Sets local-host only mode, which sets the server address to `localhost` (disabling the server so that the server address cannot be overridden). Connections to the local server use a TSIG key that is found in the `/var/run/named/session.key` file, which is automatically generated by the **named** daemon if any local primary zone has set the **update-policy** option to `local`. The location of this key file can be overridden by using the **-k** option.

-L level

Sets the debug logging level. If the value is zero, the logging operation is disabled.

-p port

Sets the port that must be used for connecting to a name server. The default value is 53.

-P

Prints the list of private BIND-specific resource record types whose format is identified by the **nsupdate** command. Also, see the **-T** option.

-r udpretries

Sets the number of UDP retries. The default value is 3. If the value is zero, only one update request is processed.

-t timeout

Sets the maximum time that an update request can take before it is aborted. The default value is 300 seconds. If the value is zero, the timeout operation is disabled.

-T

Prints the list of Internet Assigned Numbers Authority (IANA) standard resource record types whose format is identified by the **nsupdate** command. The **nsupdate** command exits after the lists are printed. You can combine the **-T** option with the **-P** option.

Other types can be entered by entering `TYPEXXXXX` where `XXXXX` is the decimal value of the type with no leading zeros. The `rdata` value, if present, is parsed by using the UNKNOWN rdata format, (`<backslash> <hash> <space> <length> <space> <hexstring>`).

-u udptimeout

Sets the UDP retry interval. The default value is 3 seconds. If the value is zero, the interval is computed from the timeout interval and number of UDP retries.

-v

Specifies that TCP must be used even for small update requests. By default, the **nsupdate** command uses User Datagram Protocol (UDP) to send update requests to the name server unless a UDP request cannot be used, in which case TCP is used. TCP is preferable when a batch of update requests is received.

-V

Prints the version number and exits.

-y [hmac:]keyname:secret

Sets the literal TSIG authentication key. Key name is the name of the key, and `secret` is the base64 encoded shared secret. `hmac` is the name of the key algorithm; valid values are `hmac-md5`, `hmac-sha1`, `hmac-sha224`, `hmac-sha256`, `hmac-sha384`, or `hmac-sha512`. If `hmac` is not specified, the default value is `hmac-md5`, or if MD5 was disabled, the default value is `hmac-sha256`.

Note: Use the **-y** flag sparingly because the shared secret is specified as a command-line argument in clear text. The shared secret might be visible in the output from the `ps1` prompt or in a history file maintained by the user's shell.

Input format

The **nsupdate** command reads input from the specified *filename* or standard input. Each command is specified on exactly one line of input. Some commands are used for administrative purposes; other commands are either update instructions or are used to perform prerequisite checks on the contents of the zone. These checks set conditions that some specific name or a set of resource records (RRset) either

exists or is absent in the zone. These conditions must be met to complete the entire update request. Updates are rejected if the tests for the prerequisite conditions fail.

Every update request consists of zero or more prerequisites and zero or more updates. This allows a suitably authenticated update request to proceed if some specified resource records are either present or missing in the zone. A blank input line (or the **send** command) causes the accumulated commands to be sent as one Dynamic DNS update request to the name server.

The command formats and their meanings follow:

server *servername* *port*

This command sends all dynamic update requests to the name server *servername*. If a server statement is not specified, the **nsupdate** command sends updates to the primary server of the correct zone. The MNAME field of that zone's SOA record identifies the primary server for that zone. *port* is the port number on server name where the dynamic update requests are sent. If a port number is not specified, the default DNS port number of 53 is used.

local *address* *port*

This command sends all dynamic update requests by using the local address. When a local statement is not specified, the **nsupdate** command sends updates by using an address and port chosen by the system. *port* can also be used to force requests from a specific port. If a port number is not specified, the system assigns a port.

zone *zonename*

This command specifies that all updates must be made to the zone *zonename*. If a zone statement is not specified, the **nsupdate** command attempts to determine the correct zone to update based on the input.

class *classname*

This command specifies the default class. If a class is not specified, the default class is IN.

ttl *seconds*

This command specifies the default time-to-live, in seconds, for records that are added. The value none clears the default TTL.

key *hmac:keyname* *secret*

This command specifies that all updates must be TSIG-signed by using the *keyname-secret* pair. If *hmac* is specified, it sets the signing algorithm. The default value is `hmac -md5`; if MD5 was disabled, the default value is `hmac -sha256`. The **key** command overrides any key that is specified on the command line by using the **-y** or **-k** flag.

gsstsig

This command uses GSS-TSIG to sign the updates. This command is equivalent to specifying the **-g** flag on the command line.

oldgsstsig

This command uses the Windows 2000 version of GSS-TSIG to sign the updates. This is equivalent to specifying the **-o** flag on the command line.

realm [*realm_name*]

When you use GSS-TSIG, this command specifies the **realm_name** parameter instead of the default realm in the `krb5.conf` file. If realm is not specified, the saved realm is cleared.

check-names [*yes_or_no*]

This command turns on or turns off check-names processing on records that are added. Check-names does not affect prerequisites or records that you want to delete. By default, check-names processing is turned on. If the **check-names** command fails to run, the record is not added to the UPDATE message.

prereq *nxdomain* *domain-name*

This command asserts a prerequisite that no resource record of any type exist with the name *domain-name*.

prereq *yxdomain* *domain-name*

This command asserts a prerequisite that the *domain-name* must exist (at least one resource record, of any type).

prereq nxrrset *domain-name class type*

This command requires that no resource record exist of the specified type, class, and domain-name. If *class* is omitted, IN (Internet) is assumed.

prereq yxrrset *domain-name class type*

This command requires that a resource record of the specified type, class and domain-name exist. If *class* is omitted, IN (internet) is assumed.

prereq yxrrset *domain-name class type data*

By using this command, the data from each set of prerequisites of this form that is sharing a common type, class, and domain-name are combined to form a set of RRs. This set of RRs must exactly match the set of RRs existing in the zone for the specified type, class, and domain-name. The data is written in the standard text representation of the resource record's RDATA.

update delete *domain-name ttl class type data*

This command deletes any resource records named *domain-name*. If *type* and *data* are specified, only matching resource records are removed. The IN class is assumed if *class* is not specified. The *ttl* is ignored, and it is only allowed for compatibility.

update add *domain-name ttl class type data*

This command adds a resource record with the specified ttl, class, and data.

show

This command displays the current message, containing all of the prerequisites and updates specified since the last send operation.

send

This command sends the current message. This operation is equivalent to entering a blank line.

answer

This command displays the answer.

debug

This command turns on debugging.

version

This command prints the version number.

help

This command prints a list of commands.

Lines beginning with a semicolon (;) are comments and are ignored.

Examples

The following examples show the **nsupdate** command usage to insert and delete resource records from the `example.com` zone. The input in each example contains a trailing blank line so that a group of commands can be specified as one dynamic update request to the primary name server, `example.com`.

1. To delete resource records from an existing zone and to add resource record to a new zone, enter the following command:

```
# nsupdate
> update delete oldhost.example.com A
> update add newhost.example.com 86400 A 172.16.1.1
> send
```

Any A records for `oldhost.example.com` are deleted, and an A record for `newhost.example.com` with IP address `172.16.1.1` is added. The new record has a TTL of 1 day (86400 seconds).

2. To specify a prerequisite condition before updating the DNS server, enter the following command:

```
# nsupdate
> prereq nxdomain nickname.example.com
> update add nickname.example.com 86400 CNAME somehost.example.com
> send
```

The prerequisite condition specifies the name server to verify that no resource records of any type exist for `nickname.example.com`. If the zone has resource records, the update request fails. If this

name does not exist, a CNAME is added, which ensures that the prerequisite condition cannot conflict with the long-standing rule that is specified in RFC 1034, which states that a name must not exist as any other record type if it exists as a CNAME. (The rule has been updated for DNSSEC in RFC 2535 to allow CNAMEs to have RRSIG, DNSKEY, and NSEC resource records.)

Files

/etc/resolv.conf

Used to identify the default name server.

/var/run/named/session.key

Sets the default TSIG key that must be used in local-only mode.

K{name}.+157.+{random}.key

Base-64 encoding of the HMAC-MD5 key created by the **dnssec-keygen** command.

K{name}.+157.+{random}.private

Base-64 encoding of the HMAC-MD5 key created by the **dnssec-keygen** command.

ntpd4 Daemon

Purpose

Network Time Protocol (NTP) Daemon.

Syntax

```
ntpd4 [ -4 ] [ -6 ] [ -a ] [ -A ] [ -b ] [ -c conffile ] [ -d ] [ -D level ] [ -f driftfile ] [ -g ] [ -i jailedir ] [ -k keyfile ] [ -l logfile ] [ -L ] [ -n ] [ -N ] [ -p pidfile ] [ -P priority ] [ -q ] [ -r broadcastdelay ] [ -s statsdir ] [ -t key ] [ -u user[:group] ] [ -U interface update interval ] [ -v variable ] [ -V variable ] [ -x ]
```

Description

The **ntpd** program is an operating system daemon that sets and maintains the system time-of-day in synchrony with the Internet Standard Time servers. The **ntpd** program is a complete implementation of the Network Time Protocol (NTP) version 4, and also retains compatibility with version 3, as defined by the RFC-1305, and version 1 and 2, as defined by RFC-1059 and RFC-1119. The **ntpd** program generally computes in 64-bit floating point arithmetic mode. If a precision of 232 picoseconds needs to be maintained, then **ntpd** computes in 64-bit fixed-point mode. The ultimate precision of 232 picoseconds is not achievable with existing workstations and networks, however, this precision might be required with future Gigahertz CPU clocks and Gigabit LANs.

Frequency discipline

The **ntpd** behavior at startup depends on the frequency file, usually **ntp.drift**. This file contains the latest estimate of clock frequency error. When the **ntpd** daemon is started and the file does not exist, the **ntpd** enters a special mode that is designed to quickly adapt to the particular system clock oscillator time and frequency error. This takes approximately 15 minutes after which the time and frequency are set to nominal values and the **ntpd** enters normal mode of operation, where the time and frequency are continuously tracked relative to the server. After one hour, the frequency file is created and the current frequency offset is written to this file. When the **ntpd** is started and the file does exist, the **ntpd** frequency is initialized from the file and **ntpd** enters the normal mode of operation. After that the current frequency offset is written to the file at hourly intervals.

Operating modes

The **ntpd** program can operate in any of the several modes, including symmetric active/passive, client/server, and broadcast/multicast. The **ntpd** normally operates continuously while monitoring for small changes in frequency and trimming the clock for the ultimate precision. The **ntpd** can operate in a

one-time mode where the time is set from an external server and frequency is set from a previously recorded frequency file. A broadcast or multicast client can discover remote servers, compute server-client propagation delay correction factors and configure itself automatically. This makes it possible to deploy a fleet of workstations without specifying configuration details specific to the local environment.

By default, **ntpd** runs in continuous mode where each of the possibly several external servers is polled at intervals determined by an intricate state machine. The state machine measures the incidental roundtrip delay jitter and the oscillator frequency wander and determines the best poll interval using a heuristic algorithm. Ordinarily, and in most operating environments, the state machine starts with 64-seconds intervals and eventually increases in steps to 1024 seconds. A small amount of random variation is introduced in order to avoid bunching at the servers. In addition, should a server become unreachable for some time, the poll interval is increased in steps to 1024 seconds in order to reduce network overhead.

In some cases, it might not be practical for **ntpd** to run continuously. A common workaround has been to run the **ntpdate** program from a **cron** job at designated times. However, this program does not have the crafted signal processing, error checking and mitigation algorithms of **ntpd**. The **-q** option is intended for this purpose. Setting this option will cause **ntpd** to exit just after setting the clock for the first time. The procedure for initially setting the clock is the same as in continuous mode; most applications specify the **iburst** command with the server configuration command. With this command a volley of messages are exchanged to groom the data and the clock is set in to about 10 second. If no response is received, after a couple of minutes, the daemon times out and exits. After a certain period if no response is received, the **ntpdate** program is stopped.

Flags

Item	Description
-4	Forces DNS resolution of hostnames to the IP version 4 namespace.
-6	Force DNS resolution of hostnames to the IP version 6 namespace.
-a	Requires cryptographic authentication for broadcast client, multicast client and symmetric passive associations. This is the default value.
-A	Does not require cryptographic authentication for broadcast client, multicast client, and symmetric passive associations.
-b	Enables the client to synchronize to broadcast servers.
-c <i>conf</i> file	Specifies the name and path of the configuration file, default <code>/etc/ntp.conf</code> .
-d	Specifies debugging mode. This option may occur more than once, with each occurrence indicating greater detail of display.
-D <i>level</i>	Specifies the debugging level directly.
-f <i>drift</i> file	Specifies the name and path of the frequency file, default <code>/etc/ntp.drift</code> . This is the same operation as the driftfile driftfile configuration command.
-g	Allows the time to be set to any value without restriction; this can happen only once. The ntpd command exits with a message to the system log if the offset exceeds the panic threshold, which is 1000 seconds by default. If the threshold is exceeded after that, ntpd will exit with a message to the system log. This option can be used with the -q and -x options.
-i <i>jail</i> dir	The chroot command directs the server to the directory jaildir . This option also implies that the server attempts to drop root privileges at startup (otherwise, chroot gives little additional security), and it is only available if the operating system supports to run the server without full root privileges. You must specify a -u option.
-k <i>key</i> file	Specifies the name and path of the symmetric key file, default <code>/etc/ntp.keys</code> . This is the same operation as the keys keyfile configuration command.
-l <i>log</i> file	Specifies the name and path of the log file. The default is the system log file. This is the same operation as the logfile configuration command.

Item	Description
-L	Does not listen to virtual IPs. The default is to listen.
-n	Does not fork.
-N	Runs the ntpd at the highest priority level to the extent permitted by the operating system.
-p <i>pidfile</i>	Specifies the name and path of the file used to record the ntpd process ID. This is the same operation as the pidfile pidfile configuration command.
-P <i>priority</i>	Runs the ntpd at the specified priority to the extent permitted by the operating system.
-q	Exits the ntpd just after the first time the clock is set. This behavior mimics that of the ntpdate program, which is to be retired. The -g and -x options can be used with this option. Note: The kernel time discipline is disabled with this option.
-r <i>broadcastdelay</i>	Specifies the default propagation delay from the broadcast/multicast server to the client. This is necessary only if the delay cannot be computed automatically by the protocol.
-s <i>statsdir</i>	Specifies the directory path for files created by the statistics facility. This is the same operation as the statsdir configuration command.
-t <i>key</i>	Adds a key number to the trusted key list. This option can occur more than once.
-u <i>user[:group]</i>	Specifies an user, and optionally a group, to switch. This option is only available if the operating system supports running the server without complete root privileges.
-U <i>interface update interval</i>	Specifies the number of seconds to wait between the interface list scans to pick up new and deleted network interface. Set to 0 to disable dynamic interface list updating. The default is to scan every 5 minutes.
-v <i>variable</i>	Adds a system variable listed by default.
-V <i>variable</i>	
-x	Slews the time if the offset is less than the step threshold, which is 128 milliseconds by default, and steps up if above the threshold. This option sets the threshold to 600 seconds, which is well within the accuracy window to set the clock manually.

Exit Status

This command returns the following exit values:

- 0**
Successful completion.
- > 0**
An error occurred.

Security

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

Auditing Events : N/A

Examples

By default, the symbolic link `/usr/sbin/xntpd` points to NTP version 4 daemon (`/usr/sbin/ntp4/ntpd4`).

```
/usr/sbin/xntpd -> /usr/sbin/ntp4/ntpd4
```

1. To start the **xntpd** daemon, enter:

```
startsrc -s xntpd
```

2. To stop the **xntpd** daemon, enter:

```
stopsrc -s xntpd
```

Files

/usr/sbin/ntp4/ntpd4

Contains the **ntpd4** daemon.

/usr/sbin/xntpd-->/usr/sbin/ntp4/ntpd4

Default symbolic link to NTP version 4 binary file from the `/usr/sbin` directory.

/etc/ntp.conf

Contains the default configuration file.

/etc/ntp.drift

Contains the default drift file.

ntpdate4 Command

Purpose

Sets the date and time using the Network Time Protocol (NTP).

Syntax

```
ntpdate4 [ -4 ] [ -6 ] [ -a key ] [ -B ] [ -b ] [ -d ] [ -e authdelay ] [ -k keyfile ] [ -o version ] [ -p samples ] [ -q ] [ -s ] [ -t timeout ] [ -u ] [ -v ] server [...]
```

Description

The **ntpdate** command sets the local date and time by polling the Network Time Protocol (NTP) server(s) given as the server arguments to determine the correct time. The **ntpdate** must be run as root on the local host. Samples are obtained from each of the servers specified and a subset of the NTP clock filter and selection algorithms are applied to select the best. Note that the accuracy and reliability of **ntpdate** depends on the number of servers, the number of polls each time it is run and the interval between runs.

The **ntpdate** can be run manually as necessary to set the host clock, or it can be run from the host startup script to set the clock at boot time. This is useful in some cases to set the clock initially before starting the NTP daemon **ntpd**. It is also possible to run **ntpdate** from a **cron** script. However, it is important to note that **ntpdate** with contrived **cron** scripts is not a substitute for the NTP daemon, which uses complex algorithms to maximize accuracy and reliability while minimizing resource use. Finally, since **ntpdate** does not tune the host clock frequency as does **ntpd**, the accuracy using **ntpdate** is limited.

Time adjustments are made by **ntpdate** in one of two ways. If **ntpdate** determines that the clock is in error of more than 0.5 seconds it will simply step the time by calling the system **settimeofday()** routine. If the error is less than 0.5 seconds, it will slew the time by calling the system **adjtime()** routine. The latter technique is less disruptive and more accurate when the error is small, and works quite well when **ntpdate** is run by **cron** every hour or two.

The **ntpdate** will decline to set the date if an NTP server daemon (**ntpd**) is running on the same host. When running **ntpdate** on a regular basis from **cron** as an alternative to running a daemon, doing so once every hour or two will result in precise enough timekeeping to avoid stepping the clock.

Note: Where a host name is expected, a **-4** qualifier preceding the host name forces DNS resolution to the IP version 4 namespace, while a **-6** qualifier forces DNS resolution to the IP version 6 namespace.

Flags

Item	Description
- 4	Forces DNS resolution of following host names on the command line to the IP v4 namespace
- 6	Forces DNS resolution of following host names on the command line to the IP v6 namespace.
- a <i>key</i>	Enables the authentication function and specifies the key identifier to be used for authentication as the argument keyntpd . The keys and key identifiers must match in both the client and server key files. The default is to disable the authentication function.
- B	Forces the time to be slewed using the adjtime () system call, even if the measured offset is greater than + or - 128 millisecond. The default is to step the time using settimeofday () if the offset is greater than + or -128 millisecond. Note that, if the offset is much greater than + or -128 millisecond in this case, that it can take a long time (hours) to slew the clock to the correct value. During this time the host should not be used to synchronize clients.
- b	Forces the time to be stepped using the settimeofday () system call, rather than slewed (default) using the adjtime () system call. This option should be used when called from a startup file at boot time.
- d	Enables the debugging mode, in which ntpd will go through all the steps, but not adjust the local clock. Information useful for general debugging is also printed.
- e <i>authdelay</i>	Specifies the processing delays to perform an authentication function as the value <i>authdelay</i> , in seconds and fraction (See the ntpd for more details). This number is usually small enough to be negligible for most purposes, though specifying a value may improve timekeeping on very slow CPUs.
- k <i>keyfile</i>	Specifies the path for the authentication key file as the string <i>keyfile</i> . The default is /etc/ntp.keys.
- o <i>version</i>	Specifies the NTP version for outgoing packets as the integer <i>version</i> , which can be 1 or 2. The default is 3. This allows ntpd to be used with older NTP versions.
- p <i>samples</i>	Specifies the number of samples to be acquired from each server as integer <i>samples</i> , with values from 1 to 8 inclusive. The default value is 4.
- q	Specifies the query. Does not set the clock.
- s	Diverts logging output from the standard output (default) to the system syslog facility. This is designed primarily for convenience of cron scripts.
- t <i>timeout</i>	Specifies the maximum time waiting for a server response as the value <i>timeout</i> , in seconds and fraction. The value is rounded to a multiple of 0.2 seconds. The default is 1 second, a value suitable for polling across a LAN.
- u	Directs ntpd to use an unprivileged port or outgoing packets. You can use this option when behind a firewall that blocks incoming traffic to privileged ports, and you want to synchronize with hosts beyond the firewall. Note that the -d option always uses unprivileged ports.
- v	Verbose output. This option causes the ntpd version identification string to be logged.

Parameters

Item	Description
<i>Server...</i>	Specifies the servers to poll

Exit Status

This command returns the following exit values:

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

Security

Item	Description
Access Control	You must have root privilege to run this command.
Auditing Events	N/A

Examples

1. To set the local date and time by polling the NTP servers at address 9.41.254.24, enter:

```
ntpdate 9.41.254.24
```

Output similar to the following appears:

```
address: ::
address: 0.0.0.0
25 Feb 12:19:41 ntpdate[434262]: adjust time server 9.41.254.24 offset -0.005270 sec
```

Files

/usr/sbin/ntp4/ntpdate4

Contains the **ntpdate4** command.

/usr/sbin/ntpdate ->/usr/sbin/ntp4/ntpdate4

Default symbolic link to NTP version 4 binary file from the **/usr/sbin** directory.

/etc/ntp.keys

Encryption keys used by the **ntpdate4** command.

ntpdc4 Command

Purpose

Starts the query or control program for the Network Time Protocol (NTP) daemon, **ntpd**.

Syntax

```
ntpdc [ -4 ] [ -6 ] [ -d ] [ -i ] [ -l ] [ -n ] [ -p ] [ -s ] [ -c command ] [ host ] [ ... ]
```

Description

The **ntpdc** command is used to query the **ntpd** daemon about its current state and to request changes in the state. The program may be run either in interactive mode or controlled using command line arguments. Extensive state and statistics information is available through the **ntpdc** interface. In addition, all the configuration options which can be specified at startup using **ntpd**'s configuration file might also be specified at run time using **ntpdc**.

If one or more request options are included in the command line when **ntpd** is executed, each of the requests will be sent to the NTP servers running on each of the hosts given as command line arguments, or on localhost by default. If no request options are given, **ntpd** will attempt to read commands from the standard input and execute these on the NTP server running on the first host given on the command line, again defaulting to localhost when no other host is specified. **ntpd** will prompt for commands if the standard input is a terminal device.

ntpd uses NTP mode 7 packets to communicate with the NTP server, and hence can be used to query any compatible server on the network which permits it. Note that since NTP is a UDP protocol this communication will be somewhat unreliable, especially over large distances in terms of network topology. **ntpd** makes no attempt to retransmit requests, and will time requests out if the remote host is not heard from within a suitable timeout time.

The operation of **ntpd** are specific to the particular implementation of the ntpd daemon and can be expected to work only with this and maybe some previous versions of the daemon. Requests from a remote **ntpd** program that affects the state of the local server must be authenticated, which requires both the remote program and local server share a common key and key identifier.

Note that in contexts where a host name is expected, a -4 qualifier preceding the host name forces DNS resolution to the IP version 4 namespace, while a -6 qualifier forces DNS resolution to the IP version 6 namespace.

Specifying a command line option other than **-i** or **-n** will cause the specified query (queries) to be sent to the indicated host(s) immediately. Else, **ntpd** will attempt to read interactive format commands from the standard input.

Flags

Item	Description
-4	Forces DNS resolution of following host names on the command line to the IP version 4 namespace.
-6	Forces DNS resolution of following host names on the command line to the IP version 6 namespace.
-c <i>command</i>	The following argument is interpreted as an interactive format command and is added to the list of commands to be executed on the specified host(s). You can run multiple -c options.
-d	Enables debugging mode. You can use this option more than once.
-i	Forces ntpd to operate in interactive mode. Prompts will be written to the standard output and commands read from the standard input.
-l	Obtains a list of peers, which are known to the server(s). This switch is equivalent to -c listpeers .
-n	Outputs all host addresses in dotted-quad numeric format rather than converting them to the canonical host names.
-p	Prints a list of the peers known to the server as well as a summary of their state. This is equivalent to -c peers .
-s	Prints a list of the peers known to the server as well as a summary of their state. The print format is different from the -p switch. This is equivalent to -c dmpeers .

Parameters

Item	Description
<i>Host...</i>	Specifies the hosts.

Exit Status

This command returns the following exit values:

Item	Description
------	-------------

- | | |
|----|------------------------|
| 0 | Successful completion. |
| >0 | An error occurred. |

Security

Access Control: You must be part of the system group to run this command.

Auditing Events: N/A

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 the query/control program for the Network Time Protocol daemon, enter:

```
ntpdc
```

2. To print a list of the peers known to the server as well as a summary of their state, enter:

```
ntpdc -p
```

Output similar to the following appears:

```
remote          local          st  poll reach  delay  offset  disp
=====
ausgsa.austin.ibm.com 9.124.101.190  2   64   1    0.29128 -0.013381 2.81735
```

ntpdc Internal Commands

Interactive Commands

Interactive format commands consist of a keyword followed by zero to four arguments. Only enough characters of the full keyword to uniquely identify the command need be typed. The output of a command is normally sent to the standard output, but optionally the output of individual commands may be sent to a file by appending a <, followed by a file name, to the command line.

A number of interactive format commands are executed entirely within the ntpdc program itself and do not result in NTP mode 7 requests being sent to a server. The following list describes the interactive commands.

Item	Description
? [command_keyword] or help [command_keyword]	A question mark (?) by itself prints a list of all the command keywords known to this incarnation of ntpdc . A question mark (?) followed by a command keyword will print function and the use of the command.
delay milliseconds	Specifies a time interval to be added to timestamps included in requests which require authentication. This is used to enable (unreliable) server reconfiguration over long delay network paths or between machines whose clocks are unsynchronized.

Item	Description
host hostname	Sets the host to which future queries will be sent. Hostname may be either a host name or a numeric address.
hostnames [yes no]	If yes is specified, host names are printed in information displays. If no is specified, numeric addresses are printed instead. The default is yes, unless modified using the command line -n switch.
keyid keyid	Allows the specification of a key number to be used to authenticate configuration requests from ntpd to the host(s). This must correspond to a key number which the host/server has been configured to use for this purpose (server options: trustedkey , and requestkey). If authentication is not enabled on the host(s) for ntpd commands, the command keyid 0 should be given, else the keyid of the next subsequent addpeer/addserver/broadcast command will be used.
quit	Exits ntpd .
passwd	Prompts you to type in a password (which will not be echoed) which will be used to authenticate configuration requests. The password must correspond to the key configured for use by the NTP server for this purpose if such requests are to be successful.
timeout milliseconds	Specifies a timeout period for responses to server queries. The default is about 8000 milliseconds. Note: ntpd retries each query once after a timeout, and hence the total waiting time for a timeout will be twice the timeout value set.

Control Message Commands

Query commands result in NTP mode 7 packets containing requests for information being sent to the server. These are read-only commands do not make any modification to the server configuration state.

Item	Description
listpeers	Obtains and prints a brief list of the peers for which the server is maintaining state. These should include all configured peer associations as well as those peers whose stratum is such that they are considered by the server to be possible future synchronization candidates.
peers	Obtains a list of peers for which the server is maintaining state, along with a summary of that state. Summary information includes the address of the remote peer, the local interface address (0.0.0.0 if a local address has yet to be determined), the stratum of the remote peer (a stratum of 16 indicates the remote peer is unsynchronized), the polling interval, in seconds, the reachability register, in octal, and the current estimated delay, offset and dispersion of the peer, all in seconds. The character in the left margin indicates the mode this peer entry is operating in. A + denotes symmetric active, a - indicates symmetric passive, a = means the remote server is being polled in client mode, a ^ indicates that the server is broadcasting to this address, a ~ denotes that the remote peer is sending broadcasts and a * marks the peer the server is currently synchronizing to. The contents of the host field may be one of four forms. It may be a host name, an IP address, a reference clock implementation name with its parameter or REFCLK(implementation number, parameter). On hostnames no only IP-addresses will be displayed.

Item	Description
dmpeers	A slightly different peer summary list. Identical to the output of the peers command, except for the character in the leftmost column. Characters only appear beside peers which were included in the final stage of the clock selection algorithm. A . indicates that this peer was cast off in the falseticker detection, while a + indicates that the peer made it through. A * denotes the peer the server is currently synchronizing with.
showpeer peer_address [...]	Shows a detailed display of the current peer variables for one or more peers. Most of these values are described in the NTP Version 2 specification.
pstats peer_address [...]	Displays per-peer statistic counters associated with the specified peer(s).
clockinfo clock_peer_address [...]	Obtains and print information concerning a peer clock. The values obtained provide information on the setting of fudge factors and other clock performance information.
kerninfo	Obtains and print kernel phase-lock loop operating parameters. This information is available only if the kernel has been specially modified for a precision timekeeping function.
loopinfo [oneline multiline]	Prints the values of selected loop filter variables. The loop filter is the part of NTP which deals with adjusting the local system clock. The offset is the last offset given to the loop filter by the packet processing code. The frequency is the frequency error of the local clock in parts-per-million (ppm). The time_const controls the stiffness of the phase-lock loop and thus the speed at which it can adapt to oscillator drift. The watchdog timer value is the number of seconds which have elapsed since the last sample offset was given to the loop filter. The oneline and multiline options specify the format in which this information is to be printed, with multiline as the default.
sysinfo	<p>Print a variety of system state variables, i.e., state related to the local server. All except the last four lines are described in the NTP Version 3 specification, RFC-1305.</p> <p>The system flags show various system flags, some of which can be set and cleared by the enable and disable configuration commands, respectively. These are the auth, bclient, monitor, pll, pps and stats flags. See the ntpd documentation for the meaning of these flags. There are two additional flags which are read only, the kernel_pll and kernel_pps. These flags indicate the synchronization status when the precision time kernel modifications are in use. The kernel_pll indicates that the local clock is being disciplined by the kernel, while the kernel_pps indicates the kernel discipline is provided by the PPS signal.</p> <p>The stability is the residual frequency error remaining after the system frequency correction is applied and is intended for maintenance and debugging. In most architectures, this value will initially decrease from as high as 500 ppm to a nominal value in the range .01 to 0.1 ppm. If it remains high for some time after starting the daemon, something may be wrong with the local clock, or the value of the kernel variable tick may be incorrect.</p> <p>The broadcastdelay shows the default broadcast delay, as set by the broadcastdelay configuration command.</p> <p>The authdelay shows the default authentication delay, as set by the authdelay configuration command.</p>
sysstats	Prints statistics counters maintained in the protocol module.
memstats	Prints statistics counters related to memory allocation code.

Item	Description
iostats	Prints statistics counters maintained in the input-output module.
timerstats	Prints statistics counters maintained in the timer/event queue support code.
reslist	Obtains and print the server's restriction list. This list is printed in sorted order and may help to understand how the restrictions are applied.
ifstats	Lists interface statistics for interfaces used by ntpd for network communication.
ifreload	Forces the scan of current system interfaces. Outputs interface statistics for interfaces that could possibly change. Marks unchanged interfaces with ., added interfaces with + and deleted interfaces with -.
monlist [version]	Obtains and prints traffic counts collected and maintained by the monitor facility. The version number should not normally need to be specified.
clkbug clock_peer_address [...]	Obtains debugging information for a reference clock driver. This information is provided only by some clock drivers and cannot be decoded without a copy of driver source.

Runtime Configuration Requests

All requests which cause state changes in the server are authenticated by the server using a configured NTP key (the facility can also be disabled by the server by not configuring a key). The key number and the corresponding key must also be made known to **ntpd**. This can be done using the **keyid** and **passwd** commands, the latter of which will prompt at the terminal for a password to use as the encryption key. You will also be prompted automatically for both the key number and password the first time a command which would result in an authenticated request to the server is given. Authentication not only provides verification that the requester has permission to make such changes, but also gives an extra degree of protection against transmission errors.

Authenticated requests always include a timestamp in the packet data, which is included in the computation of the authentication code. This timestamp is compared by the server to its receive time stamp. If they differ by more than a small amount the request is rejected. This is done for two reasons. First, it makes simple replay attacks on the server, by someone who might be able to overhear traffic on your LAN, much more difficult. Second, it makes it more difficult to request configuration changes to your server from topologically remote hosts. While the reconfiguration facility will work well with a server on the local host, and may work adequately between time-synchronized hosts on the same LAN, it will work very poorly for more distant hosts. As such, if reasonable passwords are chosen, care is taken in the distribution and protection of keys and appropriate source address restrictions are applied, the run time reconfiguration facility should provide an adequate level of security.

The following commands run authenticated requests.

Item	Description
addpeer peer_address [keyid] [version] [minpoll# prefer iburst burst minpoll N maxpoll N ...]	Add a configured peer association at the given address and operating in symmetric active mode. Note that an existing association with the same peer may be deleted when this command is executed, or may simply be converted to conform to the new configuration, as appropriate. If the keyid is nonzero, all outgoing packets to the remote server will have an authentication field attached encrypted with this key. If the value is 0 (or not given) no authentication will be done. If ntpd 's key number has not yet been set (e.g., by the keyid command), it will be set to this value. The version# can be 1 through 4 and defaults to 3. The remaining options are either a numeric value for minpoll or literals prefer , iburst , burst , minpoll N , keyid N , version N , or maxpoll N (where N is a numeric value), and have the action as specified in the peer configuration file command of ntpd . Each flag (or its absence) replaces the previous setting. The prefer keyword indicates a preferred peer (and thus will be used primarily for clock synchronization if possible). The preferred peer also determines the validity of the PPS signal - if the preferred peer is suitable for synchronization so is the PPS signal.
addpeer peer_address [prefer iburst burst minpoll N maxpoll N keyid N version N ...]	
addserver peer_address [keyid] [version] [minpoll# prefer iburst burst minpoll N maxpoll N ...]	Identical to the addpeer command, except that the operating mode is client.
addserver peer_address [prefer iburst burst minpoll N maxpoll N keyid N version N ...]	
broadcast peer_address [keyid] [version] [prefer]	Identical to the addpeer command, except that the operating mode is broadcast. In this case a valid non-zero key identifier and key are required. The peer_address parameter can be the broadcast address of the local network or a multicast group address assigned to NTP. If a multicast address, a multicast-capable kernel is required.
unconfig peer_address [...]	This command causes the configured bit to be removed from the specified peer(s). In many cases this will cause the peer association to be deleted. When appropriate, however, the association may persist in an unconfigured mode if the remote peer is willing to continue on in this fashion.
fudge peer_address [time1] [time2] [stratum] [refid]	This command provides a way to set certain data for a reference clock. See the source listing for further information.
enable [auth bclient calibrate kernel monitor ntp pps stats]	These commands operate in the same way as the enable and disable configuration file commands of ntpd .
disable [auth bclient calibrate kernel monitor ntp pps stats]	
restrict address mask flag [flag]	This command operates in the same way as the restrict configuration file commands of ntpd .
unrestrict address mask flag [flag]	Removes the restriction of the matching entry from the restrict list.
delrestrict address mask [ntpport]	Deletes the matching entry from the restrict list.

Item	Description
readkeys	Causes the current set of authentication keys to be purged and a new set to be obtained by reading the keys file again (which must have been specified in the ntpd configuration file). This allows encryption keys to be changed without restarting the server.
trustedkey keyid [...] untrustedkey keyid [...]	These commands operate in the same way as the trustedkey and untrustedkey configuration file commands of ntpd .
authinfo	Returns information concerning the authentication module, including known keys and counts of encryptions and decryptions which have been done.
traps	Displays the traps set in the server. See the source listing for further information.
addtrap [address [port] [interface]	Sets a trap for asynchronous messages. See the source listing for further information.
clrtrap [address [port] [interface]	Clears a trap for asynchronous messages. See the source listing for further information.
reset	Clears the statistics counters in various modules of the server. See the source listing for further information.

Files

/usr/sbin/ntp4/ntpd4

Contains the **ntpd4** command.

/usr/sbin/ntpd -> /usr/sbin/ntp4/ntpd4

Default symbolic link to NTP version 4 binary file from the **/usr/sbin** directory.

ntp-keygen4 Command

Purpose

Generate public and private keys.

Syntax

```
ntp-keygen [ -d ] [ -e ] [ -G ] [ -g ] [ -H ] [ -I ] [ -M ] [ -P ] [ -T ] [ -c [RSA-MD2 | RSA-MD5 | RSA-SHA
| RSA-SHA1 | RSA-MDC2 | RSA-RIPEMD160 | DSA-SHA | DSA-SHA1 ] ] [ -i name ] [ -m modulus ] [ -p
password ] [ -q password ] [ -S [ RSA | DSA ] ] [ -s name ] [ -v nkeys ] [ -V params ]
```

Description

The **ntp-keygen4** command generates cryptographic data files used by the NTP version 4 authentication and identification schemes. It generates MD5 key files used in symmetric key cryptography. In addition, if the OpenSSL software library has been installed, it generates keys, certificate and identity files used in public key cryptography. These files are used for cookie encryption, digital signature and challenge/response identification algorithms compatible with the Internet standard security infrastructure.

By default, files are not encrypted by **ntp-keygen**. The **-p password** option specifies the write password and **-q password** option the read password for previously encrypted files. The **ntp-keygen** program prompts for the password if it reads an encrypted file and the password is missing or incorrect. If an encrypted file is read successfully and no write password is specified, the read password is used as the write password by default.

The **ntpd** configuration command `crypto pw` password specifies the read password for previously encrypted files. The daemon expires on the spot if the password is missing or incorrect. For convenience, if a file has been previously encrypted, the default read password is the name of the host running the program. If the previous write password is specified as the host name, these files can be read by that host with no explicit password.

All files are in PEM-encoded printable ASCII format, so they can be embedded as MIME attachments in mail to other sites and certificate authorities. File names begin with the prefix `ntpkey_` and end with the postfix `_hostname.filestamp`, where `hostname` is usually the string returned by the UNIX `gethostname()` routine, and `filestamp` is the NTP seconds when the file was generated, in decimal digits. This both guarantees uniqueness and simplifies maintenance procedures, since all files can be quickly removed by a `rm ntpkey*` command or all files generated at a specific time can be removed by a `rm *filestamp` command. To further reduce the risk of misconfiguration, the first two lines of a file contain the file name and generation date and time as comments.

All files are installed by default in the keys `/usr/local/etc` directory, which is normally in a shared filesystem in NFS-mounted networks. The actual location of the keys directory and each file can be overridden by configuration commands, but this is not recommended. Normally, the files for each host are generated by that host and used only by that host, although exceptions exist as noted later on this page.

Normally, files containing private values, including the host key, sign key and identification parameters, are permitted root read/write-only; while others containing public values are permitted world readable. Alternatively, files containing private values can be encrypted and these files permitted world readable, which simplifies maintenance in shared file systems. Since uniqueness is insured by the hostname and file name extensions, the files for a NFS server and dependent clients can all be installed in the same shared directory.

The recommended practice is to keep the file name extensions when installing a file and to install a soft link from the generic names specified elsewhere on this page to the generated files. This allows new file generations to be activated simply by changing the link. If a link is present, **ntpd** follows it to the file name to extract the filestamp. If a link is not present, **ntpd** extracts the filestamp from the file itself. This allows clients to verify that the file and generation times are always current. The **ntp-keygen** program uses the same extension for all files generated at one time, so each generation is distinct and can be readily recognized in monitoring data.

Running the program

The safest way to run the `ntp-keygen` program is logged in directly as root. The recommended procedure is change to the keys directory, usually `/usr/local/etc`, then run the program. When run for the first time, or if all **ntpkey** files have been removed, the program generates a RSA host key file and matching RSA-MD5 certificate file, which is all that is necessary in many cases. The program also generates soft links from the generic names to the respective files. If run again, the program uses the same host key file, but generates a new certificate file and link.

The host key is used to encrypt the cookie when required and so must be RSA type. By default, the host key is also the sign key used to encrypt signatures. When necessary, a different sign key can be specified and this can be either RSA or DSA type. By default, the message digest type is MD5, but any combination of sign key type and message digest type supported by the OpenSSL library can be specified, including those using the MD2, MD5, SHA, SHA1, MDC2 and RIPE160 message digest algorithms. However, the scheme specified in the certificate must be compatible with the sign key. Certificates using any digest algorithm are compatible with RSA sign keys; however, only SHA and SHA1 certificates are compatible with DSA sign keys.

Private/public key files and certificates are compatible with other OpenSSL applications and very likely other libraries as well. Certificates or certificate requests derived from them should be compatible with extant industry practice, although some users might find the interpretation of X509v3 extension fields somewhat liberal. However, the identification parameter files, although encoded as the other files, are probably not compatible with anything other than Autokey.

Running the program as other than root and using the UNIX `su` command to assume root may not work properly, since by default the OpenSSL library looks for the random seed file `.rnd` in the user

home directory. However, there should be only one .rnd, most conveniently in the root directory, so it is convenient to define the \$RANDFILE environment variable used by the OpenSSL library as the path to /.rnd.

Installing the keys as root might not work in NFS-mounted shared file systems, as NFS clients may not be able to write to the shared keys directory, even as root. In this case, NFS clients can specify the files in another directory such as /etc using the keydir command. There is no need for one client to read the keys and certificates of other clients or servers, as these data are obtained automatically by the Autokey protocol.

Ordinarily, cryptographic files are generated by the host that uses them, but it is possible for a trusted agent (TA) to generate these files for other hosts; however, in such cases files should always be encrypted. The subject name and trusted name default to the hostname of the host generating the files, but can be changed by command line options. It is convenient to designate the owner name and trusted name as the subject and issuer fields, respectively, of the certificate. The owner name is also used for the host and sign key files, while the trusted name is used for the identity files.

Flags

Item	Description
-c [RSA-MD2 RSA-MD5 RSA-SHA RSA-SHA1 RSA-MDC2 RSA-RIPEMD160 DSA-SHA DSA-SHA1]	Selects certificate message digest/signature encryption scheme. Note that RSA schemes must be used with a RSA sign key and DSA schemes must be used with a DSA sign key. The default without this option is RSA-MD5.
-d	Enables debugging. This option displays the cryptographic data produced in eye-friendly billboards.
-e	Writes the IFF client keys to the standard output. This is intended for automatic key distribution by mail.
-G	Generates parameters and keys for the GQ identification scheme, obsoleting any that may exist.
-g	Generates keys for the GQ identification scheme using the existing GQ parameters. If the GQ parameters do not yet exist, create them first.
-H	Generates new host keys, obsoleting any that may exist.
-I	Generates parameters for the IFF identification scheme, obsoleting any that may exist.
-i name	Sets the subject name to name. This is used as the subject field in certificates and in the file name for host and sign keys.
-M	Generates MD5 keys, obsoleting any that may exist.
-m modulus	Sets prime modulus size in bits (256 - 2048). Default size is 512.
-P	Generates a private certificate. By default, the program generates public certificates.
-p password	Encrypts generated files containing private data with password and the DES-CBC algorithm.
-q password	Sets the password for reading files to password.
-S [RSA DSA]	Generates a new sign key of the designated type, obsoleting any that may exist. By default, the program uses the host key as the sign key.
-s name	Sets the issuer name to name. This is used for the issuer field in certificates and in the file name for identity files.
-T	Generates a trusted certificate. By default, the program generates a non-trusted certificate.

Item	Description
-V nkeys	Generates parameters and keys for the Mu-Varadharajan (MV) identification scheme.

Exit Status

This command returns the following exit values:

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

Security

Access Control: You must be part of the system group to run this command.

Auditing Events: N/A

Examples

1. To generate RSA-SHA cryptographic keys, enter:

```
ntp-keygen -c RSA-SHA
```

2. To print a list of the peers known to the server as well as a summary of their state, enter:

```
ntpd -p
```

Output similar to the following appears:

```
Using OpenSSL version 90804f
Generating RSA keys (512 bits)...
RSA                               3 1 2
Generating new host file and link
ntpkey_host_aixfvt12->ntpkey_RSAkey_aixfvt12.3444540821
Using host key as sign key
Generating certificate RSA-SHA
X509v3 Basic Constraints: critical,CA:TRUE
X509v3 Key Usage: digitalSignature,keyCertSign
Generating new cert file and link
ntpkey_cert_aixfvt12->ntpkey_RSA-SHAcert_aixfvt12.3444540821
```

Files

Item	Description
/usr/sbin/ntp4/ntp-keygen4	Contains the ntp-keygen command. The default symbolic link to the NTP version 4 binary from /usr/sbin directory. <code>/usr/sbin/ntp-keygen --> /usr/sbin/ntp4/ntp-keygen4</code>

ntpq4 Daemon

Purpose

Starts the standard Network Time Protocol (NTP) query program.

Syntax

```
ntpq [-4 -6 -d -i -n -p] [-c command] [host] [...]
```

Description

The **ntpq** program is used to monitor the NTP daemon, the **ntpd** operations, and determine performance. It uses the standard NTP version 3 mode 6 control message formats defined by RFC 1305. The same formats are used in NTP version 4.

The program can be run either in interactive or controlled mode using command line arguments. The raw and printed output options enables you to assemble the requests to read and write arbitrary variables. The **ntpq** program can also obtain and print a list of peers in a common format by sending multiple queries to the server.

If one or more request options are included in the command line when the **ntpq** program is executed, each request will be sent to the NTP servers running on the hosts given by the command line arguments, or on localhost by default. If no request options are given, the **ntpq** utility will attempt to read commands from the standard input and execute them on the NTP server running on the first host given by the command line, again defaulting to localhost when no other host is specified. The **ntpq** utility will prompt for commands if the standard input is a terminal device.

The **ntpq** utility uses NTP mode 6 packets to communicate with the NTP server, and hence can be used to query any compatible server on the network which permits it.

In the instance where a host name is expected, and when you add a **-4** qualifier preceding the host name, the utility forces the DNS resolution to the IP version 4 namespace. Similarly, and a **-6** qualifier forces DNS resolution to the IP version 6 namespace.

Specifying a command line option other than **-i** or **-n** will cause the specified query or queries to be sent to the indicated host or hosts immediately. Otherwise, the **ntpq** utility will attempt to read interactive format commands from the standard input.

Flags

Item	Description
-4	Forces DNS resolution of the host names on the command line to the IP version 4 namespace.
-6	Forces DNS resolution of the host names on the command line to the IP version 6 namespace.
-c	The following argument is interpreted as an interactive format command and is added to the list of commands to be executed on the specified host or hosts. Multiples of the -c options might be added.
-d	Enables the debugging mode.
-i	Forces the ntpq utility to operate in interactive mode. The results will be written to the standard output and the commands are read from the standard input.
-n	Outputs all host addresses in dotted-quad numeric format rather than converting to the canonical host names.

Item	Description
-p	Prints a list of peers known to the server as well as a summary of their state. This is equivalent to the peers interactive command.

Parameters

Item	Description
Host...	Specifies the hosts.

Exit Status

This command returns the following exit values:

- 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 start the Network Time Protocol query program in interactive mode, enter:

```
ntpq -i
```

- To print a list of peers known to the server and the summary of their state, enter:

```
ntpq -p
```

Output similar to the following is displayed:

remote	refid	st	t	when	poll	reach	delay	offset	jitter
ausgsa.austin.ibm.com	9.41.253.167	2	u	19	64	377	285.962	-8.792	2.989

ntpq Internal Commands

Interactive Format Commands

Interactive format commands consist of a keyword followed by a maximum of 4 arguments. You must type only the required number of characters of the keyword to uniquely identify the command. The output of a command is normally sent to the standard output. You can also opt to send the output of individual commands by appending a greater than symbol (>), followed by a file name, to the command line. A number of interactive format commands are executed entirely within the **ntpq** program and do not send the NTP mode 6 requests to a server.

Item	Description
? [command_keyword] or help [command_keyword]	A question mark (?) by itself will print a list of all the command keywords known to this incarnation of ntpq. A question mark (?) followed by a command keyword will print function and the use of the command.

Item	Description
addvars <i>variable_name</i> [= <i>value</i>] [...] or rmvars <i>variable_name</i> [...] or clearvars	The data carried by the NTP mode 6 messages consists of a list of items of the form <i>variable_name</i> = <i>value</i> , where the equals symbol (=) value is ignored, and can be omitted, in requests to the server to read variables. The ntpq program maintains an internal list in which data to be included in control messages can be assembled, and sent using the readlist and writelist commands described below. The addvars command allows variables and their optional values to be added to the list. If more than one variable is to be added, the list must be separated by using commas, and must not contain any white space. The rmvars command can be used to remove individual variables from the list, while the clearlist command removes all variables from the list.
cooked	Causes the output from query commands to be cooked, so that variables which are recognized by the ntpq command will have their values reformatted for human consumption. The ntpq program marks the variables with a trailing question mark symbol (?) when the variable value cannot be decoded.
debug <i>more</i> <i>less</i> <i>no</i>	Adjusts the level of ntpq debugging. The default is debug no .
delay <i>milliseconds</i>	Specifies a time interval to be added to timestamps included in requests which require authentication. This is used to enable server reconfiguration over long delay network paths or between machines whose clocks are not synchronized.
host <i>hostname</i>	Sets the host to which future queries will be sent. Hostname may be either a host name or a numeric address.
hostnames [<i>yes</i> <i>no</i>]	If yes is specified, host names are printed in the information display. If no is specified, numeric addresses are printed. The default is yes, unless modified using the command line -n switch.
keyid <i>keyid</i>	Specifies the key number to be used to authenticate configuration requests. This must correspond to a key number the server has been configured to use for this purpose.
ntpversion <i>1</i> <i>2</i> <i>3</i> <i>4</i>	Sets the NTP version number which ntpq claims in packets. The default is 2. The mode 6 control messages did not exist in NTP version 1.
passwd	Prompts for a password that will not be echoed, and which will be used to authenticate configuration requests. The password must correspond to the key configured for NTP server for this purpose.
quit	Exits ntpq .
raw	Prints the output of query commands received from the remote server. The only formatting done on the data is transforming non-ASCII data to a printable form.
timeout <i>milliseconds</i>	Specifies a timeout period for responses to server queries. The default is 5000 milliseconds. Since ntpq retries each query once after a timeout, the total waiting time for a timeout will be twice the timeout value set.

Control Message Commands

Each association known to an NTP server has a 16 bit integer association identifier. The NTP control messages that carry peer variables must identify the corresponding peer values, which are its association ID. An association ID 0 indicates that the variables are system variables, and their names are drawn from a separate name space.

Control message commands result in one or more NTP mode 6 messages being sent to the server, and cause the data returned to be printed in a format. Most commands currently implemented send a single message and expect a single response. The current exceptions is the **peers** command, which will send a pre-programmed series of messages to obtain the data it needs, and the **mreadlist** and **mreadvar** commands, which will iterate over a range of associations.

Item	Description
associations	<p>Obtains and prints a list of association identifiers and peer statuses for in-spec peers of the server being queried. The list is printed in columns.</p> <p>The first column indicates the index numbering of associations from 1. The second column specifies the actual association identifier returned by the server, and the third column indicates the status word for the peer.</p> <p>This is followed by a number of columns containing data decoded from the status word. See the peers command for a decode of the condition field.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. The data returned by the associations command is cached internally in ntpq. 2. The index in the form &index is used when dealing with servers that use association identifiers wherein the subsequent commands require an association identifier as an argument.
<p>clockvar [assocID] [variable_name [= value [...]] [...]</p> <p>cv [assocID] [variable_name [= value [...]] [...]</p>	<p>Requests the server to send a list of the server's clock variables. Servers, which have a radio clock or other external synchronization will respond positively to this. If the association identifier is omitted or is a zero, you are requesting for the system clock variables and will get a positive response from all servers with a clock. If the server treats clocks as pseudo-peers, and hence can possibly have more than one clock connected at once, referencing the appropriate peer association ID will show the variables of a particular clock. Omitting the variable list will cause the server to return a default variable display.</p>
lassociations	<p>Obtains and prints a list of association identifiers and peer statuses for all associations for which the server is maintaining state. This command differs from the associations command only for servers which retain state for out-of-spec client associations (i.e., fuzzballs). Such associations are normally omitted from the display when the associations command is used, but are included in the output of lassociations.</p>
lpassociations	<p>Prints data for all associations, including out-of-spec client associations, from the internally cached list of associations. This command differs from passociations only when dealing with fuzzballs.</p>
lpeers	<p>Similar to R peers, except that a summary of states of all associations that the server is maintaining are printed. This can produce a much longer list of peers from fuzzball servers.</p>
<p>mreadlist assocID assocID mrl assocID assocID</p>	<p>Similar to the readlist command, except that the query is done for a range of (nonzero) association IDs. This range is determined from the association list cached by the most recent associations command.</p>

Item	Description
mreadvar assocID assocID [variable_name [= value[...]	Similar to the readvar command, except that the query is done for a range of (nonzero) association IDs. This range is determined from the association list cached by the most recent associations command.
mrvar assocID assocID [variable_name [= value[...]	
opeers	An old form of peers command with the reference ID replaced by the local interface address.
passociations	Displays association data concerning in-spec peers from the internally cached list of associations. This command performs identically to the associations except that it displays the internally stored data rather than making a new query.
peers	Obtains a current list peers of the server, along with a summary of each peer's state. Summary information includes the address of the remote peer, the reference ID (0.0.0.0 if this is unknown), the stratum of the remote peer, the type of the peer (local, unicast, multicast or broadcast), when the last packet was received, the polling interval, in seconds, the reachability register, in octal, and the current estimated delay, offset and dispersion of the peer, all in milliseconds.
pstatus assocID	Sends a read status request to the server for the given association. The names and values of the peer variables returned will be printed. Note that the status word from the header is displayed preceding the variables, both in hexadecimal and in pidgeon English.
readlist [assocID] rl [assocID]	Requests that the values of the variables in the internal variable list be returned by the server. If the association ID is omitted or is 0 the variables are assumed to be system variables. Otherwise they are treated as peer variables. If the internal variable list is empty a request is sent without data, which should induce the remote server to return a default display.
readvar assocID variable_name [= value] [...] rv assocID [variable_name [= value] [...]	Requests that the values of the specified variables be returned by the server by sending a read variables request. If the association ID is omitted or is given as zero the variables are system variables, otherwise they are peer variables and the values returned will be those of the corresponding peer. Omitting the variable list will send a request with no data which should induce the server to return a default display. The encoding and meaning of the variables derived from NTPv3 is given in RFC-1305; the encoding and meaning of the additional NTPv4 variables are given later in this page.
writevar assocID variable_name [= value [...]	Similar to the readvar request, except that the specified variables are written.
writelist [assocID]	Similar to the readlist request, except that the internal list of variables are written.

Files

/usr/sbin/ntp4/ntpq4

Contains the **ntpq4** command.

/usr/sbin/ntpq-->/usr/sbin/ntp4/ntpq4

Default symbolic link to NTP version 4 binary file from the **/usr/sbin** directory.

ntptrace4 Command

Purpose

Traces a chain of Network Time Protocol (NTP) hosts back to their master time source.

Syntax

```
ntptrace [ -n ] [ server ]
```

Description

The **ntptrace** command determines the time source for the Network Time Protocol (NTP) server and follows the chain of NTP servers back to their master time source. If no arguments are provided, it starts with localhost. Following is an example of the output of the **ntptrace** command:

```
% ntptrace
localhost: stratum 4, offset 0.0019529, sync distance 0.144135
server2ozo.com: stratum 2, offset 0.0124263, sync distance 0.115784
usndh.edu: stratum 1, offset 0.0019298, sync distance 0.011993, refid 'WWVB'
```

On each line, the fields from left to right are host name, host stratum, time offset between that host and the local host as measured by the **ntptrace** command. This is why it is not always zero for "localhost", host synchronization distance, and (applies only for the stratum-1 servers) the reference clock ID. All times are given in seconds. Note that the stratum is the server hop count to the primary source, while the synchronization distance is the estimated error relative to the primary source. These terms are precisely defined in RFC-1305.

Flags

Item	Description
-n	Turns off the printing of host names; instead, host IP addresses are printed. This may be useful if a nameserver is down.

Parameters

Item	Description
<i>Server</i>	Specifies the server. The default is the local host.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion
>0	An error occurred

Security

Access Control: You must be part of the system group to run this command.

Auditing Events: N/A

Examples

1. To trace the time source for the local host NTP server, enter:

```
ntptrace
```

Output similar to the following appears:

```
loopback: stratum 5, offset 0.000076, synch distance 0.18291
ganga08.in.ibm.com: stratum 4, offset -0.001854, synch distance 0.30600
ganga10.in.ibm.com: stratum 3, offset 0.000251, synch distance 0.30550
ausgsa.austin.ibm.com: stratum 2, offset -0.010158, synch distance 0.01921
gsantp.austin.ibm.com: stratum 1, offset 0.016067, synch distance 0.00000, refid
'GPS'
```

Files

/usr/sbin/ntp4/ntptrace4

Contains the **ntptrace4** command.

/usr/sbin/ntptrace-->/usr/sbin/ntp4/ntptrace4

Default symbolic link to NTP version 4 binary file from the **/usr/sbin** directory.

nulladm Command

Purpose

Creates active accounting data files.

Syntax

```
/usr/sbin/acct/nulladm [ File ... ]
```

Description

The **nulladm** command creates the file specified by the *File* parameter, gives read (r) and write (w) permission to the file owner, and group and read (r) permission to other users, and ensures that the file owner and group are **adm**. Various accounting shell procedures call the **nulladm** command. A user with administrative authority can use this command to set up the active data files, such as the **/var/adm/wtmp** file.

Note: You should not share accounting files among nodes in a distributed environment. Each node should have its own copy of the various accounting files.

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 accounting commands.
/var/adm/acct/sum	Contains accounting data files.

number Command

Purpose

Displays the written form of a number.

Syntax

number

Description

The **number** command translates the numerical representation of an entered number to the written form. The largest number it can translate accurately contains 66 digits. For example:

```
12345678
twelve million.
three hundred forty five thousand.
six hundred seventy eight.
```

In the above example, you entered 12345678 and the computer translated it to twelve million three hundred forty five thousand six hundred seventy eight.

The **number** command does not prompt you for a number. Once started, it simply waits for input. To exit the program, press the Interrupt (Ctrl-C) or End Of File (Ctrl-D) key sequence.

Files

Item	Description
<code>/usr/games</code>	Contains the system games.

>nvmesed Command

Purpose

Manages the nonvolatile memory express (NVMe) self-encrypting drives (SED).

Syntax

To set a user pin in platform keystore (PKS):

```
nvmesed -K -l nvmeX -n user_pin [ -t tag]
```

To initialize a drive to enable SED:

```
nvmesed -I -l nvme# -s sid_pin -a admin_pin -u user_pin
```

To revert a drive to disable SED:

```
nvmesed -R -l nvme# -a <admin_pin>
```

To return a drive to the original factory state (OFS):

```
nvmesed -O -l nvme# -s <sid_pin>
```

```
nvmesed -O -l nvme# -p psid_pin
```

To change user pins on a drive:

```
nvmesed -C -l nvme# -a admin_pin -n new_admin_pin
```

```
nvmesed -C -l nvme# -u user_pin -n new_user_pin
```

```
nvmesed -C -l nvme# -s sid_pin -n new_sid_pin
```

```
nvmesed -C -l nvme# -s sid_pin -r
```

To unlock a drive:

```
nvmesed -U -l nvme# -a admin_pin
```

```
nvmesed -U -l nvme# -u user_pin
```

To query the status of an SED:

```
nvmesed -Q -l nvme#
```

```
nvmesed -Q -l nvme# -a admin_pin
```

```
nvmesed -Q -l nvme# -u user_pin
```

Description

Starting from AIX 7.3 with Technology level 1, the AIX operating system supports NVMe self-encrypting drives (SED). The NVMe SED encrypts the user data with a secret pin. The pin is stored in platform keystore (PKS), which is a secure nonvolatile storage in Power Systems servers. As the pin is stored outside the drive, even when the drive is lost or stolen, the data in the drive is rendered meaningless without the pin.

The AIX device driver provides the pin to the drive, when required. The **nvmesed** user space utility handles the SED management. The pin cannot be used similar to the Media Encryption Key (MEK) that is used to encrypt the data. The MEK can be deciphered or unwrapped only when the pin is known. The drive only retains a wrapped MEK that was created by using the pin to encrypt a hardware-generated plain text MEK. The plain text MEK cannot be accessed from outside the drive and exists only in the drive after a valid pin is entered.

The `nvmesed` command manages the following SED functions:

- NVMe drives are shipped without self-encryption. This state is known as Original Factory State (OFS). You can use the `nvmesed` command to enable self-encryption to protect the drive data with a pin. In most cases, self-encryption preserves existing data on the drive.
- You can use the `nvmesed` command to revert the drive to its original state before initializing SED. This usage includes the following cases:
 - Return to the OFS by securely purging all the user data on the drive.
 - Revert to the original state before initializing SED by turning off the pin-controlled encryption while preserving the user data.

You can use the `nvmesed` command to change the pin periodically. The drive does not store the pin, but stores only a cryptographic hash of the pin so that it can determine whether the specified pin is valid. Thus, if you change the pin on the drive, a new hash of the pin on the drive is stored on the drive. Management includes the following authorities and types of pins:

Security Identifier (SID)

The SID authority represents the drive owner. Enabling self-encrypting drives (SED) and returning the drive to OFS requires the SID authority. If the SID pin is lost, you must return the drive to OFS by using a pointer to the security identifier (PSID) pin that is a fixed pin and is printed on the drive label. You can establish a new set of pins by enabling the SED again as a new drive.

Admin

The Admin authority manages the data encryption, power on locking, and various access control policies of the SED. Enabling SED, reverting the SED, and unlocking the drive requires the Admin authority.

User

The locked device can only be unlocked by a user authority. The AIX operating system stores only the user pin with the most restricted capabilities in the PKS. In addition, both Admin and User authorities can query the status of a SED drive.

The following prerequisites must be met to use the `nvmesed` utility:

- PKS must be enabled for the AIX logical partition in the Hardware Management Console (HMC).
- The logical partition must be running AIX 7.3 or later.
- The host must be Power10 processor-based servers (or later) that support NVMe SED.
- The drives must be NVMe drives that support SED. SED-capable drives other than NVMe do not suffice the requirement.

Review the following practices for using the `nvmesed` utility:

- Each of the three pins must be unique and must be composed of 32 bytes of ASCII characters.
- Maintain a record of all pins, including previously used pins, and track pin activities for future reference.
- PKS stores only the User pin. An AIX logical partition contains a single user pin. All drives in the logical partition can use the same user pin. Therefore, use the same SID and Admin pins on a logical partition for ease of administration. SID and Admin pins are specified in the `nvmesed` command and are not stored in PKS.

If you are moving a drive to a system that supports SED, you must reinitialize the drive. If you know the existing SID and Admin pins, you can reinitialize the drive while preserving the existing data. If the data cannot be preserved, you must return the drive to the OFS before you can re-enable SED.

Flags

-a *admin_pin*

Specifies the existing Admin pin.

-C

Changes pins on a specific drive.

Each authority can change its own pin if the current pin is specified.

-I *nvmeX* | *nvme#*

Specifies the NVMe drive. *nvmeX* is any available drive on which SED is supported. *nvme#* is a specific target drive on which the `nvmesed` command is run.

Before you change the User pins, one drive at a time, you can first set the new pin on PKS as an alternative pin. After you change the User pin on all drives to the new User pin, the primary PKS pin is changed to the new pin, and the alternative pin must be erased. Because PKS does not store SID and Admin pins, these pins might not need coordination with PKS.

-I

Initializes a drive to enable SED.

-K

Sets a user pin that is stored in the PKS.

You must use this flag as the first step before you enable SED for the first time in the logical partition. Select a user pin that is shared by all drives and store it in PKS.

-n *pin*

Specifies a new SID, Admin, or User pin that must be used.

-O

Returns the drive to the original factory state (OFS).

You must specify the SID pin that was set during initialization or the PSID pin that is printed on drive label. After the drive is returned to the OFS successfully, the SID, Admin, and User pins become invalid on the drive. All existing data is purged cryptographically, and the drive can be considered as a new drive.

-p *psid_pin*

Specifies the existing PSID pin.

-Q

Queries the SED status.

-R

Disables SED and reverts the drive such that the data can be accessed without any pin.

-r

Omits the SID pin and returns the drive into OFS with unprotected data and no pins.

-s *sid_pin*

Specifies the existing SID pin.

-t *tag*

Specifies an optional tag which is an 8-character alternate name for the pin that is used when PKS pins are queried. PKS displays only this tag but not the 32-byte pin. The tag must be unique so that it can easily correlate to the pin.

-u *user_pin*

Specifies an optional tag which is an 8-character alternate name for the pin that is used when PKS pins are queried. PKS displays only this tag but not the 32-byte pin. The tag must be unique so that it can easily correlate to the pin.

-U

Unlocks the drive manually.

If the PKS pin is not available or does not match the User pin that was set on the drive, the drive remains locked after a power cycle. You can use this flag to unlock the drive manually by specifying either the User pin or the Admin pin that was set during initialization.

Error messages

start_session(*xxx_sp*) failed, status=0x30/0x1

This error is observed when drive has been partially impacted by an earlier command that altered the pin, which failed at a later stage.

cannot start_session(*locking_sp*) to unlock, status=0x30/0xc

This error is observed when the drive is not SED-enabled but the requested action needs the drive to be SED-enabled. To confirm, use the Query (-Q) option.

start_session(*xxx_sp*) failed, status=0x30/0x12

This error is observed when the drive fails to authenticate the pin several times in a row and the authority (SID, Admin, or User) is locked out.

Note: Do not run the `nvmesed` command unless you know the correct pins. If the authentication fails several times and results in this error, a valid pin cannot gain access until the drive is power cycled that can be disruptive.

nvmesed -Q -l nvme0

This error is observed when `nvme0` is not an SED-enabled drive and does not support security protocol.

Maintenance requirements for the SED

Consider the following maintenance requirements for the SED:

Namespace management

The same SED state is applicable to the entire drive and is shared by all namespaces, including the ones that will be created in the future. When namespaces do not exist, a drive can either enable or

disable the SED. Certain namespace management operations cannot be performed on a locked drive. SED cannot be enabled or disabled for each namespace separately.

Format

The format operation (on any namespace) fails if the drive is locked.

Sanitize

The sanitize operation is used to purge all data on the drive. An SED-enabled drive turns off the sanitize command and the data can be purged by returning the drive to OFS.

Device Self-test (diagnostics)

The device self-test is used for diagnostics. This command might fail if the drive is locked, depending on the drive vendor.

Moving a SED-enabled drive from another system

It is recommended to reinitialize the drive. In most cases, the drive can be reinitialized while preserving existing data if the SID and admin pins are known. The drive must be returned to OFS before the drive can be enabled for SED in an AIX operating system (if the data cannot be preserved).

Disabling Key Store

If the partition no longer owns or plans to own SED enabled drives, any PKS pins must be erased before the keystore can be disabled from the HMC. To disable the keystore, shut down the partition and set the keystore size to 0 by running the following commands:

```
# erase primary pin (-i 0)
nvmesed -K -l nvmeX -e -i 0
```

```
# erase alternate (secondary) pin (-i 1)
nvmesed -K -l nvmeX -e -i 1
```

Examples

The pins chosen for the user, admin, and SID authorities in the examples are AIX_PIN_USER, AIX_PIN_ADMIN, and AIX_PIN_SID. Pins from an earlier enablement, which may or may not be from an AIX environment, are identified as OLD_PIN_USER, OLD_PIN_ADMIN, and OLD_PIN_SID.

Note: It is recommended to, use 32-character pins during deployment.

1. To enable SED on a new drive, complete the following steps:
 - a. Verify that the drive is not SED-enabled by running the following command:

```
# nvmesed -Q -l nvme1
nvme1: SED_Enabled = No PKS pin, Drive Locked = No
```

The command output displays that the PKS pin does not exist. If you run the same query command with the user pin that you want to use, the command output displays that the drive is not SED-enabled.

```
# nvmesed -Q -l nvme1 -u AIX_PIN_USER
nvme1: SED_Enabled = No, Drive Locked = No
```

- b. Set the PKS pin by running the following command:

```
# nvmesed -K -l nvme1 -n AIX_PIN_USER
successfully set keystore pin (pri)
```

- c. If you run the query command without specifying the PKS pin, the command output displays that the drive is not SED-enabled.

```
# nvmesed -Q -l nvme1
nvme1: SED_Enabled = No, Drive Locked = No
```

- d. Enable self-encryption on the nvme1 drive by running the following command. When the operation is complete, the disk description is updated with the encrypted status. The data on the drive is preserved. In this example, only a single disk is present under the nvme1 (hdisk1) drive. However, if the disk is in use, the update in the encryption status cannot be determined until the disk is reconfigured or the system is rebooted.

```
# nvmesed -I -l nvme1 -s AIX_PIN_SID -a AIX_PIN_ADMIN -u AIX_PIN_USER
start_session(admin), hsn=0x1234, tsn=0x1024
MSID PIN, len = 32:83WX4EL50X2WYALH6KLWLM8MDLZN4FCK
Locking SP state = 8
max_ranges = 8
initialization completed successfully, drive encryption is enabled.
updating hdisk1...
# nvmesed -Q -l nvme1
nvme1: SED_Enabled = Yes, Drive Locked = No
# lsdev|grep hdisk1
hdisk1 Available 01-00 NVMe Encrypted 4K Flash Disk
```

2. To move an SED-enabled drive from another system to an AIX system, complete the following steps. In this example, you know the existing pins (OLD_PIN_XXX) from another system but the pins are different from the pins (AIX_PIN_XXX) that you want to use in the AIX system.

- a. Query the self-encryption status by running the following command:

```
# nvmesed -Q -l nvme1
nvme1: SED_Enabled = Not Authorized, Drive Locked = No
```

The command output reports "Not Authorized," which means that the user pin that is set on the drive does not match the AIX pin in the PKS. The query returns SED_Enabled=Yes only when the drive is SED-enabled.

- b. To determine whether the self-encryption conforms with the AIX operating system, you need a valid pin to check various SED states in the drive. If the SED_Enabled field reports a value other than No, the drive might have some form of SED enablement.

```
# nvmesed -Q -l nvme1 -u AIX_PIN_USER
cannot start_session(locking_sp) to query, status=0x30/0x1
nvme1: SED_Enabled = Not Authorized, Drive Locked = No
```

When such a drive is configured in the AIX system, the command output logs errors that indicate that the system might fail to unlock the drive since it does not have the correct pin.

```
LABEL: NVME_ERR4
Resource Name: nvme1
Description
NVME CONTROLLER SOFTWARE ERROR
...
ADDITIONAL INFORMATION
Drive has unknown SED enablement and may lock up
```

- c. Re-initialize the drive by specifying the AIX pins for the AIX system in the following command. Even if the required pins are set for transferred drive, you must reinitialize the drive in the AIX system. You must use the existing SID and admin pins, but apply the AIX user pin.

```
# nvmesed -I -l nvme1 -s OLD_PIN_SID -a OLD_PIN_ADMIN -u AIX_PIN_USER
start_session(admin), hsn=0x1234, tsn=0x102f
MSID PIN, len = 32:83WX4EL50X2WYALH6KLWLM8MDLZN4FCK
Locking SP state = 9
locking is active, trying re-init...
reverted Locking SP back to inactive
Locking SP state = 8
max_ranges = 8
initialization completed successfully, drive encryption is enabled.
updating hdisk1...
```

The output shows "locking is active," which indicates that the prior enablement is turned off and then re-activated.

- d. Change the SID and admin pins to match the AIX pins by running the following commands:

```
# nvmesed -C -l nvme1 -s OLD_PIN_SID -n AIX_PIN_SID
change pin completed successfully.
# nvmesed -C -l nvme1 -a OLD_PIN_ADMIN -n AIX_PIN_ADMIN
change pin completed successfully.
# nvmesed -Q -l nvme1
nvme1: SED_Enabled = Yes, Drive Locked = No
```

3. To return the drive to OFS, run the following commands:

Note: All data is lost in the process. The drive shows it is no longer encrypted.

```
# nvmesed -O -l nvme1 -s AIX_PIN_SID
ofs completed successfully, drive is at factory default.
updating hdisk1...
# lsdev|grep hdisk1
hdisk1 Available 01-00 NVMe 4K Flash Disk
# nvmesed -Q -l nvme1
nvme1: SED_Enabled = No, Drive Locked = No
```

4. To revert an SED-enabled drive to a non-encrypted state and to disable pin-based protection, run the following command:

```
# nvmesed -R -l nvme1 -a AIX_PIN_ADMIN
revert completed successfully, drive encryption is disabled.
updating hdisk1...
# nvmesed -Q -l nvme1
nvme1: SED_Enabled = No, Drive Locked = No
```

This command preserves the existing data. A reverted drive retains the earlier SID pin but loses admin and user pins, whereas an OFS drive does not retain any pins. If you can try to re-initialize an OFS drive with a new set of pins by using the following command, the command reports an error because the SID pin (AIX_PIN_SID) is already set.

```
# nvmesed -I -l nvme1 -u AIX_PIN_NEW_USER -a AIX_PIN_NEW_ADMIN -s AIX_PIN_NEW_SID
start_session(admin), hsn=0x1234, tsn=0x10e8
MSID PIN, len = 32:83WX4EL50X2WYALH6KLWLM8MDLZN4FCK
start_session(admin_sp) failed, status=0x30/0x1
cannot start session as SID, try another SID pin or reset to OFS
```

You can use the pin with the **-s** flag in the re-initialization command or clear the pin on the drive by using the **-C** or **-r** flags and try to re-initialize an OFS drive.

```
# nvmesed -C -l nvme1 -s AIX_PIN_SID -r
reset sid pin successfully.
# nvmesed -I -l nvme1 -u AIX_PIN_NEW_USER -a AIX_PIN_NEW_ADMIN -s AIX_PIN_NEW_SID
start_session(admin), hsn=0x1234, tsn=0x10e8
MSID PIN, len = 32:83WX4EL50X2WYALH6KLWLM8MDLZN4FCK
Locking SP state = 8
max_ranges = 8
initialization completed successfully, drive encryption is enabled.
updating hdisk1...
```

To change the existing SID pin to the new SID pin, enter the following command:

```
# nvmesed -I -l nvme1 -u AIX_PIN_NEW_USER -a AIX_PIN_NEW_ADMIN -s AIX_PIN_SID
# nvmesed -C -l nvme1 -s AIX_PIN_SID -n AIX_PIN_NEW_SID
```

5. To move an SED-enabled drive with unknown pins from another system to the AIX system, complete the following steps:

- a. Return the drive to OFS by running the following commands:

```
# nvmesed -O -l nvme1 -s AIX_PIN_SID
cannot start_session(admin_sp) to ofs, status=0x30/0x1
# nvmesed -O -l nvme1 -s OLD_PIN_SID
cannot start_session(admin_sp) to ofs, status=0x30/0x1
```

- b. Locate the PSID pin that is printed on the drive label and run the following command:

```
# nvmesed -Q -l nvme1 -p QLUVL1FD3N8299FTM31VFH0ACTPUDHCN
ofs completed successfully, drive is at factory default.
updating hdisk1...
```

- c. Enable self-encryption as shown in [Example 1](#).
6. To change the user pin for two SED-enabled drives where the user pin is set to AIX_PIN_USER as the PKS primary pin, complete the following steps:
 - a. Query the drives to determine the status of encryption and PKS primary and secondary pins by running the following commands:

```
# nvmesed -Q -l nvme1
nvme1: SED_Enabled = Yes, Drive Locked = No
# nvmesed -Q -l nvme2
nvme2: SED_Enabled = Yes, Drive Locked = No
# nvmesed -K -l nvme1 -d
Keystore pin (pri): set on Wed Jun 22 11:17:28 2022
Keystore pin (sec): erased
```

- b. Set the PKS secondary pin to a new pin (AIX_PIN_NEW_USER) by running the following command:

```
# nvmesed -K -l nvme1 -n AIX_PIN_NEW_USER -i 1
successfully set keystore pin (sec)
```

The PKS primary pin is still set to the old pin.

- c. Change the user pin on each drive by running the following commands:

```
# nvmesed -C -l nvme1 -u AIX_PIN_USER -n AIX_PIN_NEW_USER
change pin completed successfully.
# nvmesed -C -l nvme2 -u AIX_PIN_USER -n AIX_PIN_NEW_USER
change pin completed successfully.
```

- d. Change the PKS primary pin to the new pin by running the following command:

```
# nvmesed -K -l nvme1 -n AIX_PIN_NEW_USER -i 0
successfully set keystore pin (pri)
```

- e. Clear the PKS secondary pin and verify that the pin is cleared and only the PKS primary pin is set by running the following commands:

```
# nvmesed -K -l nvme1 -e -i 1
successfully cleared keystore pin (sec)
# nvmesed -K -l nvme1 -d
Keystore pin (pri): set on Wed Jun 22 11:33:09 2022
Keystore pin (sec): erased
```

- f. Query each drive by using new PKS pin and verify that the SED_Enabled field is set to Yes by running the following commands:

```
# nvmesed -Q -l nvme1
nvme1: SED_Enabled = Yes, Drive Locked = No
# nvmesed -Q -l nvme2
nvme2: SED_Enabled = Yes, Drive Locked = No
```

If the old pin is used to query, the command fails and the SED_enabled field reports Not Authorized.

Limitation

SED-enabled drives are dependent on the PKS to save the keys to access SED drive content for that partition. Each partition can have its own unique PKS keys. However HMC allows multiple profiles for the same partition and hence the same PKS content. Thus, the HMC does not prevent multiple operating system instances (located on different boot disks in different profiles) from accessing the same PKS content. The keys might be shared and potentially modified. Sharing a partition's PKS for SED between different operating system instances is not supported. You can update the operating system to a newer level on an SED-enabled drive.



nxstat Command

Purpose

The **nxstat** command displays usage statistics of on-chip hardware accelerators that can implement GZIP functions.

Syntax

```
nxstat {-S | [-AUWT] [-a accel_type] [-u unit_index] [-w window_index] [-t window_type] interval [count]}
```

Description

Applications in an AIX logical partition use the `zlibNX` library to open a communication channel with the GZIP accelerator. This communication channel is also known as a *window*. For more information about the type of communication channels, see [Nest Accelerators](#).

The low-level usage statistics per window might include a large amount of data that might not be relevant for a system administrator. The **nxstat** command can aggregate the usage statistics to provide global usage statistics for a specific accelerator type or for a specific hardware accelerator unit based on the options and arguments that are specified with the command.

The following values are displayed in the usage statistics for each aggregation level along with the window type and corresponding credits:

- **Opens**: Number of open and close operations on a window.
- **Bytes**: Number of bytes processed by the hardware accelerator unit.
- **Faults**: Number of page faults that are detected by the hardware accelerator unit.
- **Intrs**: Number of completion interrupts. This value is optional for accelerator operations.
- **Polls**: Number of completion interrupts that are detected during the polling phase of the **nxstat** command.
- **Waits**: Number of operations that were completed after the caller was actually blocked. The total of **Polls** and **Waits** values represent the total number of operations that were completed during the specific time interval.

The kernel service that waits for completion of active GZIP operations, first polls a completion bit for a period of time before blocking the thread. Total count of **Polls** and **Waits** values provide insight about efficiency and usefulness of the polling phase.

The **nxstat** command aggregates the usage statistics of a specific accelerator type during a specified time interval. The **nxstat** command can also display the accelerator usage statistics for successive time intervals. The **-A**, **-U**, **-T**, and **-W** flags determine the aggregation levels. If more than one flags are specified, for example, **-AU**, the **nxstat** command displays usage statistics for both aggregation levels. The **-a**, **-u**, **-t**, **-w** flags are used to limit the statistics data to specific objects such as accelerator unit, window, and window type.

Flags

>|-s

Displays types of accelerators that are available. |<

-A

Displays usage statistics for a specific type of accelerator. This is the default option.

-U

Displays usage statistics for a specific hardware accelerator unit.

-T

Displays usage statistics for a specific window type.

-W

Displays usage statistics for each window.

-a

Specifies the accelerator type for which the usage statistics is displayed. The **-a** flag value for the GZIP accelerator is 0. Currently, the only supported value for the **-a** flag is 0. Therefore, if you do not specify the **-a** flag, the default value of 0 is applied.

-u

Specifies the accelerator unit for which the usage statistics is displayed. You must specify the **-u** flag only with the **-U** flag. The accelerator unit index value for this flag is a number between zero and the number of accelerator units minus 1. For example, if the AIX logical partition has access to three accelerator units, valid values that can be specified for this flag are 0, 1, and 2.

If the **-u** flag is not specified along with the **-U** flag, the **nxstat** command displays the data for all the hardware accelerator units that can be accessed by the logical partition.

-w

Specifies a specific window for which the usage statistics is displayed. If the logical partition has access to **n** windows, the windows are numbered from 0 to n-1. You can specify the **-w** flag only with the **-W** flag.

-t

Specifies a specific window type (0 for Quality of Service (QoS), 1 for default) for which the usage statistics is displayed. You can specify the **-t** flag only with the **-T** flag.

Note: The **-u**, **-w** and **-t** flags are mutually exclusive.

interval

Specifies the time interval, in seconds, during which the usage statistics must be captured. This flag is mandatory. By default, the kernel service captures usage statistics every 2 seconds, so you must specify a time interval more than 2 seconds to display useful data.

count

Specifies the number of successive time intervals for which you want to display the usage statistics. The default value is 1.

Examples

- To display total usage statistics for all the accelerator type and for all window types, run the following command:

```
nxstat 10 4
```

An output that is similar to the following example is displayed:

Accelerator Type	Cred	GZIP: Opens	Window Types: Bytes	2 Faults	Units: 1 Intrs	Credits: 72 Polls	Waits
ALL	72	266826	35.747 GB	0	17281	222795	61223
ALL	72	262812	35.638 GB	0	17524	220368	59641
ALL	72	300647	33.583 GB	0	15990	257351	58969
ALL	72	259405	35.557 GB	0	17111	216629	60048
** Average **							
ALL	72	272422	35.131 GB	0	16976	229285	59970

- To display accelerator level statistics by window type, run the following command:

```
nxstat -AT 10
```

An output that is similar to the following example is displayed:

Accelerator Type	Cred	GZIP: Opens	Window Types: Bytes	2 Faults	Units: 4 Intrs	Credits: 92 Polls	Waits
ALL	72	266826	35.747 GB	0	17281	222795	61223
ALL	72	262812	35.638 GB	0	17524	220368	59641
ALL	72	300647	33.583 GB	0	15990	257351	58969
ALL	72	259405	35.557 GB	0	17111	216629	60048
** Average **							
ALL	72	272422	35.131 GB	0	16976	229285	59970

QOS	12	0	49.892 GB	0	28695	4998	52476
DEF	80	0	4.406 GB	0	2561	422	4654

- To display unit level statistics by accelerator unit, run the following command:

```
nxstat -U 10 4
```

An output that is similar to the following example is displayed:

Unit Index	0:	SRAD ID	0	Window Types:	2	Credits:	23
Type	Cred	Opens	Bytes	Faults	Intrs	Polls	Waits
ALL	23	0	16.735 GB	0	9589	1551	17728
Unit Index	1:	SRAD ID	1	Window Types:	2	Credits:	23
ALL	23	0	12.438 GB	0	7219	1270	13058
Unit Index	2:	SRAD ID	2	Window Types:	2	Credits:	23
ALL	23	0	12.427 GB	0	7131	1310	13006
Unit Index	3:	SRAD ID	3	Window Types:	2	Credits:	23
ALL	23	0	12.430 GB	0	7230	1383	12935

- [>](#) To display types of accelerators that are available, run the following command:

```
nxstat -S
```

An output that is similar to the following example is displayed:

```
GZIP accelerator available
```

[|<](#)

O

The following AIX commands begin with the letter o.

od Command

Purpose

Displays files in a specified format.

Syntax

To display files using a type-string to format the output

```
od [ -v ] [ -A AddressBase ] [ -N Count ] [ -j Skip ] [ -t TypeString ... ] [ File ... ]
```

To display files using flags to format the output

```
od [ -a ] [ -b ] [ -c ] [ -C ] [ -d ] [ -D ] [ -e ] [ -f ] [ -F ] [ -h ] [ -H ] [ -i ] [ -I ] [ -l ] [ -L ] [ -o ]  
[ -O ] [ -p ] [ -P ] [ -s ] [ -v ] [ -x ] [ -X ] [ -S [N] ] [ -w [N] ] [ File ] [ + ] Offset [ . | b | B ]  
[ + ] Label [ . | b | B ] [ File ... ]
```

Description

The **od** command displays the file specified by the *File* parameter in the format specified. If the *File* parameter is not given, the **od** command reads standard input. Multiple types can be specified by using multiple **-bcCDdFfOoSstvXx** options.

In the first syntax format, the output format is specified by the **-t** flag. If no format type is specified, **-t o2** is the default.

In the second syntax format, the output format is specified by a combination of flags. The *Offset* parameter specifies the point in the file where the file output begins. By default, the *Offset* parameter is interpreted as octal bytes. If the **.** (dot) suffix is appended, the parameter is interpreted as a decimal; if the parameter begins with a leading **x** or **0x**, it is treated as a hexadecimal. If the **b** suffix is added to the parameter, it is interpreted in blocks of 512 bytes; if the **B** suffix is added to the parameter, it is interpreted in blocks of 1024 bytes.

The *Label* parameter is interpreted as a pseudo-address for the first byte displayed. If used, it is given in **()** (parentheses) following the *Offset* parameter. The suffixes have the same meanings as for the *Offset* parameter.

When the **od** command reads standard input, the *Offset* parameter and the *Label* parameter must be preceded by a **+** (plus sign).

The setting of environment variables such as **LANG** and **LC_ALL** affects the operation of the **od** command.

The **od** command copies each input file sequentially, writes them to the standard output and transforms the input data according to the types of output that are specified by using the **-t** flag. The default number of bytes in the input data correspond to the size of the type specifier characters **d**, **o**, **u**, or **x**. These type specifier characters allow you to specify an optional number of bytes in the input data that corresponds to the number of bytes in the **char**, **short**, **int**, and **long** data types in the C programming language. The number of bytes can also be specified by using an application (any program or use case that uses the **od** command) in which the character **C** is used for **char** data type, character **S** is used for **short** data type, character **I** is used for **int** data type, and character **L** is used for **long** data type. The type specifier characters supports the values 1, 2, 4 and 8 even though the data type as per the C programming language is not specified for the size of the input data. The type specifier characters also supports the decimal value similar to the **long long** data type in the C programming language.

Flags

The flags for the first format are:

Table 5. Flags to display files using a type-string to format the output

Item	Description
-A <i>AddressBase</i>	<p>Specifies the input offset base. The <i>AddressBase</i> variable is one of the following characters:</p> <ul style="list-style-type: none">d Offset base is written in decimal.o Offset base is written in octal.x Offset base is written in hexadecimal.n Offset base is not displayed. <p>Unless -A n is specified, the output line will be preceded by the input offset, cumulative across input files, of the next byte to be written. In addition, the offset of the byte following the last byte written will be displayed after all the input data has been processed. Without the -A <i>address_base</i> option and the [<i>offset_string</i>] operand, the input offset base is displayed in octal.</p>
-j <i>Skip</i>	<p>Jumps over the number of bytes given by the <i>Skip</i> variable before beginning to display output. If more than one file is specified, the od command jumps over the designated number of bytes of the concatenated input files before displaying output. If the combined input is not at least the length of the skip bytes, the od command will write a diagnostic message to standard error and exit non-zero status.</p> <p>By default, the value of the <i>Skip</i> variable is interpreted as a decimal number. With a leading 0x or 0X, the offset is interpreted as a hexadecimal number; otherwise, with a leading 0, the offset shall be interpreted as an octal number. If the characters b, k, or m are appended to the number contained by the <i>Skip</i> variable, the offset is equal to the value, in bytes, of the <i>Skip</i> variable multiplied by 512, 1024, or 1024*1024, respectively.</p>
-N <i>Count</i>	<p>Formats no more than the number of input bytes specified by the <i>Count</i> variable. By default, the value of the <i>Count</i> variable is interpreted as a decimal number. With a leading 0x or 0X, it is treated as a hexadecimal number. If it begins with a 0, it is treated as an octal number. The base of the address displayed is not implied by the base of the <i>Count</i> option-argument.</p>

Table 5. Flags to display files using a type-string to format the output (continued)

Item	Description
-t <i>TypeString</i>	<p>Specifies the output type. The <i>TypeString</i> variable is a string specifying the types to be used when writing out data. Multiple types can be concatenated within the same <i>TypeString</i> variable, and the -t flag can be specified more than once. Output lines are written for each type specified, in the order in which the type specification characters are given. The <i>TypeString</i> variable can consist of the following characters:</p> <p>a</p> <p>Displays bytes as named characters. Bytes with the least seven bits in the range of 0 through 01777 are written using the corresponding names for those characters.</p> <p>c</p> <p>Displays bytes as characters. The number of bytes transformed by the c type string is determined by the LC_CTYPE local category. Printable multibyte characters are written in the area corresponding to the first byte of the character; the two character sequence ** is written in the area corresponding to each remaining byte in the character, as an indication that the character is continued. The following nongraphic characters are used as C-language escape sequences:</p> <pre data-bbox="516 814 1471 1045"> \ Backslash \a Alert \b Backspace \f Form-feed \n New-line character \0 Null \r Carriage return \t Tab \v Vertical tab </pre> <p>d</p> <p>Displays bytes as signed decimals. By default, the od command transforms the corresponding number of bytes in the C-language type int. The d type string can be followed by an unsigned decimal integer that specifies the number of bytes to be transformed by each instance of the output type.</p> <p>An optional C, I, L, or S character can be appended to the d option, indicating that the conversion should be applied to an item of type char, int, long, or short, respectively.</p> <p>f</p> <p>Displays bytes as floating points. By default, the od command transforms the corresponding number of bytes in the C-language type double. The f type string can be followed by an unsigned decimal integer that specifies the number of bytes to be transformed by each instance of the output type.</p> <p>An optional F, D, or L character can be appended to the f option, indicating that the conversion should be applied to an item of type float, double, or long double, respectively.</p> <p>o</p> <p>Displays bytes as octals. By default, the od command transforms the corresponding number of bytes in the C-language type int. The o type string can be followed by an unsigned decimal integer that specifies the number of bytes to be transformed by each instance of the output type.</p> <p>An optional C, I, L, or S character can be appended to the o option, indicating that the conversion should be applied to an item of type char, int, long, or short, respectively.</p>

Table 5. Flags to display files using a type-string to format the output (continued)

Item	Description
	<p>u</p> <p>Display bytes as unsigned decimal. By default, the od command transforms the corresponding number of bytes in the C-language type int. The u type string can be followed by an unsigned decimal integer that specifies the number of bytes to be transformed by each instance of the output type.</p> <p>An optional C, I, L, or S character can be appended to the u option, indicating that the conversion should be applied to an item of type char, int, long, or short, respectively.</p>
	<p>x</p> <p>Display bytes as hexadecimal. By default, the od command transforms the corresponding number of bytes in the C-language type int. The x type string can be followed by an unsigned decimal integer that specifies the number of bytes to be transformed by each instance of the output type.</p> <p>An optional C, I, L, or S character can be appended to the x option, indicating that the conversion should be applied to an item of type char, int, long, or short, respectively.</p>
-v	Writes all input data. By default, output lines that are identical to the immediately preceding output lines are not printed, but are replaced with a line containing only an * (asterisk). When the -v flag is specified, all the lines are printed.

The flags for the second format are:

Table 6. Flags to display files using flags to format the output

Item	Description
-a	Displays bytes as characters and displays them with their ASCII names. If the -p flag is also given, bytes with even parity are underlined. The -P flag causes bytes with odd parity to be underlined. Otherwise, parity is ignored.
-b	Displays bytes as octal values.
-c	Displays bytes as ASCII characters. The following nongraphic characters appear as C-language escape sequences:
	<pre> \ Backslash \a Alert \b Backspace \f Form-feed \n New-line character \o Null \r Carriage return \t Tab \v Vertical tab </pre>
	Others appear as three-digit octal numbers.
-C	Displays extended characters as standard printable ASCII characters (using the appropriate character escape string) and displays multibyte characters in hexadecimal form.
-d	Displays 16-bit words as unsigned decimal values.
-D	Displays long words as unsigned decimal values.
-e	Displays long words as double-precision, floating point. (same as the -F flag)
-f	Displays long words as floating points.
-F	Displays long words as double-precision, floating point. (same as the -e flag)

Table 6. Flags to display files using flags to format the output (continued)

Item	Description
-h	Displays 16-bit words as unsigned hexadecimal.
-H	Displays long words as unsigned hexadecimal values.
-i	Displays 16-bit words as signed decimal.
-I	(Uppercase i) Displays long words as signed decimal values.
-l	(Lowercase L) Displays long words as signed decimal values.
-L	Displays long words as signed decimal values.
-o	Displays 16-bit words as unsigned octal.
-O	Displays long words as unsigned octal values.
-p	Indicates even parity on -a conversion.
-P	Indicates odd parity on -a conversion.
-s	Displays 16-bit words as signed decimal values.
-S[N]	Searches for strings of characters ending with a null byte. The <i>N</i> variable specifies the minimum length string to be recognized. If the <i>N</i> variable is omitted, the minimum length defaults to 3 characters.
-v	Writes all input data. By default, output lines that are identical to the immediately preceding output lines are not printed, but are replaced with a line containing only an * (asterisk). When the -v flag is specified, all the lines are printed.
-w [<i>N</i>]	Specifies the number of input bytes to be interpreted and displayed on each output line. If the -w flag is not specified, 16 bytes are read for each display line. If the -w flag is specified without the <i>N</i> variable, 32 bytes are read for each display line. The maximum input value is 4096 bytes. Input values greater than 4096 bytes will be reassigned the maximum value.
-x	Displays 16-bit words as hexadecimal values.
-X	Displays long words as unsigned hexadecimal values. (same as the -H flag)

Note: The flags **-I** (uppercase i), **-l** (lowercase L), and **-L** are identical.

Exit Status

This command returns the following exit values:

Table 7. Exit status

Item	Description
0	All input files were processed successfully.
>0	An error occurred.

Examples

1. To display a file in octal, a page at a time, enter:

```
od a.out | pg
```

This command displays the `a.out` file in octal format and pipes the output through the `pg` command.

2. To translate a file into several formats at once, enter:

```
od -t cx a.out > a.xcd
```

This command writes the contents of the `a.out` file, in hexadecimal format (**x**) and character format (**c**), into the `a.xcd` file.

3. To start displaying a file in the middle (using the first syntax format), enter:

```
od -t acx -j 100 a.out
```

This command displays the `a.out` file in named character (**a**), character (**c**), and hexadecimal (**x**) formats, starting from the 100th byte.

4. To start in the middle of a file (using the second syntax format), enter:

```
od -bcx a.out +100.
```

This displays the `a.out` file in octal-byte (**b**), character (**c**), and hexadecimal (**x**) formats, starting from the 100th byte. The `.` (period) after the offset makes it a decimal number. Without the period, the output would start from the 64th (100 octal) byte.

Files

Table 8. Files

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

odmadd Command

Purpose

Adds objects to created object classes.

Syntax

```
odmadd [ InputFile ... ]
```

Description

The `odmadd` command takes as input one or more *InputFile* files and adds objects to object classes with data found in the stanza files. Each *InputFile* file is an ASCII file containing the data that describes the objects to be added to object classes. If no file is specified, input is taken from stdin (standard input).

The classes to be added to are specified in the ASCII input file. The file is in the following general format:

```
class1name:
    descriptor1name = descriptor1value
    descriptor2name = descriptor2value
    descriptor3name = descriptor3value

class2name:
    descriptor4name = descriptor4value
.
.
.
```

The input file can contain the `\` (backslash), which is handled as it is in C language. String and method values in the input file must be enclosed in `" "` (double-quotation marks). A descriptor value can span more than one 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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

An ASCII input file used by the **odmadd** command looks like the following:

```
Fictional_Characters:
Story_Star      = "Cinderella"
Birthday        = "Once upon a time"
Age             = 19
Friends_of      = Cinderella
Enemies_of      = "Cinderella"

Friend_Table:
Friend_of       = "Cinderella"
Friend          = "Fairy godmother"

Friend_Table:
Friend_of       = "Cinderella"
Friend          = "Mice"

Enemy_Table:
Enemy_of        = "Cinderella"
Enemy           = "Wicked sisters"

Enemy_Table:
Enemy_of        = "Cinderella"
Enemy           = "Mean stepmother"
```

If the preceding file is named `NewObjects`, the following command adds the objects to existing object classes:

```
odmadd NewObjects
```

See [html](#)

odmchange Command

Purpose

Changes the contents of a selected object in the specified object class.

Syntax

```
odmchange -o ObjectClass [ -q Criteria] [ InputFile ]
```

Description

The **odmchange** command, given the object class to modify, the search criteria, and the new object (only for attributes that need to change), modifies all objects that satisfy the search criteria. The *InputFile* file has the same format as the *InputFile* file (the ASCII input file) for the **odmadd** command.

Flags

Table 9. Flags

Item	Description
-o <i>ObjectClass</i>	Specifies the object class to modify.

Table 9. Flags (continued)

Item	Description
<code>-q Criteria</code>	Specifies the search criteria used to select objects from the object class or classes.

odmcreate Command

Purpose

Produces the `.c` (source) and `.h` (include) files necessary for ODM application development and creates empty object classes.

Syntax

odmcreate [`-p`] [`-c` | `-h`] *ClassDescriptionFile*

Description

The **odmcreate** command is the ODM class compiler. The command takes as input an ASCII file that describes the objects a user wishes to use in a specific application. The **odmcreate** command can create empty object classes as part of its execution.

The output of the **odmcreate** command is a `.h` file (an include file) that contains the C language definitions for the object classes defined in the ASCII *ClassDescriptionFile* file. The resulting include file is used by the application for accessing objects stored in ODM. The **odmcreate** command also produces a `.c` file that must be compiled and bound in with the application. The `.c` file contains structures and definitions that are used internally by ODM at run time.

The *ClassDescriptionFile* parameter specifies an ASCII file that contains descriptions of one or more object classes. The general syntax for the *ClassDescriptionFile* parameter is as follows:

Table 10. Syntax for the *ClassDescriptionFile* parameter

Item	Description
file	: classes
classes	: class classes class
class	: head body tail
head	: struct ClassName {
tail	: }
body	: elements
elements	: elements elements element
element	: char <i>DescriptorName</i> [<i>DescriptorSize</i>] ; vchar <i>DescriptorName</i> [<i>DescriptorSize</i>] ; binary <i>DescriptorName</i> [<i>DescriptorSize</i>] ; short <i>DescriptorName</i> ; long <i>DescriptorName</i> ; long64 or int64 or ODM_LONG_LONG <i>DescriptorName</i> ; method <i>DescriptorName</i> ; link <i>StdClassName StdClassName ColName DescriptorName</i> ;

The default suffix for a *ClassDescriptionFile* file is **.cre**. If no suffix is specified on the **odmcreate** command, then a **.cre** suffix is appended. The file can have C language comments if run with the **-p** flag, and can include **#define** and **#include** lines that can be preprocessed if the **-p** flag is used to run the C language preprocessor on the file.

Note: ODM databases are 32-bit databases. The long type, when used in the class description file is a 32-bit data item. The long64 or int64 type, when used in the class description file is a 64-bit data item. The generated files will function the same for both 32- and 64-bit applications.

Flags

Table 11. Flags

Ite	Description
-----	-------------

- | | |
|-----------|--|
| -c | Creates empty object classes only; does not generate the C language .h and .c files. |
| -h | Generates the .c and .h files only; does not create empty classes. |
| -p | Runs the C language preprocessor on the <i>ClassDescriptionFile</i> file. |

Example

Assuming that a *ClassDescriptionFile* file named `FileName.cre` exists, the following command creates object classes:

```
odmcreate FileName.cre
```

Following is the `FileName.cre` source file and the resulting **.h** file:

```
/* This is an example odmcreate input file */
/* FileName.cre */

    class Class2 {
        char keys[32];
        method card;
        long cash;
    };
class TstObj {
    long a;
    char b[80];
    link Class2 Class2 card Class2Ln;
};

/* End of FileName.cre */

/* This is the generated header file FileName.h */
#include <odmi.h>

struct Class2 {
    long _id;           /* unique object id within object class */
    long _reserved;    /* reserved field */
    long _scratch;     /* extra field for application use */
    char keys[32];
    char card[256];    /* method */
    long cash;
};
#define Class2_Descs 3

extern struct Class Class2_CLASS[];
#define get_Class2_list (a,b,c,d,e) (struct Class2 * ) odm_get_list (a,b,c,d,e)

struct TstObj {
    long _id;           /* unique object id within object class */
    long _reserved;    /* reserved field */
    long _scratch;     /* extra field for application use */
    long a;
    char b[80];
    struct Class2 *Class2Ln; /* link */
    struct objlistinfo *Class2Ln_info; /* link */
};
```

```

    char Class2Ln_Lvalue[256];          /* link */
    };
#define TstObj_Descs 3

extern struct Class TstObj_CLASS[];
#define get_TstObj_list (a,b,c,d,e) (struct TstObj * ) odm_get_list (a,b,c,d,e)

/* End of generated header file FileName.h */

```

odmdelete Command

Purpose

Deletes selected objects from a specified object class.

Syntax

```
odmdelete -o ObjectClass [ -q Criteria ]
```

```
odmdelete -o ObjectClass [ -q Criteria ]
```

Description

The **odmdelete** command, given the object class to delete from and the search criteria, deletes all objects that meet those criteria.

Flags

Item	Description
-o <i>ObjectClass</i>	Specifies the object class to delete from.
-q <i>Criteria</i>	Specifies the search criteria used to select objects from the object class or classes.

odmdrop Command

Purpose

Removes an object class.

Syntax

```
odmdrop -o ClassName
```

Description

The **odmdrop** command removes an entire object class and all of its objects. No checking is done to see if other object classes are linked to this one.

Flags

Table 12. Flags

Item	Description
-o <i>ClassName</i>	Specifies the object class to remove.

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

Assuming that an object class named `MyObjectClass` exists, the following command removes the object class:

```
odmdrop -o MyObjectClass
```

odmget Command

Purpose

Retrieves objects from the specified object classes into an **odmadd** input file.

Syntax

```
odmget [ -q Criteria ] ObjectClass ...
```

Description

The **odmget** command takes as input a search criteria and a list of object classes, retrieves the selected objects from the specified object classes, and writes an ASCII **odmadd** input file to standard output.

Flags

Table 13. *Flags*

Item	Description
-q <i>Criteria</i>	Specifies the search criteria used to select objects from the object class or classes.

Examples

1. To display information about the Customized Device Drivers (CuDvDr) or the major (value1) and minor (value2) numbers of a device that has a logical name, `$logical_name`, enter the following command:

```
# odmget -q "resource='devno' and value3='$logical_name'" CuDvDr
```

The output might be similar to the following example:

```
CuDvDr:
  resource = "devno"
  value1 = "19"
  value2 = "8"
  value3 = "hdisk0"
```

2. To display information about the Customized Devices (CuDv) and the corresponding Predefined Devices (PdDv) of a device that has a logical name, `$logical_name`, enter the following command:

```
# odmget -q uniquetype=$(odmget -q name=$logical_name CuDv | tee /dev/tty | grep PdDvLn |
cut -d'"' -f2) PdDv
# logical_name=hdisk0
```

```
# odmget -q uniquetype=$(odmget -q name=$logical_name CuDv | tee /dev/tty | grep PdDvLn |
cut -d'"' -f2) PdDv
```

The output might be similar to the following example:

```
CuDv:
  name = "hdisk0"
  status = 1
  chgstatus = 2
  ddins = "scsidisk"
  location = "C5-T1-01"
  parent = "fscsi1"
  connwhere = "W_2"
  PdDvLn = "disk/fcp/mpioapdisk"

PdDv:
  type = "mpioapdisk"
  class = "disk"
  subclass = "fcp"
  prefix = "hdisk"
  devid = ""
  base = 1
  has_vpd = 1
  detectable = 1
  chgstatus = 0
  bus_ext = 0
  fru = 1
  led = 1574
  setno = 2
  msgno = 0
  catalog = "scdisk.cat"
  DvDr = "scsidisk"
  Define = "/usr/lib/methods/define"
  Configure = "/usr/lib/methods/cfgscsidisk"
  Change = "/usr/lib/methods/chgdisk"
  Unconfigure = "/usr/lib/methods/ucfgdevice"
  Undefine = "/usr/lib/methods/undefine"
  Start = ""
  Stop = ""
  inventory_only = 0
  uniquetype = "disk/fcp/mpioapdisk"
```

- To display information about the CuDv and Customized Attribute (CuAt) of a device that has a logical name, \$logical_name, enter the following command:

```
# logical_name=vio0
# odmget -q "name='$logical_name'" CuDv CuAt
```

The system displays the following output:

```
CuDv:
  name = "vio0"
  status = 1
  chgstatus = 2
  ddins = "vdev_busdd"
  location = ""
  parent = "sysplanar0"
  connwhere = "vdevice"
  PdDvLn = "bus/chrp/vdevice"

CuAt:
  name = "vio0"
  attribute = "bus_id"
  value = "0x90000340"
  type = "R"
  generic = "D"
  rep = "n"
  nls_index = 5
```

odmshow Command

Purpose

Displays an object class definition on the screen.

Syntax

```
odmshow ObjectClass
```

Description

The **odmshow** command takes as input an object class name (*ObjectClass*) and displays the class description on the screen. The class description is in the format taken as input to the **odmcreate** command.

Example

Assuming that an object class named `MyObjectClass` exists, the following command displays the description of `MyObjectClass` on the screen:

```
odmshow MyObjectClass
```

Also, see the **odmcreate** command or **ODM Example Code and Output** in *General Programming Concepts: Writing and Debugging Programs* for an example of the output listing.

on Command

Purpose

Executes commands on remote systems.

Syntax

```
/usr/bin/on [ -i ] [ -d ] [ -n ] Host Command [ Argument ... ]
```

Description

The **on** command executes commands on other systems in an environment that is similar to the one running the program. The **on** command passes the local environment variables to the remote machine, thus preserving the current working directory. When using the **on** command, both users must have the same user identification. Relative path names work only if they are within the current file system. Absolute path names can cause problems since commands are issued at one machine and executed on another.

The standard input is connected to the standard input of the remote command. The standard output and standard error from the remote command are sent to the corresponding files for the **on** command. The root user cannot execute the **on** command.



Attention: When the working directory is remotely mounted over the Network File System (NFS), the Ctrl-Z key sequence causes the window to hang.

Flags

Table 14. Flags

Item	Description
-d	Specifies debug mode. Displays status messages as work progresses.
-i	Specifies interactive mode. Uses remote echoing and special character processing. This option is needed for programs that expect to be talking to a terminal. All terminal modes and window size changes are increased.
-n	Specifies no input. This option causes the remote program to get an end-of-file (EOF) message when it reads from standard input. This flag is necessary when running commands in the background with job control.

Example

To execute the **ls -al** command on another machine and display the in-progress status messages on your terminal, enter:

```
on -d zorro ls -al
```

In this example, the **on** command executes the **ls** command on a workstation named **zorro**.

Files

Table 15. Files

Item	Description
/etc/inetd.conf	Defines how the inetd daemon handles Internet service requests.

openpts Command

Purpose

Allows enrolling and certifying a remote system.

Syntax

```
openpts [-i [-f] | [-v] | [-r] | [-D] | [-h] | [-V] | [-u] | [-l username] | [-p port] | [-c configfile] host
```

Description

The **openpts** command allows the system (the verifier) to connect to a remote **host** (the collector) to determine whether the collector has performed a trusted boot. A machine is considered to have performed trusted boot when the contents of the collector's trusted platform module (TPM) is interrogated for consistency against a reference set of measurements (reference manifest) maintained by the verifier. To acquire the set of reference measurements, the verifier must first enroll the collector by using the **-i** option. After enrollment, the verifier can attest the collector with the default **-v** option that compares the current values represented in the integrity report against the reference set. The success or failure of this operation is reported to you along with the reason of failure. Examples of operations that may cause a failed certification include booting from a different device, changing the boot flags, and modifying the boot image.

If updates are pending to the state of the collector (for example, an OS upgrade that affects the next boot operation) these updates are reported during an attestation. The user is prompted to accept or reject the new values. Updates can be automatically accepted by using the **-u** option. The attestation request uses

secure shell (SSH) as the communication mechanism between the collector and the verifier. The **openpts** command uses parameters such as **-l** for ssh command username and **-p** for port.

Flags

Table 16. Flags

Item	Description
-c <i>configfile</i>	Specifies the configuration file to use. The default is <code>~/.openpts/openpts.conf</code> .
-D	Displays the configuration settings of the target and all the options.
-h	Displays the command usage information.
-i [-f]	Enrolls a new collector partition or forces the enrollment of an existing collector.
-l <i>username</i>	Specifies the ssh command username.
-p <i>port</i>	Specifies the ssh command port number.
-r	Removes all information about a target system.
-u	Allow the command to accept updates to the manifest from the collector without prompting the yes option. The default is no.
-v (default)	Verifies a collector against its existing reference manifest.
-V	Displays the information in verbose mode. Multiple -V options increase the verbosity. This is used for debugging the data.

Files

Table 17. Files

Item	Description
<code>~/.openpts/</code>	This directory is the default location for all configuration and remote host information.
<code>~/.openpts/openpts.conf</code>	The configuration of the verifier.
<code>~/.openpts/uuid</code>	The UUID file of the verifier.
<code>~/.openpts/UUID/ir.xml</code>	The last integrity report received from the remote host.
<code>~/.openpts/UUID/newrm_uuid</code>	The UUID file of the new reference manifest (for example, for the next boot operation after a system update).
<code>~/.openpts/UUID/policy.conf</code>	The policy to verify the properties of a remote host.
<code>~/.openpts/UUID/rm_uuid</code>	The UUID file of the reference manifest.
<code>~/.openpts/UUID/UUID/rmN.xml</code>	The reference manifests of the remote host.
<code>~/.openpts/UUID/target.conf</code>	The configuration of the remote host.
<code>~/.openpts/UUID/vr.properties</code>	The platform properties of the remote host derived from the integrity report.

OS_install Command

Purpose

Performs network installation operations on OS_install objects.

Syntax

Traditional usage:

```
OS_install [ -K keyfile_path_name ] { -o Operation } [ -F ] [ -a attr=value... ] { ObjectName }
```

For system plan installations (System Plan mode):

```
OS_install [ -K keyfile_path_name ] -i sysplan { -x sysplan.xml } [ -d ] [ -F ]
```

For listing OS_install objects (List mode):

```
OS_install -l [ -v ] [ -t object_type | object_name ]
```

For managing network daemons:

```
OS_install -S | -U
```

Description

The OS_install command performs a network installation operation on an OS_install object. The type of operation is dependent on the type of object specified by the *ObjectName* parameter. The object pointed to by the *ObjectName* parameter can be one of four types: Client, OS_Resource, Remote_Resource or Control_Host. Command operations involve the creation and management of OS_install objects that enable network installation to install an operation system on a client system.

OS_install can also be run in System Plan mode by passing the **-i** sysplan flag instead of specifying an operation. This operation provides the ability to combine multiple OS_install operations into a single XML document.

The operations involving Remote_Resource objects require configuring an SSH key that is generated with the ssh-keygen command. The SSH key is required to run ssh commands on the local platform and remote resource server. On an HMC, the default name of the file *keyfile_path_name* containing the SSH key is /home/hscroot/ssh_keys. This file name can be overridden with the **-K** option. On other platforms, there is no default file name for the SSH key file. If the **-K** option is not specified on other platforms, the standard path names of SSH key files must be accessible to the OS_install command process.

The List mode of OS_install is used to list the current configuration of objects in the OS_install environment.

The HMC or IVM network daemons can be started and stopped with the **S** and **U** options, without modifying the OS_install objects.

Flags

Table 18. Flags

Item	Description
-a <i>attr=value</i>	Assigns the specified value to the specified attribute. <i>Operations</i> lists the required and optional attributes for a specific operation.
-d	Deletes all OS_install objects created during System Plan mode after all operations are completed.

Table 18. Flags (continued)

Item	Description
-F	Authorizes a reset of the existing remote server client system objects if required, during an OS_install allocate operation or system plan installation.
-i sysplan	Specifies System Plan mode.
-K <i>keyfile_path_name</i>	Specifies the absolute path name of the file where the SSH keys are generated.
-l	Lists all OS_install objects in the environment by default.
-o <i>Operation</i>	Specifies an operation to perform on an OS_install object.
-S	Starts the network daemons without modifying the OS_install objects.
-t <i>object_type</i> <i>object_name</i>	Narrows the list returned by the -l flag to only objects of type <i>object_type</i> or to the single OS_install object specified by <i>object_name</i> .
-U	Stops the network daemons without modifying the OS_install objects.
-v	Displays the list returned by the -l flag.
-x <i>sysplan.xml</i>	Specifies the XML file that contains the system plan.

Operations

Table 19. Operations			
Operation	Description	Required Attributes	Optional Attributes
define_client [-a attr=value...] {ClientObjectName}	Defines a new client object.	<p>ip_addr Client's IP address.</p> <p>mac_addr MAC address of the network interface of the client system.</p> <p>gateway IP gateway address of the client system.</p> <p>subnet_mask IP subnet mask of the client system.</p> <p>lpar LPAR name to install client (required attribute for the netboot operation).</p> <p>profile LPAR profile to use for the client (required attribute for the netboot operation).</p> <p>managed_system Name of the managed system that contains LPAR (required attribute for the netboot operation).</p> <p>ctrl_host Name of the Hardware Control Host object for this client (required attribute for the netboot operation).</p>	<p>adapter_speed Speed of the network adapter of the client system.</p> <p>adapter_duplex Duplex setting of the network adapter of the client system.</p> <p>disk_location Location of the disk to install client.</p> <p>vlan_tag Specifies the virtual logical area network (VLAN) tag to be used for tagging Ethernet frames during network installation for virtual network communication. Valid values are 0 - 4094.</p> <p>vlan_pri Specifies the virtual logical area network (VLAN) tag to be used for tagging Ethernet frames during network installation for virtual network communication. Valid values are 0 - 7.</p>

Table 19. Operations (continued)

Operation	Description	Required Attributes	Optional Attributes
define_resource [-a attr=value...] {ResourceObjectName}	Defines a new OS_Resource object.	type AIX or VIOS. version OS version. location Absolute path where OS_Resource resides. source Source of installation images.	configfile Install configuration file.
define_remote_resource [-a attr=value...] {ResourceObjectName}	Defines a new Remote_Resource object.	server Host name of the remote resource server. type AIX or Linux. remote_identifier Name of the resource or resource set on the remote resource server.	communication_method Supports ssh communication method.
define_ctrl_host [-a attr=value...] {ControlHostObjectName}	Defines a new Hardware Control_Host object.	communication_method Supports ssh communication method. hostname Host name of control host (the host name localhost can be specified if OS_install is run on the HMC control host). type hmc or ivm.	None.

Table 19. Operations (continued)

Operation	Description	Required Attributes	Optional Attributes
allocate [-F][-a attr=value...] {ClientObjectName}	Allocates an OS_Resource or Remote_Resource to a client object. Both objects must exist in the OS_install environment. An error occurs if the client object has an OS_Resource or Remote_Resource already allocated to it.	os_resource Existing OS_Resource or Remote_Resource object to allocate to the client object. remote_resource Existing Remote_Resource object to allocate to the client object. install_resource Existing OS_Resource or Remote_Resource object to allocate to the client object.	config_file Install configuration file (applies for an OS_Resource object).
netboot {ClientObjectName}	Instructs the hardware control host of the client object to initiate a network boot.	None.	None.
monitor_installation {ClientObjectName}	Monitors the installation status of the client object.	None.	None.
deallocate {ClientObjectName}	Deallocates the OS_Resource or Remote_Resource that was allocated to the client object by an allocate operation.	None.	None.
remove {ObjectName}	Removes the object from the OS_install environment.	None.	None.

Exit Status

Table 20. Exit status

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

Examples

- To define a client object, enter a command similar to the following:

```
OS_install -o define_client -a ip_addr=128.0.64.117 -a mac_addr=ab:cc:de:10:23:45 -a \
gateway=128.0.64.1 -a subnet_mask=255.255.255.0 -a ctrl_host=myhmc -a lpar=AIX1 -a \
profile=AIX1 -a managed_system=myMngSys myclient01
```

The preceding client object is a logical partition in a managed system.

2. To define an OS_Resource object, enter a command similar to the following:

```
OS_install -o define_resource -a location=/images/AIX/53ML3 -a type=AIX -a version=53ML3
my53resource
```

3. To define a Remote_Resource object (using the OS_install default SSH key file for HMC), enter a command similar to the following:

```
OS_install -o define_remote_resource -a server=MyNimServer -a type=AIX
-a remote_identifier=NimResGrp1 myRemoteResource
```

4. To define a Remote_Resource object (using a previously generated ssh-keygen key located in /home/hscroot/id_dsa file), enter the following:

```
OS_install -K /home/hscroot/id_dsa -o define_remote_resource -a server=MyNimServer -a
type=AIX -a
remote_identifier=NimResGrp1 myRemoteResource
```

5. To allocate the OS_Resource object defined in example 2 to a client object, enter a command similar to the following:

```
OS_install -o allocate -a os_resource=my53resource myclient01
```

or

```
OS_install -o allocate -a install_resource=my53resource myclient01
```

6. To allocate the Remote_Resource object defined in example 3 to a client object and authorize reset on an existing client, enter a command similar to the following:

```
OS_install -o allocate -F -a remote_resource=myRemoteResource myclient01
```

or

```
OS_install -o allocate -F -a install_resource=myRemoteResource myclient01
```

7. To deallocate the my53resource client object that was allocated in the example 5, enter:

```
OS_install -o deallocate myclient01
```

8. To define a Control_Host object to be specified for the ctrl_host attribute of a Client object, enter a command similar to the following:

```
OS_install -o define_ctrl_host -a type=hmc -a hostname=hmc_hostname -a
communication_method=ssh myhmc
```

Although the preceding example shares the same name of the ctrl_host attribute in the first example, the define_client operation allows an undefined Control_Host object to be specified for the ctrl_host attribute. In that case the controlling host of the Client object must be the HMC or IVM on which the netboot operation for the client is executed.

9. To execute a netboot operation, enter:

```
OS_install -o netboot myclient01
```

10. To view a myclient01 installation, enter:

```
OS_install -o monitor_installation myclient01
```

11. To remove the definition of the my53resource object, enter:

```
OS_install -o remove my53resource
```

12. To remove the definition of the myclient01 object, enter:

```
OS_install -o remove myclient01
```

If an `OS_Resource` object is specified, the `remove` operation removes OS images that exist in the file system directory specified by the `location` attribute of the object.

Configuring SSH

- Generate SSH Rivest-Shamir-Adleman (RSA) keys and place them in an accessible `ssh_keys` file in the HMC HOME directory, by entering the command:

```
ssh-keygen -t rsa -f /home/hscroot/ssh_keys
```

- On the remote resource server, append or copy the content of the `/home/hscroot/ssh_keys.pub` file that is generated by using the **ssh-keygen** command to the resource server's `.ssh/authorized_keys` file.
- If `OS_install` command is used to run a netboot operation on a target client of a remote HMC control host, append the content of the `/home/hscroot/ssh_keys.pub` file that is generated by using the **ssh-keygen** command to the remote HMC hscroot user's `.ssh/authorized_keys2` file, by entering the following command as a hscroot user on the remote HMC:

```
mkauthkeys -a '<content_of_ssh_keys.pub>'
```

Location

Table 21. Location

Item	Description
<code>/usr/sbin/OS_install</code>	
<code>/opt/osinstall</code>	Directory containing the <code>OS_install</code> Perl module files.

Files

Table 22. Files

Item	Description
<code>/var/osinstall</code>	Directory containing configuration files for the <code>OS_install</code> environment.
<code>/home/hscroot/ssh_keys</code>	Default file name for SSH keys on an HMC.

oslevel Command

Purpose

Reports the latest installed level (technology level, maintenance level and service pack) of the system.

Syntax

```
oslevel [ -l Level | -g Level | -q ] [-r | -s] [-f]
```

Description

The **oslevel** command reports the technology level and service pack of the operating system using a subset of all filesets installed on your system. These filesets include the Base Operating System (BOS), base devices, base printers, and X11.

The **oslevel** command also prints information about the technology level and service pack, including which filesets are not at a specified technology level or service pack.

Flags

Table 23. Flags

Item	Description
-l <i>Level</i>	Lists filesets that are earlier (less) than the technology level or service pack specified by the <i>Level</i> parameter.
-f	Forces the oslevel command to rebuild the cache for this operation.
-g <i>Level</i>	Lists filesets that are later (greater) than the technology level or service pack specified by the <i>Level</i> parameter.
-q	Lists names of known technology levels (when used with the -r flag) or service packs (when used with the -s flag) that can be specified using the -l or -g flag.
-r	Applies all flags to technology levels.
-s	Applies all flags to service packs. The service pack level returned is in the format 6100-00-01-0748, where 6100 refers to base level 6.1.0.0; 00 refers to technology level 0; 01 refers to service pack 1; and 0748 is the build sequence identifier, which is used to determine valid technology levels and service packs that can be applied to the current level. Attempts to apply a technology level or service pack with an earlier build sequence identifier will fail.

If no flags are specified, the base system software is entirely at or higher than the level that is listed in the output of the **oslevel** command.

Examples

1. To determine the base level of the system, type:

```
oslevel
```

The output will be similar to the following:

```
6.1.0.0
```

2. To determine the highest technology level reached for the current version of AIX on the system, type:

```
oslevel -r
```

3. To list all known technology levels on the system, type:

```
oslevel -rq
```

The levels returned can be used with the [**-r -l**] or [**-r -g**] flags, and will be similar to the following:

```
Known Recommended Maintenance Levels
-----
5300-02
5300-01
5300-00
```

4. To list which software is below AIX Version 5.3 technology level 1, type:

```
oslevel -r -l 5300-01
```

5. To list which software is at a level later than AIX Version 5.3 technology level 1, type:

```
oslevel -r -g 5300-01
```

6. To determine the highest service pack reached for the current technology level on the system, type:

```
oslevel -s
```

7. To list the known service packs on a system, type:

```
oslevel -sq
```

The levels returned can be used with the [**-s -l**] or [**-s -g**] flags, and will be similar to the following:

```
Known Service Packs
-----
6100-00-02-0750
6100-00-01-0748
6100-00-00-0000
```

8. To list which software is below AIX Version 6.1 technology level 0, service pack 1, type:

```
oslevel -s -l 6100-00-01-0748
```

9. To list which software is at a level later than AIX Version 6.1 technology level 0, service pack 1, type:

```
oslevel -s -g 6100-00-01-0748
```

Files

Table 24. Files

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

ospf_monitor Command

Purpose

Monitors the OSPF gateways.

Syntax

```
ospf_monitor mon_db_file
```

Description

The **ospf_monitor** command is used to query OSPF routers. The **ospf_monitor** command operates in interactive mode. It allows the user to query the various OSPF routers to provide detailed information on I/O statistics, error logs, link-state data bases, AS external data bases, the OSPF routing table, configured OSPF interfaces, and OSPF neighbors.

Specify the complete pathname of a database composed of records configuring destinations for **ospf_monitor** remote commands with *mon_db_file*. Each destination record is a single-line entry which lists the destination IP address, the destination hostname, and an OSPF authentication key (if authentication is activated by the destination). Since authentication keys may be present in the destination records, it is recommended that general access to this database be restricted.

Refer to RFC-1583 (OSPF Specification, version 2) for details about OSPF database and packet formats.

Commands

Upon entering interactive mode, **ospf_monitor** presents the '[#] dest command params >' prompt, at which you can enter any of **ospf_monitor**'s interactive commands. Interactive commands can be interrupted at any time with a keyboard interrupt.

Note: The command line length must be less than 200 characters.

Local Commands

Table 25. Local commands

Item	Description
?	Displays all local commands and their functions.
?R	Displays all remote commands and their functions.
d	Displays all configured destinations. This command displays <i>dest_index</i> , the IP address, and the hostname of all potential ospf_monitor command destinations configured in <i>mon_db_file</i> .
h	Displays the command history buffer showing the last 30 interactive commands.
x	Exits the ospf_monitor program.
@ remote_command	Sends <i>remote_command</i> to the same (previous) destination.
@dest_index remote_command	Sends <i>remote_command</i> to configured destination <i>dest_index</i> .
F filename	Sends all ospf_monitor output to <i>filename</i> .
S	Sends all ospf_monitor output to stdout.

Remote Commands

Table 26. Remote commands

Item	Description
a <i>area_id type ls_id adv_rtr</i>	<p>Displays link state advertisement. <i>Area_id</i> is the OSPF area for which the query is directed. <i>adv_rtr</i> is the router-id of the router which originated this link state advertisement. <i>Type</i> specifies the type of advertisement to request and should be specified as follows:</p> <ol style="list-style-type: none">1 Request the router links advertisements. They describe the collected states of the router's interfaces. For this type of request, the <i>ls_id</i> field should be set to the originating router's Router ID.2 Request the network links advertisements. They describe the set of routers attached to the network. For this type of request, the <i>ls_id</i> field should be set to the IP interface address of the network's Designated Router.3 Request the summary link advertisements describing routes to networks. They describe inter-area routes, and enable the condensing of routing information at area borders. For this type of request, the <i>ls_id</i> field should be set to the destination network's IP address.4 Request the summary link advertisements describing routes to AS boundary routers. They describe inter-area routes, and enable the condensing of routing information at area borders. For this type of request, the <i>ls_id</i> field should be set to the Router ID of the described AS boundary router.5 Request the AS external link advertisements. They describe routes to destinations external to the Autonomous System. For this type of request, the <i>ls_id</i> field should be set to the destination network's IP address.
c	Displays cumulative log. This log includes input/output statistics for monitor request, hello, data base description, link-state request, link-state update, and link-state ack packets. Area statistics are provided which describe the total number of routing neighbors and number of active OSPF interfaces. Routing table statistics are summarized and reported as the number of intra-area routes, inter-area routes, and AS external data base entries.
e	Displays cumulative errors. This log reports the various error conditions which can occur between OSPF routing neighbors and shows the number of occurrences for each.
h	Displays the next hop list. This is a list of valid next hops mostly derived from the SPF calculation.
l [<i>retrans</i>]	Displays the link-state database (except for ASE's). This table describes the routers and networks making up the AS. If <i>retrans</i> is non-zero, the retransmit list of neighbors held by this lsdb structure will be printed.

Table 26. Remote commands (continued)

Item	Description
A [<i>retrans</i>]	Displays the AS external data base entries. This table reports the advertising router, forwarding address, age, length, sequence number, type, and metric for each AS external route. If <i>retrans</i> is non-zero, the retransmit list of neighbors held by this lsdB structure will be printed.
o [<i>which</i>]	Displays the OSPF routing table. This table reports the AS border routes, area border routes, summary AS border routes, networks, summary networks and AS external networks currently managed via OSPF. If <i>which</i> is omitted, all of the fields are listed. If specified, the value of <i>which</i> (between 1 and 63) specifies that only certain tables should be displayed. The appropriate value is determined by adding up the values for the desired tables from the following list:
	1 Routes to AS border routers in this area.
	2 Routes to area border routers for this area.
	4 Summary routes to AS border routers in other areas.
	8 Routes to networks in this area.
	16 Summary routes to networks in other areas.
	32 AS routes to non-OSPF networks.
I	Displays all interfaces. This report shows all interfaces configured for OSPF. Information reported includes the area, interface IP address, interface type, interface state, cost, priority, and the IP address of the DR and BDR for the network.
N	Displays all OSPF routing neighbors. Information reported includes the area, local interface address, router ID, neighbor IP address, state, and mode.
V	Displays Gated version information.

p

The following AIX commands begin with the letter *p*.

pac Command

Purpose

Prepares printer/plotter accounting records.

Syntax

```
/usr/sbin/pac [ -c] [ -m] [ -pPrice] [ -PPrinter] [ -qFile] [ -r] [ -s] [ Name ... ]
```

Description

The **pac** command prepares printer/plotter accounting records for each user of the selected printer or for the users specified by the *Name* parameter. For printer choices, see the **-P** flag.

The unit of measure is the number of pages, with the exception of raster devices, for which feet of paper is measured. Output is expressed both as the number of units used and the charge in dollars. For information on the charge (price) per unit, see the **-p** flag.

The accounting file specified in the **/etc/qconfig** file and the file created to contain the summary information must grant read and write permissions to the root user or printq group. The **pac** command generates the summary file name by appending **_sum** to the path name specified by the `acctfile =` clause in the **/etc/qconfig** file. For example, if the **qconfig** file reads:

```
acctfile = /var/adm/1p0acct
```

The **pac** command expects the summary file to be named **/var/adm/1p0acct_sum**.

Flags

Item	Description
-c	Sorts the output by price instead of alphabetically by user.
-m	Groups all the printing charges for a user, regardless of the host machine.
-pPrice	Specifies the price, in dollars, charged per unit of output. By default, the system charges \$0.02 per unit.
-PPrinter	Specifies the printer for which accounting records are prepared. By default, the system selects the printer named by the PRINTER environment variable or the default value lp0 . Note: When the LPDEST environment variable is set, it takes precedence over the PRINTER environment variable, which has an identical function. Any destination options issued from the command line override both the LPDEST and PRINTER environment variables.
-qFile	Specifies the queue configuration file. The default value is the /etc/qconfig file.
-r	Reverses the sorting order, so that records are sorted alphabetically from z to a, or in descending order by price.
-s	Summarizes the accounting information in a summary file. This flag is needed for busy systems.

Examples

1. To produce printer/plotter accounting information for all users of the lp0 printer, enter:

```
/usr/sbin/pac
```

The command displays the number of printed pages and the charge, sorted by user. This example assumes that there is no **PRINTER** environment variable.

2. To collect printer/plotter accounting records in a summary file, enter:

```
/usr/sbin/pac -s
```

3. To produce printer/plotter accounting information for smith, jones, and greene from the lp12 printer, enter:

```
/usr/sbin/pac -Plp12 smith jones greene
```

Note: Do not place a space between a flag and its variable; for example, the **-pPrice**, **-PPrinter**, and **-qFile**.

Files

Item	Description
<code>/usr/sbin/pac</code>	Contains the pac command.
<code>/etc/qconfig</code>	Specifies the path to 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.

pack Command

Purpose

Compresses files.

Syntax

```
pack [ -f ] [ - ] File ...
```

Description

The **pack** command stores the file specified by the *File* parameter in a compressed form. The input file is replaced by a packed file with the same name and the suffix **.z** appended. If the invoking process has appropriate privileges, the packed file maintains the same access modes, access and modification dates, and owner as the original file. The input file name can contain no more than 253 bytes to allow space for the added **.z** suffix. If the **pack** command is successful, the original file is removed. Packed files can be restored to their original form using the **compress** command.

The exit value of the **pack** command is the number of files that it could not pack. The **pack** command does not pack under any of the following conditions:

- The file is already packed.
- The input file name has more than 253 bytes.

- The file has links.
- The file is a directory.
- The file cannot be opened.
- No storage blocks are saved by packing.
- A file called *File.z* already exists.
- The *.z* file cannot be created.
- An I/O error occurred during processing.

Flags

Item Description

- f** Forces packing of the file specified by the *File* parameter. This is useful for packing an entire directory, even if some of the files will not benefit.

Parameters

Item Description

File Specifies the file to be packed.

- Displays statistics about the file specified by the *File* parameter. The statistics are calculated from a Huffman minimum redundancy code tree built on a byte-by-byte basis. Additional occurrences of the - (minus sign) parameter on the command line toggles this function for the next specified file. See example 2.

Exit Status

This command returns the following exit values:

Item Description

- 0** Specifies that the file was successfully packed.
- >0** Specifies that an error occurred.

Examples

1. To compress the files named chap1 and chap2 and display the revised file names, enter:

```
pack chap1 chap2
```

The compressed versions are renamed chap1.z and chap2.z. The **pack** command displays the percent decrease in size for each file compressed.

2. To display statistics about the amount of compression done, enter:

```
pack -_chap1 -_chap2
```

This compresses the files named chap1 and chap2 and displays statistics about the file named chap1, but not about the file named chap2. The first - (minus sign) parameter turns on the statistic display, and the second - parameter turns it off.

Files

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

packf Command

Purpose

Compresses the contents of a folder into a file.

Syntax

```
packf [ +Folder ] [ Messages ] [ -file File ]
```

Description

The **packf** command compresses the messages in a folder into a specified file. By default, the **packf** command compresses messages from the current folder and places them in the **msgbox** file. If the file does not exist, the system prompts you for permission to create it. Each message in the file is separated with four Ctrl-A characters and a new-line character.

Note: You can use the **inc** command to unpack compressed messages.

Flags

Item	Description
-file <i>File</i>	Specifies the file in which to put compressed messages. The default is the ./msgbox file. If the file exists, the packf command appends the messages to the end of the file. Otherwise, the system prompts you for permission to create the file.
+Folder	Identifies the folder containing the messages you want to pack.
-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.

Item	Description
<i>Messages</i>	<p>Specifies what messages to pack. The <i>Messages</i> parameter can specify several messages, a range of messages, or a single message. If several messages are specified, the first message packed becomes the current message. Use the following references to specify messages:</p> <p>Number Number of the message. When specifying several messages, separate each number with a space. When specifying a range, separate the first and last numbers in the range with a hyphen.</p> <p>Sequence A group of messages specified by the user. Recognized values include:</p> <p>all All the messages in the folder. This is the default.</p> <p>cur or . (period) Current message.</p> <p>first First message in a folder.</p> <p>last Last message in a folder.</p> <p>next Message immediately after the current message.</p> <p>prev Message immediately before the current message.</p>

Profile Entries

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

Item	Description
Current-Folder:	Sets your default current folder.
Msg-Protect:	Sets the protection level for your new message files.
Path:	Specifies the user's MH 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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

- To pack all the messages in the current folder and place the resulting text in the **schedule** file, enter:

```
packf -file schedule
```

The system responds with a message similar to the following:

```
Create file "/home/mary/schedule"?
```

Enter y to create the file.

2. To pack the range of messages from 3 to 7 from the **test** folder into an existing **msgbox** file, enter:

```
packf +test 3-7
```

The system responds with the shell prompt when the command is complete.

3. To pack the current, first, and last message in the **inbox** folder into an existing **msgbox** file, enter:

```
packf cur first last
```

Files

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

pagdel Command

Purpose

Removes any existing PAG association within the current process' credentials.

Syntax

```
paginit [ -R module_name ] [ username ]
```

Description

The **pagdel** command will remove the PAG identifier from the current process' credentials structure. If the **-R** option is omitted, the registry attribute will be used as the **module_name**.

Flags

Item	Description
-R <i>module_name</i>	Specifies a load module found in /usr/lib/security/modules.cfg . The load_module will be asked to delete any PAG currently associated with the process.

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.

Auditing

USER_PagDelete

Example

To remove the PKI authentication certificate associated with the current process, type:

```
pagdel -R FPKI
```

pagesize Command

Purpose

Displays the system page size.

Syntax

```
pagesize [ -a ] [ -f ]
```

Description

The **pagesize** command prints the size, in bytes, of a page of memory, as returned by the **getpagesize** subroutine. Provided for system compatibility, this command is useful when constructing portable shell scripts.

If the **-a** flag is specified, the **pagesize** command prints all of the page size values (in bytes) supported on the system.

Flags

-a

Prints all of the page size values (in bytes) supported on the system.

-f

Prints the formatted page sizes with an alphabetical suffix rather than the page size in bytes (for example, 4K)

Example

1. To obtain the size system page, enter:

```
pagesize
```

The system returns the number of bytes, such as 4096.

2. To print the formatted page size, enter:

```
pagesize -f
```

The system returns the formatted page size (for example, 4K).

3. To print all of the supported page size with an alphabetical suffix, enter:

```
pagesize -af
```

The system returns all of the supported page sizes. For example:

```
4K
64K
16M
```

Files

Item

/usr/bin/pagesize

Description

Contains the **pagesize** command.

paginit Command

Purpose

Authenticate a user and create a PAG association.

Syntax

```
paginit [ -R module_name ] [ username ]
```

Description

The **paginit** command authenticates *username* (by default, the user issuing the command) and creates an association between the *username* and a kernel token called a Process Authentication Group entry (PAG). A new login shell is spawned by this command.

If the **-R** flag is not given, **paglist** queries the user's registry attribute and use that value for *module_name*.

To associate the *username* with an alternate Identification and Authentication (I&A) mechanism, the **-R** flag can be used to specify the I&A load module used to create the user. Load modules are defined in the **/usr/lib/security/methods.cfg** file.

Flags

Item	Description
-R <i>module_name</i>	Specifies the loadable I&A module used to authenticate the user.

Parameters

Item	Description
<i>username</i>	Specifies the user. This parameter defaults to the user issuing the command. Only the root user may override the default.

Security

Access Control: This command should be executable by all. It should be owned by **root** and should be **setuid**.

Auditing

USER_Paginit

Example

```
paginit -R FPKI
```

The user is authenticated using the registry FPKI, which is defined in the **/usr/lib/security/methods.cfg** file. A PAG is associated with the current process credentials.

paglist Command

Purpose

Lists authentication information associated with the current process.

Syntax

paglist [**-R** *module_name*]

Description

The **paglist** command queries the current process' credentials to display its authentication certificate.

If the **-R** option is not given, **paglist** will query the user's registry attribute and use that value for **module_name**.

Flags

Item	Description
-R <i>module_name</i>	Specifies that the load module <i>module_name</i> is to list its authentication certificate associated with the current process.

Security

Access Control: This command runs with the ID of the invoking user, without any elevated privileges. It should be owned by root, but executable by all.

Example

```
paglist -R FPKI
```

This example will list the PAG associated with the current process within the FPKI registry.

panel20 Command

Purpose

Diagnoses activity between an HIA and the 5080 Control Unit.

Syntax

panel20 [**HIA0** | **HIA1** | **HIA2**]

Description

Use the **panel20** command as a diagnostic tool to determine whether the Host Interface Adapter (HIA) is correctly installed and communicating with the 5088 Graphics Channel Control Unit (GCCU).

The **panel20** command displays a diagnostic screen with the following columns: Device Name, Channel Address, Link Address, Link Status, Poll Counter, SNRM Counter.

If the HIA is correctly installed and the host operating system is correctly configured to support 3270 devices on the 5088, the entries in the Set Normal Response Mode (SNRM Counter) column will be increasing. If the entries in SNRM Counter are not increasing, refer to problem determination procedures for the HIA and verify that the host operating system is correctly configured.

Examples

To start the **panel20** command, enter:

```
panel20
```

By default, the **panel20** command will monitor HIA0. To monitor HIA1 or HIA2, enter:

```
pane120 HIA1
```

OR

```
pane120 HIA2
```

passwd Command

Purpose

Changes a user's password.

Syntax

```
passwd [ -R load_module ] [ -f | -s -a ] [ User ]
```

Description

The **passwd** command sets and changes passwords for users. Use this command to change your own password or another user's password. You can also use the **passwd** command to change the full name (gecos) associated with your login name and the shell you use as an interface to the operating system.

Depending on how the user is defined, the user's password can exist locally or remotely. Local passwords exist in the **/etc/security/passwd** database. Remote passwords are stored in the database provided by the remote domain.

To change your own password, enter the **passwd** command. The **passwd** command prompts the nonroot user for the old password (if one exists) and then prompts for the new password twice. (The password is never displayed on the screen.) If the two entries of the new password do not match, the **passwd** command prompts for the new password again.

Note: The **passwd** command uses only the first eight characters of your password for local and NIS passwords. Only 7-bit characters are supported in passwords. For this reason, globalization code points are not allowed in passwords. You can set a password of up to 255 characters.

To change another user's password, enter the **passwd** command and the user's login name (the *User* parameter). Only the root user or a member of the security group is permitted to change the password for another user. The **passwd** command prompts you for the old password of the user as well as the new password. For local passwords, the **passwd** command does not prompt the root user for either the old user password or the root password. For remote passwords, by default the root user will be prompted to input the old password so the remote domain can make the decision to use the password or ignore it. To change this behavior, see the **rootrequiresopw** option in the **/usr/lib/security/methods.cfg** file. The **passwd** command does not enforce any password restrictions upon the root user.

The **/etc/passwd** file records your full name and the path name of the shell that you use. To change your recorded name, enter the **passwd -f** command. To change your login shell, enter the **passwd -s** command.

Construct locally-defined passwords according to the password restrictions in the **/etc/security/user** configuration file. This file contains the following restrictions:

Item	Description
dictionlist	Specifies the list of dictionary files checked when a password is changed.
histexpire	Specifies the number of weeks that a user cannot reuse a password.
histsize	Specifies the number of previous passwords that the user cannot reuse.
maxage	Specifies the maximum age of a password. A password must be changed after a specified amount of time measured in weeks.

Item	Description
maxexpired	Specifies the maximum number of weeks beyond the maxage value that a password can be changed by the user.
maxrepeats	Specifies the maximum number of times a single character can be used in a password.
minalpha	Specifies the minimum number of alphabetic characters.
minother	Specifies the minimum number of other characters.
minlen	Specifies the minimum number of characters. Note: This value is determined by either the minalpha value plus the minother value or the minlen value, whichever is greater.
mindiff	Specifies the minimum number of characters in the new password that are not in the old password. Note: This restriction does not consider position. If the new password is abcd and the old password is edcb, the number of different characters is 1.
minage	Specifies the minimum age at which a password can be changed. Passwords must be kept for a minimum period. This value is measured in weeks.
minloweralpha	Specifies the minimum number of lower case alphabetic characters.
minupperalpha	Specifies the minimum number of upper case alphabetic characters.
mindigit	Specifies the minimum number of digits.
minspecialchar	Specifies the minimum number of special characters.
pwdchecks	Specifies the list of external password restriction methods invoked when a password is changed.

If the root user adds the **NOCHECK** attribute to your flags entry in the **/etc/security/passwd** file, your password does not need to meet these restrictions. Also, the root user can assign new passwords to other users without following the password restrictions.

If the root user adds the **ADMIN** attribute to your flags entry or if the **password** field in the **/etc/passwd** file contains an * (asterisk), only the root user can change your password. The root user also has the exclusive privilege of changing your password if the **password** field in **/etc/passwd** contains an ! (exclamation point) and the **password** field in the **/etc/security/passwd** file contains an * (asterisk).

If the root user changes your password, the **ADMCHG** attribute is automatically added to your flags entry in the **/etc/security/passwd** file. In this case, you must change the password the next time you log in.

If the user's **registry** value in the **/etc/security/user** file is either DCE or NIS, the password change can only occur in the specified database.

The **passwd** command creates the user keystore, if the keystore does not exist and if the **efs_keystore_access** attribute value of the user is not **none**. The keystore is created with the Encrypted File System (EFS) attributes that are found in the **/etc/security/user** file. If the old password can open the keystore, it also changes the keystore password. That is to say, if the login and keystore passwords are same, then the **passwd** command changes both of the passwords. If the file system is an Encrypted File System (EFS), then the command performs as though the **-a** flag is specified. If you specify the **-a** flag, the result is that the EFS password is not synchronized with user login password after a password change. Therefore, the keystore is not be loaded automatically on next logins.

Flags

Item	Description
-a	Changes a user's password in all modules (compat, LDAP, NIS, and so on).

Item	Description
-f	Changes the user information accessed by the finger command. You can use this flag to provide your full name in the /etc/passwd file.
-s	Changes the login shell.
-R load_module	Specifies the loadable I&A module used to change a user's password.

Security

The **passwd** command is a PAM-enabled application with a service name of **passwd**. 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 **passwd** service in **/etc/pam.conf**. The **passwd** command requires **/etc/pam.conf** entries for the password module type. Listed below is a recommended configuration in **/etc/pam.conf** for the **passwd** service:

```
#
# AIX passwd configuration
#

passwd 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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To change your password, type:

```
passwd
```

The **passwd** command prompts you for your old password, if it exists and you are not the root user. After you enter the old password, the command prompts you twice for the new password.

2. To change your full name in the **/etc/passwd** file, type:

```
passwd -f
```

The **passwd** command displays the name stored for your user ID. For example, for login name **sam**, the **passwd** command could display this message:

```
sam's current gecos:
                  "Sam Smith"
Change (yes) or no)? >
```

If you type a **Y** for yes, the **passwd** command prompts you for the new name. The **passwd** command records the name you enter in the **/etc/passwd** file.

3. To use a different shell the next time you log in, type:

```
passwd -s
```

The **passwd** command lists the path names of the available shells and the shell you are currently using. The command also displays a prompt:

```
Change (yes) or (no)? >
```

If you type a **Y** for yes, the **passwd** command prompts you for the shell to use. The next time you log in, the system provides the shell that you specify here.

Files

Item	Description
<code>/usr/bin/passwd</code>	Contains the passwd command.
<code>/etc/passwd</code>	Contains user IDs, user names, home directories, login shell, and finger information.
<code>/etc/security/passwd</code>	Contains encrypted passwords and security information.

paste Command

Purpose

Joins the lines of different files.

Syntax

```
paste [ -s ] [ -d List ] File1 ...
```

Description

The **paste** command reads input from the files specified on the command line. The command reads from standard input if a - (minus sign) appears as a file name. The command concatenates the corresponding lines of the given input files and writes the resulting lines to standard output.

By default, the **paste** command treats each file as a column and joins them horizontally with a tab character (parallel merging). You can think of the **paste** command as the counterpart of the **cat** command (which concatenates files vertically, that is, one file after another).

With the **-s** flag, the **paste** command combines subsequent lines of the same input file (serial merging). These lines are joined with the tab character by default.

Notes:

1. The **paste** command supports up to 32767 input files (the **OPEN_MAX** constant).
2. The action of the **pr -t -m** command is similar to that of the **paste** command, but creates extra spaces, tabs, and lines for a nice page layout.
3. Input files should be text files, but may contain an unlimited number of line lengths.

Flags

Item	Description
-d <i>List</i>	Changes the delimiter that separates corresponding lines in the output with one or more characters specified in the <i>List</i> parameter (the default is a tab). If more than one character is in the <i>List</i> parameter, then they are repeated in order until the end of the output. In parallel merging, the lines from the last file always end with a new-line character instead of one from the <i>List</i> parameter. The following special characters can also be used in the <i>List</i> parameter: <ul style="list-style-type: none">\n New-line character\t Tab\\ Backslash\0 Empty string (not a null character)c An extended character You must put quotation marks around characters that have special meaning to the shell.
-s	Merges subsequent lines from the first file horizontally. With this flag, the paste command works through one entire file before starting on the next. When it finishes merging the lines in one file, it forces a new line and then merges the lines in the next input file, continuing in the same way through the remaining input files, one at a time. A tab separates the lines unless you use the -d flag. Regardless of the <i>List</i> parameter, the last character of the file is forced to be a new-line character.

Exit Status

This command returns the following exit values:

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

Examples

1. To paste several columns of data together, enter:

```
paste names places dates > npd
```

This creates a file named `npd` that contains the data from the `names` file in one column, the `places` file in another, and the `dates` file in a third. If the `names`, `places`, and `dates` file look like:

```
names      places      dates
rachel     New York   February 5
jerry      Austin     March 13
mark       Chicago    June 21
marsha     Boca Raton July 16
scott      Seattle    November 4
```

then the npd file contains:

```
rachel      New York      February 5
jerry       Austin        March 13
mark        Chicago       June 21
marsha      Boca Raton    July 16
scott       Seattle       November 4
```

A tab character separates the name, place, and date on each line. These columns do not always line up because the tab stops are set at every eighth column.

2. To separate the columns with a character other than a tab, enter:

```
paste -d"!@" names places dates > npd
```

This alternates ! and @ as the column separators. If the names, places, and dates files are the same as in example 1, then the npd file contains:

```
rachel!New York@February 5
jerry!Austin@March 13
mark!Chicago@June 21
marsha!Boca Raton@July 16
scott!Seattle@November 4
```

3. To display the standard input in multiple columns, enter:

```
ls | paste - - - -
```

This lists the current directory in four columns. Each - (minus) tells the **paste** command to create a column containing data read from the standard input. The first line is put in the first column, the second line in the second column, and so on.

This is equivalent to:

```
ls | paste -d"\t\t\t\n" -s -
```

This example fills the columns across the page with subsequent lines from the standard input. The -d"\t\t\t\n" defines the character to insert after each column: a tab character (\t) after the first three columns, and a new-line character (\n) after the fourth. Without the -d flag, the **paste -s -** command would display all of the input as one line with a tab character between each column.

Files

Item	Description
/usr/bin/paste	Contains the paste command.

patch Command

Purpose

Applies changes to files.

Syntax

```
patch [ -b [ -B Prefix ] ] [ -f ] [ -l ] [ -N ] [ -R ] [ -s ] [ -v ] [ -c | -e | -n | -u ] [ -d Directory ] [ -D Define ] [ -F Number ] [ -i PatchFile ] [ -o OutFile ] [ -p Number ] [ -r RejectFile ] [ -x Number ] [ File ]
```

Description

The **patch** command reads a source file's instructions on how to change a file, then applies the changes. The source file contains difference listings (or *diff* listings) produced by the **diff -c** or **-u** command, and one or more sets of **diff** command output, customarily called *hunks*.

The **patch** command skips any leading text in a patch file, applies the actual diff listing, and skips any trailing text. Thus, you could use as a patch file or message that includes a diff listing, and the **patch** command would still work. In such a case, if the entire diff listing is indented by a consistent amount, the **patch** command will also adjust for that spacing.

To change a line range within the original file, each hunk within a patch must be a separate diff listing. The line numbers for successive hunks within a patch must occur in ascending order.

File Name Determination

If no *File* parameter is specified, the **patch** command performs the following steps to obtain the name of the file to edit:

1. In the header of a context diff listing,
 - If the type of the diff is copied context, the file name is determined from lines beginning with ******* (three asterisks) or **---** (three dashes). A line beginning with ******* indicates the name of the file from which the patches were taken, while a line beginning with **---** indicates the name of the file to which the patches should be applied. The shortest name of an existing file is selected.
 - If the type of the diff is unified context, the file name is determined from lines beginning with **---** (three dashes) or **+++** (three pluses). A line beginning with **---** indicates the name of the file from which the patches were taken, while a line beginning with **+++** indicates the name of the file to which the patches should be applied. The shortest name of an existing file is selected.
2. If there is an **Index:** line in the leading text, the **patch** command tries to use the file name from that line.
3. A context diff header takes precedence over an **Index:** line.
4. If no file name can be determined from the leading text, the **patch** command prompts you for the name of the file to patch.
5. If the original file cannot be found, but a suitable SCCS or RCS file is available, the **patch** command attempts to get or check out the file.
6. If the leading text contains a **Prereq:** line, the **patch** command takes the first word from the prerequisites line (normally a version number) and checks the input file to see if that word can be found. If not, the **patch** command prompts you for confirmation before proceeding.

Patch Application

If the patch file contains more than one patch, the **patch** command tries to apply each diff listing as if it came from a separate patch file. In this case, the name of the file to patch is determined for each diff listing, and the header text before each diff listing is examined for information such as file name and revision level.

If you specify the **-c**, **-e**, **-n**, or **-u** flag, the **patch** command interprets information within each hunk as a copied context difference, an ed editor difference, a normal difference, or a unified context difference respectively. Otherwise, the **patch** command determines the type of difference based on the format of the information within the hunk.

The **patch** command searches for the place to apply each hunk by taking the first line number of the hunk and adding or subtracting any line offset caused by applying the previous hunk. If an exact match is not possible at this line location, the **patch** command scans both forward and backward for a set of lines matching the hunk's content exactly.

If no such place is found, and if the **patch** command is applying a context diff listing, the **patch** command can search for a less exact match. A *fuzz factor* specifies how many lines can be inexactly matched. If the fuzz factor is set to 1 or more, the **patch** command performs a second scan, this time ignoring the first and last line of context. If no match results, and the maximum fuzz factor is set to 2 or more, the **patch** command performs a third scan, this time ignoring the first two lines and the last two lines of the context. (The default maximum fuzz factor is 2.) If no match is found, the **patch** command places the hunk in a reject file. The reject file is created with the same name as the output file and the suffix **.rej**. This naming convention can be overridden by using the **-r** flag.

The rejected hunk is written in copied context diff listing form, regardless of the format of the patch file. If the input was a normal or ed editor style difference, the reject file may contain differences with zero lines of copied context format. The line numbers on the hunks in the reject file may be different from the line numbers in the patch file. This is because the reject file line numbers reflect the approximate locations for the failed hunks in the new file rather than the old one.

As each hunk is completed, the **patch** command tells you whether the hunk succeeded or failed. You are also informed of the new line number assumed for each hunk. If this is different from the line number specified in the diff listing, you are notified of the offset. The **patch** command also tells you if a fuzz factor was used to make the match.

Note: A single large offset may be an indication that a hunk was installed in the wrong place. Use of a fuzz factor may also indicate bad placement.

Preparing Patches for Other Users

Programmers preparing patches that will be shipped to other users should consider the following additional guidelines:

- If you try to apply the same patch twice, the **patch** command assumes the second application should be a reverse patch and prompts you for confirmation of this reversal. Therefore, avoid sending out reversed patches, since this makes users wonder whether they already applied the patch.
- It is recommended that you keep a **patchlevel.h** file that is updated with the latest patch level. The patch level can then be used as the first diff listing in the patch file you send out. If your patch includes a `PREREQ:` line, users cannot apply patches out of order without receiving a warning.
- Make sure you specify the file names correctly, either in a context diff listing header or with an `Index :` line. If you are patching something in a subdirectory, be sure to tell the patch user to specify a **-p** flag as needed.
- You can create a file by sending out a diff listing that compares a null file to the file you want to create. However, this only works if the file you want to create does not already exist in the target directory.
- While you may be able to put many diff listings into one file, it is advisable to group related patches into separate files.
- The **patch** command cannot tell if the line numbers are incorrect in an ed script, and can only detect bad line numbers in a normal diff listing when it finds a change or a delete command. A context diff listing using a fuzz factor of 3 may have the same line-number problem. Until a suitable interactive interface is added, use a context diff listing in such cases to check the changes for accuracy. Compilation without errors usually means that the patch worked, but it is not an infallible indicator.
- The results of the **patch** command are guaranteed only when the patch is applied to exactly the same version of the file from which the patch was generated.
- If the code has been duplicated, for example:

```
#ifdef
... NEWCODE
#else
... OLDCODE
# endif
```

the **patch** command is incapable of patching both versions. If the **patch** command succeeds, it may have patched the wrong version and return a successful exit status.

Flags

Item	Description
-b	Saves a copy of each modified file before the differences are applied. The copied original is filed with the same name and the suffix .orig . If a file by that name already exists, it is overwritten. If multiple patches are applied to the same file, only one copy is made of the original file at the time of the first patch. If the -o <i>OutFile</i> flag is also specified, the .orig file is not created. But if the specified out file already exists, <i>OutFile.orig</i> is created.
-B <i>Prefix</i>	Specifies a prefix to the backup file name. This flag only works in conjunction with the -b flag.
-c	Interprets the patch file as a copied context diff listing (the output of the diff -c or diff -C command). This flag cannot be used with the -e , -n , or -u flag.
-d <i>Directory</i>	Changes the current directory to the specified directory before processing.
-D <i>Define</i>	Marks changes with the following C preprocessor construct: <pre>#ifndef Define ... (NEWCODE) #else ... (OLDCODE) #endif /* Define */</pre> The <i>Define</i> variable is used as the differentiating symbol. This flag only works when the normal or context form of diff listing is used as a patch file.
-e	Interprets the patch file as an ed editor script. This flag cannot be used with the -c , -n , or -u flag.
-f	Suppresses queries to the user. To suppress commentary, use the -s flag.
-F <i>Number</i>	Sets the maximum fuzz factor. This flag applies to context diff listings only and causes the patch command to ignore the specified number of lines when determining where to install a hunk. If the -F flag is not specified, the default fuzz factor is 2. The factor may not be set to more than the number of lines of content in the context diff listing (ordinarily 3). Note: A larger fuzz factor increases the odds of a faulty patch.
-i <i>PatchFile</i>	Reads the patch information from the specified file, rather than from standard input.
-l	(lowercase L) Causes any sequence of blank characters in the diff listing script to match any sequence of blank characters in the input file. Other characters are matched exactly.
-n	Interprets the script as a normal diff listing. This flag cannot be used with the -c , -e , or -u flag.
-N	Ignores patches where the differences have already been applied to the file. By default, already-applied patches are rejected.
-o <i>OutFile</i>	Copies the files to be patched, applies the changes, then writes the modified version to the specified output file. Multiple patches for a single file are applied to the intermediate versions of the file created by any previous patches. Therefore, multiple patches result in multiple, concatenated versions of the output file.

Item	Description
-p <i>Number</i>	<p>Sets the path name strip count, which controls how path names found in the patch file are treated. This flag is useful if you keep your files in a directory different from the specified path. The strip count specifies how many slashes are stripped from the front of the path name. Any intervening directory names are also stripped. For example, assume a patch file specified <code>/u/leon/src/blurf1/blurf1.c</code>:</p> <ul style="list-style-type: none"> • -p 0 leaves the entire path name unmodified. • -p 1 removes the leading slash, leaving <code>u/leon/src/blurf1/blurf1.c</code>. • -p 4 removes four slashes and three directories, leaving <code>blurf1/blurf1.c</code>. <p>If the -p flag is not specified, only the base name (the final path name component) is used. This flag works only when the <i>File</i> parameter is not specified.</p>
-r <i>RejectFile</i>	<p>Overrides the default reject file name. The default reject file name is formed by appending the suffix .rej to the original file name.</p>
-R	<p>Reverses the sense of the patch script. For example, if the diff listing was created from new version to old version, using the -R flag causes the patch command to reverse each portion of the script before applying it. Rejected differences are saved in swapped format. The -R flag cannot be used with ed scripts, because there is too little information to reconstruct the reverse operation. If the -R flag is not specified, the patch command attempts to apply each portion in its reversed sense as well as in its normal sense, until a portion of the patch file is successfully applied. If the attempt is successful, the user is prompted to determine if the -R flag should be set.</p> <p style="padding-left: 40px;">Note: This method cannot detect a reversed patch if used with a normal diff listing where the first command is an append (that is, would have been a delete). Appends always succeed because a null context matches anywhere. Fortunately, most patches add or change lines rather than delete lines. Therefore most reversed normal diff listings begin with a delete, causing a failure and triggering heuristics.</p>
-s	<p>Patches silently unless an error occurs.</p>
-u	<p>Interprets the patch file as a unified context difference (the output of the diff command when you specify the -u or -U flag). You cannot specify this flag with the -c, -e, or -n flag.</p>
-v	<p>Prints the revision header and patch level. If the -v flag is used with other flags, the other flags are ignored.</p>
-x <i>Number</i>	<p>Sets internal debugging flags. This flag is only for patch command developers.</p>

Exit Status

The following exit values are returned:

Item	Description
0	Successful completion.
1	An error occurred and one or more lines are written to the rejected file.
>1	An error occurred.

Examples

1. To apply diff listings in the `difflisting` file to the `prog.c` file, enter:

```
patch -i difflisting prog.c
```

2. To save the original version of the `prog.c` file, enter:

```
patch -b -i difflisting prog.c
```

This applies changes to `prog.c` and saves the original contents of `prog.c` in the file `prog.c.orig`.

3. To patch the `prog.c` file without altering the original version, enter:

```
patch -i difflisting -o prog.new prog.c
```

This uses `prog.c` as a source file, but the changed version is written to a file named `prog.new`.

Files

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

pathchk Command

Purpose

Checks path names.

Syntax

```
pathchk [ -p ] [ -P ] pathname...
```

Description

The **pathchk** command checks that one or more path names are valid and portable. By default, the **pathchk** command checks each component of each path name specified by the *pathname* parameter based on the underlying file system. An error message is sent for each path name that meets the following criteria:

- The byte length of the full path name is longer than allowed by the system.
- The byte length of a component is longer than allowed by the system.
- Search permission is not allowed for a component.
- A character in any component is not valid in its containing directory.

It is not an error if one or more components of a path name do not exist. If a file that matches the path name specified by the *pathname* parameter can be created and it must not violate any of the above criteria.

More extensive portability checks are run when the `-p` flag is specified.

Flags

Item	Description
-p	Checks the path name based on POSIX portability standards. An error message is sent for each path name that meets the following criteria: <ul style="list-style-type: none">• The byte length of the full path name is longer than allowed by POSIX standards.• The byte length of a component is longer than allowed by POSIX standards.• A character in any component is not in the portable file name character set.
-P	Checks the <i>pathname</i> operand and returns an error message if the <i>pathname</i> operand meets the following criteria: <ul style="list-style-type: none">• The <i>pathname</i> operand contains a component whose first character is the hyphen character.• The <i>pathname</i> operands are empty.

Exit Status

This command returns the following exit values:

Item	Description
0	All <i>pathname</i> operands passed all of the checks.
>0	An error occurred.

Examples

1. To check the validity and portability of the `/home/bob/work/tempfiles` path name on your system, enter:

```
pathchk /home/bob/work/tempfiles
```

2. To check the validity and portability of the `/home/bob/temp` path name for POSIX standards, enter:

```
pathchk -p /home/bob/temp
```

Files

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

pax Command

Purpose

Extracts, writes, and lists members of archive files; copies files and directory hierarchies.

Syntax

To List Member Files of Archived Files

```
pax [ -c | -n ] [-d] [-U] [ -v ] [ -H | -L ] [-f Archive] [ -s ReplacementString... ] [-x Format] [-o Options] [ -Z ] [Pattern...]
```

To Extract Archive Files Using the -r Flag

pax -r [**-c** | **-n**] [**-d**] [**-i**] [**-k**] [**-U**] [**-u**] [**-v**] [**-H** | **-L**] [**-f** *Archive*] [**-o** *Options*] [**-p** *String...*] [**-s** *ReplacementString...*] [**-x** *Format*] [**-Z**] [*Pattern ...*]

To Write Archive Files Using the -w Flag

pax -w [**-d**] [**-i**] [**-t**] [**-U**] [**-u**] [**-v**] [**-X**] [**-H** | **-L**] [**-E**] [**-b** *Blocking*] [[**-a**] **-f** *Archive*] [**-o** *Options*] [**-s** *ReplacementString...*] [**-x** *Format*] [**-Z**] [*File...*]

To Copy Files Using the -r and -w Flags

pax -r -w [**-d**] [**-i**] [**-k**] [**-l**] [**-t**] [**-U**] [**-u**] [**-v**] [**-X**] [**-H** | **-L**] [**-p** *String...*] [**-o** *Options*] [**-s** *ReplacementString...*] [**-x** *Format*] [**-Z**] [*File ...*] *Directory*

Description

pax stands for portable archive interchange. The **pax** command extracts and writes member files of archive files; writes lists of the member files of archives; and copies directory hierarchies. The **-r** and **-w** flags specify the type of archive operation.

Note: **pax** actively sparse files that are being restored. If a file blocks an aligned and sized areas that are NULL populated, **pax** does not cause physical space for those file system blocks to be allocated. The file size in bytes remains the same, but the actual space that is taken within the file system is for the non-NULL areas.

Listing Member Files of Archived Files (List Mode)

When the **-r** flag or the **-w** flag is not specified, the **pax** command lists all the member files of the archive file that is read from standard input. If the *Pattern* parameter is specified, only the member files with path names that match the specified patterns are written to standard output. If a named file is a directory, the file hierarchy that is contained in the directory is also written. When the **-r** flag or the **-w** flag is not specified, the **-c**, **-d**, **-f**, **-n**, **-s**, and **-v** flags, and the *Pattern* parameter can be specified.

Extracting Archive Files Using the -r Flag (Read Mode)

When the **-r** flag is specified, but the **-w** flag is not, the **pax** command extracts all the member files of the archive files that are read from standard input. If the *Pattern* parameter is specified, only the member files with path names that match the specified patterns are written to standard output. If a named file is a directory, the file hierarchy that is contained in the directory is also extracted. The **-r** flag can be specified with the **-c**, **-d**, **-f**, **-i**, **-k**, **-n**, **-s**, **-u**, and **-v** flags, and with the *Pattern* parameter.

The access and modification times of the extracted files are the same as the archived files. The file modes of the extracted files are the same as when they were archived, unless they are affected by the user's default file creation mode (**umask**). The **S_ISUID** and **S_ISGID** bits of the extracted files are cleared.

If intermediate directories are necessary to extract an archive member, the **pax** command creates the directories with access permissions set as the bitwise inclusive OR of the values of the **S_IRWXU**, **S_IRWXG**, and **S_IRWXO** masks.

If the selected archive format supports the specification of linked files, it is an error if these files cannot be linked when the archive is extracted.

Writing Archive Files Using the -w Flag (Write Mode)

When the **-w** flag is specified and the **-r** flag is not, the **pax** command writes the contents of the files that are specified by the *File* parameter to standard output in an archive format. If no *File* parameter is specified, a list of files to copy, one per line, is read from the standard input. When the *File* parameter specifies a directory, all of the files that are contained in the directory are written. The **-w** flag can be specified with the **-a**, **-b**, **-d**, **-f**, **-i**, **-o**, **-s**, **-t**, **-u**, **-v**, **-x**, and **-X** flags and with *File* parameters.

Copying Files Using the -r and -w Flags (Copy Mode)

When both the **-r** and **-w** flags are specified, the **pax** command copies the files that are specified by the *File* parameters to the destination directory specified by the *Directory* parameter. If no files are specified, a list of files to copy, one per line, is read from the standard input. If a specified file is a directory, the file hierarchy that is contained in the directory is also copied. The **-r** and **-w** flags can be specified with the

-d, -i, -k, -l, -o, -p, -s, -t, -u, -v, and **-X** flags and with *File* parameters. The *Directory* parameter must be specified.

Copied files are the same as written to an archive file and are later extracted, except that there are hard links between the original and the copied files.

Modifying the Archive Algorithm Using the **-o** Flag

Use the **-o** flag to modify the archive algorithm according to keyword-value pairs. The keyword-value pairs must adhere to a correct archive format. A list of valid keywords and their behavior is given in the subsequent description of the **-o** flag.

Further notes

In read or copy modes, if intermediate directories are necessary to extract an archive member, the **pax** command does actions similar to the **mkdir()** subroutine with the intermediate directory used as the path argument and the value **S_IRWXU** as the mode argument.

If any specified pattern or file operands are not matched by at least one file or archive member, **pax** writes a diagnostic message to standard error for each one that did not match and exits with an error status.

In traversing directories, the **pax** command detects the infinite loops by entering a previously visited directory that is an ancestor of the last file visited. Upon detection of an infinite loop, the **pax** command writes a diagnostic message to standard error and terminates.

When **pax** command is in read mode or list mode, by using the **-x pax** archive format, a file name, link name, owner name, or any other field in an extended header record cannot be converted from the **pax** UTF8 code set format to the current code set and locale. The **pax** command writes a diagnostic message to standard error, processes the file as described for the **-o invalid=** option, and then processes the next file in the archive.

For AIX 5.3 the **pax** command ignores the extended attributes by default. The **-U** option informs **pax** to archive or restore extended attributes, which include ACLs. The **-pe** option preserves ACLs. When the **-pe** option is specified and if **pax** fails to preserve the ACLs, a diagnostic message is written to the standard error but the extracted file is not deleted. A nonzero exit code is returned. A new record type is required for extended attribute entries in the **pax** archive files.

Variables

Item	Description
<i>Directory</i>	Specifies the path of a destination directory when copying files.
<i>File</i>	Specifies the path of a file to be copied or archived. If no file matches the <i>File</i> parameter, the pax command detects the error, exits, and writes a diagnostic message.
<i>Pattern</i>	Specifies a pattern that matches one or more paths of archive members. A / (backslash) character is not recognized in the <i>Pattern</i> parameter and it prevents the subsequent character from having any special meaning. If no <i>Pattern</i> parameter is specified, all members are selected in the archive. If a <i>Pattern</i> parameter is specified, but no archive members are found that match the pattern that is specified, the pax command detects the error, exits, and writes a diagnostic message.

Flags

Item	Description
-a	Appends files to the end of an archive. Note: Streaming tape devices do not allow the append function.

Item	Description
-b <i>Blocking</i>	<p>Specifies the block size for output. The <i>Blocking</i> parameter specifies a positive decimal integer value that specifies the number of bytes per block. Application conforming to POSIX2 must not specify a blocksize value greater than 32256. Devices and archive formats might impose restrictions on blocking. Blocking is automatically determined on input. Default blocking when the archives are created depends on the archive format. (See the -x flag definition.)</p> <p>The <i>Blocking</i> parameter accepts one of the following values:</p> <p>Integer b Specifies that the block size, in bytes, be the value of the positive decimal integer that is specified by the <i>Integer</i> parameter that is multiplied by 512.</p> <p>Integer k Specifies that the block size, in bytes, be the value of the positive decimal integer that is specified by the <i>Integer</i> parameter that is multiplied by 1024.</p> <p>Integer m Specifies that the block size, in bytes, be the value of the positive decimal integer that is specified by the <i>Integer</i> parameter that is multiplied by 1024 x 1024.</p> <p>Integer+Integer Specifies that the block size, in bytes, be the sum of the positive decimal integers that are specified by the <i>Integer</i> parameters.</p>
-c	Matches all file or archive members except the files that are specified by the <i>Pattern</i> parameter.
-d	Causes directories being copied, archived, or extracted, to match the directory and not the contents of the directory.
-E	Avoids truncation of the long user and group names during addition of files to a new or existing archive.
-f <i>Archive</i>	Specifies the path of an archive file to be used instead of standard input (when the -w flag is not specified) or standard output (when the -w flag is specified but the -r flag is not). When specified with the -a flag option, any files that are written to the archive are appended to the end of the archive.
-H	If a symbolic link that refers to a directory is specified on the command line, pax archives the file hierarchy that is rooted in the directory that is referenced in the link, by using the name of the link as the name of the file hierarchy. By default, pax archives the symbolic link itself.
-i	Renames files or archives interactively. For each archive member that matches the <i>Pattern</i> parameter or file that matches a <i>File</i> parameter, a prompt is written to the display device that contains the name of a file or archive member. A line is then read from the display device. If this line is empty, the file or archive member is skipped. If this line consists of a single period, the file or archive member is processed with no modification to its name. Otherwise, its name is replaced with the contents of the line.
-k	Prevents the pax command from writing over existing files.
-l	Links files when copying files. Hard links are established between the source and destination file hierarchies whenever possible.

Item	Description
-L	If a symbolic link that refers to a directory is specified on the command line or encountered during the traversal of a file hierarchy, pax archives the file hierarchy that is rooted in the directory that is referenced in the link, by using the name of the link as the name of the file hierarchy. By default, pax archives the symbolic link itself.
-n	Selects the first archive member that matches each <i>Pattern</i> parameter. No more than one archive member is matched for each pattern.

Item**-o Options****Description**

Modifies the archiving algorithm according to the keyword-value pairs specified in the *Options* parameter. The keyword-value pairs must be in the following format:

keyword:=value,keyword:=value,...

Some keywords apply only to certain file formats, as indicated with each description. Use of keywords that are inapplicable to the file format being processed will be ignored by **pax**.

Keywords can be preceded with white space. The *value* field consists of zero or more characters; within *value*, any literal comma must be preceded with a backslash (\). A comma as the final character, or a comma that is followed by white space as the final character, in *Options* is ignored. Multiple **-o** options can be specified. If keywords given to these multiple **-o** options conflict, the keywords and values that appear later in the command-line sequences take precedence. The earlier values are ignored.

The following keyword-value pairs are supported for the indicated file formats:

datastream=pathname,datastr_size=size (Applicable to all file formats.)

The **datastream** keyword indicates that the incoming archive file is not in a file format; instead, it is a `DataStream` from the standard input device. Consequently, the data must be archived as a regular file in a format that is recognized by the **-x** flag. The file name of the `DataStream` must be specified in the *pathname* parameter and must include the identification of the person who invoked the command, the group identification, and the **umask** for the file mode.

Note: The **datastream** keyword does not have a default variable size. You must specify one.

The **datastr_size** keyword denotes the size of the `DataStream` input in bytes by using decimal digits. If the **pax** command reaches the end of file (EOF) character before it reads the *size* parameter, it pads the archive file with null values. The null values make the archive file the same size as specified by the *size* parameter. If the data in the archive file exceeds the size that is specified, the **pax** command truncates the archive file to the size specified by the *size* parameter. The **pax** command also stops taking input and closes the archive file.

Note: You can specify multiple instances of keyword pairs. If you assign different values to the same keyword, the **pax** command uses the last value that is assigned to the keyword to run the **-o** flag.

delete=pattern (Applicable only to the **-x pax** format.)

When used in write or copy mode, **pax** omits any keywords matching *pattern* from the extended header records that it produces. When used in read or list mode, **pax** ignores any keywords matching *pattern* in the extended header records. In all cases, matching is done using standard shell pattern-matching notation. For example, **-o delete=security.*** suppresses security-related information.

Item**Description**

-o *Options (Continued)*

exthdr.name=string (Applicable only to the **-x pax** format.)

This keyword allows user control over the name written into the **ustar** header blocks for the extended header records. The name is the contents of *string* after the following character substitutions have been made:

string includes:

Replaced by:

%d

The directory name of the file, equivalent to the result of the **dirname** utility on the transformed path name.

%f

The filename of the file, equivalent to the result of the **basename** utility on the transformed path name.

%p

The process ID of the **pax** process.

%%

A % character.

Any other % characters in *string* produce undefined results. The **pax** takes the default value of the name as **%d/PaxHeaders.%p/%f**, if the keyword-value pair is not specified in the **-o Options** list as **-o exthdr.name=string**.

globexthdr.name=string (Applicable only to the **-x pax** format.)

When used in write or copy mode with the appropriate options, **pax** creates global extended header records with **ustar** header blocks that will be treated as regular files by previous versions of **pax**. This keyword allows user control over the name that is written into the **ustar** header blocks for global extended header records. The name is the contents of *string* after the following character substitutions have been made:

string includes:

Replaced by:

%n

An integer that represents the sequence number of the global extended header record in the archive starting at 1

%%

A % character

Any other % characters in *string* produce undefined results. If this keyword-value pair is not specified in the **-o Options** list, the default value of the name is

\$TMPDIR/GlobalHead.%n

where **\$TMPDIR** is either the value of the **TMPDIR** environment variable or **/tmp** if **TMPDIR** is unset.

invalid=action (Applicable only to the **-x pax** format.)

This keyword allows user control over the action **pax** takes upon encountering values in an extended header record that:

- in read or copy mode, are invalid in the destination hierarchy, or
- in list mode, cannot be written in the code set and current locale.

Item**Description****-o Options (Continued)**

pax recognizes these invalid values:

- In read or copy mode, a file name or link name that contains character encodings invalid in the destination hierarchy. (For example, the name may contain embedded NULLs.)
- In read or copy mode, a file name or link name that is longer than the maximum allowed in the destination hierarchy (for either a path name component or the entire path name).
- In list mode, any character string value (file name, link name, username, and so on) that cannot be written in the code set and current locale.

These mutually exclusive values of the *action* argument are supported:

- **bypass**

In read or copy mode, **pax** bypasses the file, causing no change to the destination hierarchy. In list mode, **pax** writes all requested valid values for the file, but its method for writing invalid values is unspecified.

- **rename**

In read or copy mode, **pax** acts as if the **-i** flag is in effect for each file with invalid file name or link name values, allowing the user to provide a replacement name interactively. In list mode, **pax** behaves identically to the **bypass** action.

- **UTF8**

When used in read, copy, or list mode and a file name, link name, owner name, or any other field in an extended header record cannot be translated from the **pax UTF8** code set format to the current code set and locale, **pax** uses the actual UTF8 encoding for the name.

- **write**

In read or copy mode, **pax** writes the file, translating or truncating the name, regardless of whether this may overwrite an existing file with a valid name. In list mode, **pax** behaves identically to the **bypass** action.

If no **-o invalid=action** is specified, **pax** acts as if the **bypass** action is specified. Any overwriting of existing files that may be allowed by the **-o invalid=actions** is subject to permission (**-p**) and modification time (**-u**) restrictions, and is suppressed if the **-k** flag is also specified.

linkdata (Applicable only to the **-x pax** format.)

In write mode, the **pax** command writes the contents of a file to the archive, even when that file is a hard link to a file whose contents are written to the archive.

Item**Description****-o Options (Continued)****listopt=format** (Applicable to all file formats.)

This keyword specifies the output format of the table of contents that are produced when the **-v** option is specified in list mode. To avoid ambiguity, this keyword-value pair must be used as the only or final keyword-value pair following the **-o** flag; all characters in the remainder of the option-argument are considered part of the format string. If multiple **-o listopt=format** options are specified, the format strings are considered to be a single, concatenated string, evaluated in command-line order. See the **List-Mode Format Specifications** section for more information.

times (Applicable only to the **-x pax** format.)

When used in write or copy mode, **pax** includes atime, ctime, and mtime extended header records for each file.

Extended header keywords(Applicable only to the **-x pax** format.)

If the **-x pax** format is specified, the keywords and values that are defined in the list below can be used as parameters to the **-o** flag, in either of two modes:

keyword=value

When used in write or copy mode, these keyword-value pairs are written into the global extended header records of the new archive. When used in read or list mode, these keyword-value pairs act as if they were present in the global extended header records of the archive that is being read. In both cases, the given value is applied to all files that do not have a value that is assigned in their individual extended header records for the specified keyword.

keyword:=value

When used in write or copy mode, these keyword-value pairs are written into the extended header records of each file in the new archive. When used in read or list mode, these keyword-value pairs act as if they were present in the extended header records of each file in the archive that is being read. In both cases, the given value overrides any value for the specified keyword that is found in global or file-specific extended header records.

atime

The file access time for the following files, equivalent to the value of the st_atime member of the stat structure for a file.

charset

The name of the character set to encode the data in the following files. The entries in this table are defined to refer to known standards:

<u>value</u>	<u>Formal Standard</u>
"ISO-IR 646 1990"	ISO/IEC 646 IRV
"ISO-IR 8859 1 1987"	ISO 8859-1
"ISO-IR 8859 2 1987"	ISO 8859-2
"ISO-IR 10646 1993"	ISO/IEC 10646
"ISO-IR 10646 1993 UTF8"	ISO/IEC 10646, UTF8 encoding
"BINARY"	None

The encoding is included in an extended header for information only; when **pax** is used as described, it does not translate the file data into any other encoding. The **BINARY** entry indicates binary data that is not encoded.

comment

A series of characters used as a comment. All characters in the value field are ignored by **pax**.

ctime

The file creation time for the following file(s), equivalent to the value of the `st_ctime` member of the `stat` structure for a file.

gid

The group ID of the group that owns the file, expressed as a decimal number by using digits from ISO/IEC 646. This record overrides the `gid` field in the following header block(s). When used in write or copy mode, **pax** includes a `gid` extended header record for each file whose group ID is greater than 99,999,999.

gname

The group of the following file(s), formatted as a group name in the group database. This record overrides the `gid` and `gname` fields in the following header blocks, and any `gid` extended header record. When used in read, copy, or list mode, **pax** translates the name from the UTF8 encoding in the header record to the character set appropriate for the group database on the receiving system. If any of the UTF8 characters cannot be translated, and if the **-o invalid=UTF8** option is not specified, the results are undefined. When used in write or copy mode, **pax** includes a `gname` extended header record for each file whose group name cannot be represented entirely with the letters and digits of the portable character set.

> **hdrcharset** <

The name of the character set that is used to encode the value field of the `gname`, `linkpath`, `path`, and `uname` extended header records. The entries in the following table are defined to refer to the known standards. Additional names might be agreed between the originator and the recipient.

<u>value</u>	<u>Formal Standard</u>
ISO-IR106462000UTF-8	ISO/IEC 10646, UTF-8 encoding
BINARY	None

If the `hdrcharset` extended header record is not specified, the default character set (ISO/IEC 10646-1:2000 standard UTF-8 encoding) is used to encode all values in the extended header records.

The **BINARY** entry indicates that all the values that are recorded in the extended headers for affected files are unencoded binary data from the underlying system.

linkpath

The path name of a link that is created to another file, of any type, previously archived. This record overrides the `linkname` field in the following **ustar** header block(s).

The following **ustar** header block determines the type of link that is created, whether hard or symbolic. In the latter case, the `linkpath` value is the contents of the symbolic link. **pax** translates the name of the link (contents of the symbolic link) from the UTF8 encoding to the character set appropriate for the local file system.

When used in write or copy mode, **pax** includes a link path extended header record for each link whose path name cannot be represented entirely with the members of the portable character set other than `NULL`.

mtime

The file modification time of the following file(s), equivalent to the value of the `st_mtime` member of the `stat` structure for a file. This record overrides the `mtime` field in the following header block(s). The modification time is restored if the process has the appropriate privilege to do so.

path

The pathname of the following file(s). This record overrides the *name* and *prefix* fields in the following header block(s). **pax** translates the path name of the file from the UTF8 encoding to the character set appropriate for the local file system. When used in write or copy mode, **pax** includes a path extended header record for each file whose path name cannot be represented entirely with the members of the portable character set other than NULL.

realtime.any

The keywords that are prefixed by real time are reserved for future POSIX real-time standardization. **pax** recognizes but silently ignores them.

security.any

The keywords that are prefixed by security are reserved for future POSIX security standardization. **pax** recognizes but silently ignores them.

size

The size of the file in octets, expressed as a decimal number using digits from ISO/IEC 646. This record overrides the *size* field in the following header block(s). When used in write or copy mode, **pax** includes a size of extended header record for each file with a size value greater than 999,999,999,999.

uid

The user ID of the user that owns the file, expressed as a decimal number using digits from ISO/IEC 646.. This record overrides the *uid* field in the following header block(s). When used in write or copy mode, **pax** includes a uid extended header record for each file whose owner ID is greater than 99,999,999.

uname

The owner of the following file(s), formatted as a user name in the user database. This record overrides the *uid* and *uname* fields in the following header block(s), and any *uid* extended header record. When used in read, copy, or list mode, **pax** translates the name from the UTF8 encoding in the header record to the character set appropriate for the user database on the receiving system. If any of the UTF8 characters cannot be translated, and if the **-o invalid=UTF8** option is not specified, the results are undefined. When used in write or copy mode, **pax** includes a uname extended header record for each file whose user name cannot be represented entirely with the letters and digits of the portable character set.

If the *value* field is zero length, it deletes any header block field, previously entered extended header value, or global extended header value of the same name.

If a keyword in an extended header record (or in a **-o** option-argument) overrides or deletes a corresponding field in the **ustar** header block, **pax** ignores the contents of that header block field.

Extended header keyword precedence

(Applicable only to the **-x pax** format.)

This section describes the precedence in which the various header records and fields and command-line options are selected to apply to a file in the archive. When **pax** is used in read or list modes, it determines a file attribute in this sequence:

1. If **-o delete=keyword-prefix** is used, the affected attribute is determined from step (7) if applicable, or ignored otherwise.
2. If **-o keyword:=NULL** is used, the affected attribute is ignored.
3. If **-o keyword:=value** is used, the affected attribute is assigned the value.
4. If *value* exists in a file-specific extended header record, the affected attribute is assigned the value. When extended header records conflict, the last one given in the header takes precedence.
5. If **-o keyword=value** is used, the affected attribute is assigned the value.
6. If a value exists in a global extended header record, the affected attribute is assigned the value. When global extended header records conflict, the last one given in the global header takes precedence.
7. Otherwise, the attribute is determined from the **ustar** header block.

Item	Description
<p>-p <i>String</i></p>	<p>Specifies one or more file characteristics to be retained or discarded on extraction. The <i>String</i> parameter consists of the characters a, e, m, o, and p. Multiple characteristics can be concatenated within the same string and multiple -p flags can be specified. The specifications have the following meanings:</p> <p>a Does not retain file-access times.</p> <p>e Retains the user ID, group ID, file mode, access time, modification time, and ACLs.</p> <p>m Does not retain file-modification times.</p> <p>o Retains the user ID and the group ID.</p> <p>p Retains the file modes.</p> <p>If neither the -e nor the -o flag is specified, or the user ID and group ID are not preserved for any reason, the pax command does not set the S_ISUID and S_ISGID bits of the file mode. If the retention of any of these items fails, the pax command writes a diagnostic message to standard error. Failure to retain any of the items affects the exit status, but does not cause the extracted file to be deleted. If specification flags are duplicated or conflict with each other, the last flag that is specified takes precedence. For example, if -p eme is specified, file-modification times are retained.</p>
<p>-r</p> <p>-s <i>ReplacementString</i></p>	<p>Reads an archive file from the standard input.</p> <p>Modifies file or archive-member names that are specified by the <i>Pattern</i> or <i>File</i> parameters according to the substitution expression <i>ReplacementString</i>, by using the syntax of the ed command. The substitution expression has the following format:</p> <p>-s /old/new/[gp]</p> <p>where (as in the ed command), <i>old</i> is a basic regular expression and <i>new</i> can contain an & (ampersand), \n (n is a digit) back references, or subexpression matching. The <i>old</i> string can also contain new-line characters.</p> <p>Any non-null character can be used as a delimiter (the / (backslash) is the delimiter in the example). Multiple -s flag expressions can be specified; the expressions are applied in the order specified, terminating with the first successful substitution. The optional trailing g character performs as in the ed command. The optional trailing p character causes successful substitutions to be written to standard error. File or archive-member names that substitute to the empty string are ignored when reading and writing archives.</p>
<p>-t</p>	<p>Causes the access times of input files to be the same as they were before being read by the pax command.</p>

Item	Description
-U	<p>Performs archival and extraction of ACL and Extended Attributes. Attributes include Access control list (ACL) also. If the ACL type is not supported on the <i>Target</i> filesystem then it is converted to the ACL type supported by the <i>Target</i> filesystem. If the EA is not supported on the filesystem then it is not copied. When listing members of the archive this option will list the names of any named extended attributes and the type of any ACLs associated with each file that are part of the archive image.</p>
-u	<p> Ignores files that are older than a preexisting file or archive member with the same name.</p> <ul style="list-style-type: none"> • When extracting files, an archive member with the same name as a file in the file system is extracted if the archive member is newer than the file. • When writing files to an archive file, an archive member with the same name as a file in the file system is superseded if the file is newer than the archive member. If the -a flag is specified this is accomplished by appending to the archive. Otherwise it is unspecified if this is accomplished by actual replacement in the archive or by appending to the archive. • When copying files to a destination directory, the file in the destination hierarchy is replaced by the file in the source hierarchy or by a link to the file in the source hierarchy if the file in the source hierarchy is newer.
-v	<p>Writes information about the process. If neither the -r or -w flags are specified, the -v flag produces a verbose table of contents; otherwise, archive member path names are written to standard error.</p>
-w	<p>Writes files to the standard output in the specified archive format.</p>

Item	Description
-x <i>Format</i>	<p>Specifies the output archive format with the default format being ustar. The pax command recognizes the following formats:</p> <p>pax The pax interchange format. The default blocking value for this format for character-special archive files is 10240. Blocking values of 512 - 32256 in increments of 512 are supported.</p> <p>cpio Extended cpio interchange format. The default blocking value for this format for character-special archive files is 5120. Blocking values of 512 - 32256 in increments of 512 are supported.</p> <p>ustar Extended tar interchange format. The default blocking value for this format for character-special archive files is 10240. Blocking values of 512 - 32256 in increments of 512 are supported.</p> <ul style="list-style-type: none"> • Filename: The pax command supports the length of the path and the file name until the PATH_MAX limit that is defined by the system is reached. If the length of the path and the file name input exceeds the PATH_MAX limit, then the values are not archived. • gid or uid: The pax command supports the values of gid and uid until the UINTMAX limit is reached. Values greater than the UINTMAX limit are truncated. <p>If you attempt to append an archive file with a format that is different from the existing archive format causes the pax command to exit immediately with a nonzero exit status.</p> <p>In copy mode, if no -x format is specified, pax behaves as if -x pax were specified.</p>
-X	When traversing the file hierarchy specified by a pathname, the pax command does not descend into directories that have a different device ID.
-Z	<p>Archives the Encrypted File System (EFS) information of encrypted files or directories. The EFS information is extracted by default. When members of the archive are listed, 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 for other files.</p> <p>Note: Archives created with the -Z flag can be restored only on AIX 6.1 or later releases.</p>

Flag Interaction and Processing Order

The flags that operate on the names of files or archive members (**-c**, **-i**, **-n**, **-s**, **-u**, and **-v**) interact as follows:

- When extracting files, archive members are selected according to the user-specified *pattern* parameters as modified by the **-c**, **-n**, and **-u** flags. Then, any **-s**, and **-i** flags modify, in that order, the names of the selected files. The **-v** flag writes the names resulting from these modifications.
- When writing files to an archive file, or when copying files, the files are selected according to the user-specified pathnames as modified by the **-n** (this option is not valid for Copy Mode) and **-u** flags. Then, any **-s**, and **-i** flags modify, in that order, the names resulting from these modifications. The **-v** flag writes the names resulting from the modification.

- If both the **-u** and **-n** flags are specified, the **pax** command does not consider a file selected unless it is newer than the file to which it is compared.

List Mode Format Specifications

In list mode with the **-o listopt=format** option, the format argument is applied for each selected file. **pax** appends a newline character to the **listopt** output for each selected file. The format argument is used as the format string described in **printf()**, with the following exceptions:

1. The sequence *keyword* can occur before a format conversion specifier. The conversion argument is defined by the value of *keyword*. The following keywords are supported:
 - Any of the field name entries for **ustar** and **cpio** header blocks.
 - Any keyword defined for the extended header or provided as an extension within the extended header.

For example, the sequence **%(charset)s** is the string value of the name of the character set in the extended header.

The result of the keyword conversion argument is the value from the applicable header field or extended header, without any trailing NULLs.

All keyword-values used as conversion arguments are translated from the UTF8 encoding to the character set appropriate for the local file system, user database, etc., as applicable.

2. An additional conversion character, **T**, specifies time formats. The **T** conversion character can be preceded by the sequence *keyword=subformat*, where *subformat* is a date format allowed by the **date** command. The default keyword is **mtime** and the default subformat is: **%b %e %H:%M %Y**.
3. An additional conversion character, **M**, specifies the file mode string as displayed by the **ls -l** command. If *keyword* is omitted, the **mode** keyword is used. For example, **%1M** writes the single character corresponding to the *entry type* field of the **ls -l** command.
4. An additional conversion character, **D**, specifies the device for block or special files, if applicable. If not applicable and *keyword* is specified, then this conversion is equivalent to **%keyword u**. If not applicable and *keyword* is omitted, this conversion is equivalent to **<space>**.
5. An additional conversion character, **F**, specifies a pathname. The **F** conversion character can be preceded by a sequence of comma-separated keywords:

keyword,keyword...

The values for all the non-null keywords are concatenated together, each separated by a **/**. The default is *path* if the keyword *path* is defined; otherwise, the default is *prefix,name*.

6. An additional conversion character, **L**, specifies a symbolic link expansion. If the current file is a symbolic link, then **%L** expands to:

"%s -> %s", *value_of_keyword*, *contents_of_link*

Otherwise, the **%L** conversion character is equivalent to **%F**.

Exit Status

This command returns the following exit values:

Ite Description

m

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 copy the `olddir` directory hierarchy to `newdir`, enter:

```
mkdir newdir
```

```
pax -rw olddir newdir
```

2. To copy the contents of the current directory to the tape drive, enter:

```
pax -wf /dev/rmt0
```

3. To archive the file `xxx` as `XXX` and display the successful substitution, enter one of the following commands:

- ```
pax -wvf/dev/rfd0 -s /xxx/XXX/p xxx
```

- ```
pax -wvf/dev/rfd0 -s/x/X/gp xxx
```

4. To read a file from a standard input and dump it to a datastream file with a specified size, enter:

```
dd if=/dev/hd6 bs=36b count=480 | pax -wf /dev/rfd0 -o  
datastream=_filename_,datastr_size=_size_
```

5. To list the files in an archive `pax.ar` in a specified format, enter:

```
pax -v -o listopt="start %F end" -f pax.ar
```

6. To create an archive `pax.ar` in pax format, enter :

```
pax -wf pax.ar -x pax file1
```

7. To extract a file from an archive `pax.ar` in pax format with a new path, enter :

```
pax -rvf pax.ar -x pax -o path=newfilename
```

8. To copy the contents of a symbolic link from source to destination, enter:

```
pax -rWL srclink destdir
```

9. To extract files from the archive with group name as `bin`, enter:

```
pax -rvf pax.ar -x pax -o gname=bin
```

10. To ignore the path name from the archive in pax format during extraction, enter:

```
pax -rvf pax.ar -o delete=path
```

11. To avoid the truncation of long user and group names while creating the archive, enter:

```
pax -wEf file.pax file
```

12. To copy the `olddir` directory hierarchy to `newdir` with ACL and EA associated with the files, enter:

```
mkdir newdir
```

```
pax -rUw olddir newdir
```

Files

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

pcat Command

Purpose

Unpacks files and writes them to standard output.

Syntax

pcat *File* ...

Description

The **pcat** command reads the files designated by the *File* parameter, unpacks them, and writes them to standard output. Whether or not the specified file ends in the **.z** characters, the **pcat** command assumes that the file is packed and unpacks it.

The exit value of the **pcat** command is the number of files it 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.

Exit Status

This command returns the following exit values:

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

Examples

1. To display compressed files, enter:

```
pcat chap1.z chap2 | pg
```

This command sequence displays the compressed files `chap1.z` and `chap2.z` on the screen in expanded form, a page at a time (`| pg`). Note that the **pcat** command accepts files with and without the **.z** characters.

2. To use a compressed file without expanding the copy stored on disk, enter:

```
pcat chap1.z | grep 'Greece'
```

This command sequence prevents the **pcat** command from displaying the contents of `chap1.z` in its expanded form and pipes it to the **grep** command.

File

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

pdelay Command

Purpose

Enables or reports the availability of delayed login ports.

Syntax

```
pdelay [ -a ] [ Device ]
```

Description

The **pdelay** command enables delayed ports. Delayed ports are enabled like shared ports, except that the login herald is not displayed until you type one or more characters (usually carriage returns). If a port is directly connected to a remote system or connected to an intelligent modem, it is enabled as a delayed port to prevent the **getty** command from talking to a **getty** on the remote side or to the modem on a local connection. This action conserves system resources and is equivalent to **pdelay enabled=delay**. If you do not specify a *Device* parameter, the **pdelay** command reports the names of the currently enabled ports.

Use the *Device* parameter to specify the ports to be enabled. Permitted values include:

- Full device name, such as the **/dev/tty1** device
- Simple device name, such as the **tty1** device
- A number (for example, 1 to indicate the **/dev/tty1** device)

Note: You must have root user authority to run this command.

Flags

Item	Description
m	
-a	Enables all ports as delayed.

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 display the names of the delayed ports that are currently enabled, enter:

```
pdelay
```

Files

Item	Description
<code>/etc/locks</code>	Contains lock files for the pshare and pdelay commands.

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

pdisable Command

Purpose

Disables login ports.

Syntax

```
pdisable [ -a ] [ Device ]
```

Description

The **pdisable** command disables a specific port, even if a user is logged in at that port. The system disables a port by updating an entry in the `/etc/inittab` file and then sending a signal to the **init** process. When the **init** process receives the signal and reads the updated status entry, it takes the appropriate action.

Use the *Device* parameter to specify the ports to be disabled. Permitted values include:

- A full device name, such as the `/dev/tty1` device
- A simple device name, such as the `tty1` device
- A number (for example, 1 to indicate the `/dev/tty1` device).

If you do not specify a *Device* parameter, the **pdisable** command reports the names of currently disabled ports in its set.

Note: You must have root user authority to run this command.

Flag

Item	Description
------	-------------

<code>-a</code>	Disables all ports that are currently enabled.
-----------------	--

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 display the names of all ports currently disabled, enter:

```
pdisable
```

2. To disable all ports that are enabled, even if users are logged in, enter:

```
pdisable -a
```

3. To disable the workstation attached to the `/dev/tty8` port, enter:

```
pdisable tty8
```

Files

Item	Description
<code>/etc/locks</code>	Contains lock files for the pshare and delay commands.
<code>/usr/sbin/pdisable</code>	Contains the pdisable command.

pdlink Command

Purpose

Links files in partitioned sub directories.

Syntax

pdlink *dirname filename ...*

Description

The **pdlink** command allows you to make a file that exists under a partitioned subdirectory accessible to the processes running at different SLs. The file corresponds to the sensitivity label (SL) of the invoking process. The directory name that you specify using the *dirname* parameter must be a partitioned directory, and the file name that you specify using the *filename* parameter must be a file name (not a path name) under that named directory. You can specify multiple file names.

The **pdlink** command creates a hard link to the file specified, with the following qualifications:

- The link is only created in the partitioned subdirectories.
- Each partitioned subdirectory must exist at the time the **pdlink** command is running.
- The link is only created in partitioned subdirectories that have an SL that is higher than the minimum SL of the file specified by the *filename* parameter.

Security

Only authorized users can run the **pdlink** command.

Item	Description
aix.mls.pdir.link	Required to create links in partitioned sub directories with this command.

Exit Status

The **pdlink** command returns the following exit values:

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

Examples

1. To create a link of the **sample.c** file, present in the partitioned directory called **partdir**, enter:

```
pdlink partdir sample.c
```

Files

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

pdmkdir Command

Purpose

Creates partitioned directories.

Syntax

```
pdmkdir [ -m Mode ] [ -u Owner ] [ -g Group ] dirname ...
```

Description

The **pdmkdir** command creates partitioned directories that you specify using the *dirname* parameter. Normal users can create partitioned directories if the Discretionary Access Control (DAC), the Mandatory Access Control (MAC) and the Mandatory Integrity Control (MIC) permissions allow the user to create the new directory. Users with the **aix.mls.pdir.mkdir** authorization can override the DAC, MAC and MIC permissions.

Flags

Item	Description
-g <i>Group</i>	Sets the group of the newly-created directories. You can specify either a group name or group ID. Users with the aix.mls.pdir.mkdir authorization can change the group of the directory to a group that they are not members of.
-m <i>Mode</i>	Sets the permission bits for the newly created directories to the value that is specified by the <i>Mode</i> variable. Specify the <i>Mode</i> variable as a numeric value.
-u <i>Owner</i>	Sets the owner of the newly created directories. You can specify either the owner name or owner ID. Users with the aix.mls.pdir.mkdir authorization can change the owner of the directory.

Note: The *Mode*, *Owner* or *Group* variable that is set is applied to the partitioned directory and the partitioned subdirectory created based on the processes Sensitivity Level (SL) which ran the command. If another process with a different SL accesses the partitioned directory, the partitioned subdirectory that is created cannot be governed by these flags.

Security

All users can run the **pdmkdir** command. To successfully perform specific functions, users need the following authorization:

Item	Description
aix.mls.pdir.mkdir	Required to change the owner or group using the -u or -g flag. This authorization is also required to create directories in a path that ignores the DAC, MAC and MIC permissions of the parent directory.

Exit Status

The **pdmkdir** command returns the following exit values:

Item	Description
0	The command ran successfully and made all requested changes.
>0	An error occurred.

Examples

1. To create a partitioned directory, enter:

```
pdmkdir partdir
```

2. To create a partitioned directory with the permission "755", user "joe", group "staff", enter:

```
pdmkdir -m 755 -u joe -g staff partdir
```

Files

Item	Description
/usr/sbin/ pdmkdir	Contains the pdmkdir command.

pdmode Command

Purpose

Invokes a command in the virtual or real partitioned, directory-access mode.

Syntax

```
pdmode [ [ -r ] command [ arg ... ] ]
```

Description

The **pdmode** command allows you to invoke a command that you specify using the *command* parameter in the virtual or real partitioned directory access mode. When invoked without any argument, the **pdmode** command returns the partitioned directory access mode of the process which invoked this command.

If you run the **pdmode** command followed by the *command* parameter without any flag, the command is run in the virtual mode. A user can run a command in the real partitioned directory access mode by using the **-r** flag.

Flags

Item	Description
-r <i>command</i> [<i>arg</i> ...]	Sets the new process's partitioned directory access mode to the real mode. In this mode, partitioned directories are not transparent, and you must be aware of partitioned directories to navigate the subtree at a partitioned directory. To successfully run the command with this option, users need the aix.mls.pdir.mode authorization.

Security

All users can run the **pdmode** command. To successfully perform specific functions, you need the following authorization:

Item	Description
aix.mls.pdir.mode	Required to use the pdmode command with the -r flag.

Exit Status

The **pdmode** command returns the following exit values:

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

Examples

1. To get the partitioned directory access mode, enter:

```
pdmode
```

2. To run the **ls** command in the virtual mode, enter:

```
pdmode ls -l
```

3. To run the **ls** command in the real mode, enter:

```
pdmode -r ls -l
```

Files

Item	Description
/usr/sbin/pdmode	Contains the pdmode command.

pdrmdir Command

Purpose

Deletes partitioned directories.

Syntax

```
pdrmdir dirname ...
```

Description

The **pdrmdir** command deletes partitioned directories that you specify using the *dirname* parameter. Normal users can delete partitioned directories if the Discretionary Access Control (DAC), the Mandatory Access Control (MAC) and the Mandatory Integrity Control (MIC) permissions allow the user to delete the directory. Authorized users with the **aix.mls.pdir.rmdir** authorization can override the DAC, MAC and MIC permissions.

The **pdrmdir** command removes only empty partitioned subdirectories and does not remove files or directories within partitioned subdirectories. The partitioned directory is removed after all the partitioned subdirectories are removed and the directory is empty. The removal of partitioned directory fails if a file exists.

Security

All users can execute the **pdrmdir** command. To successfully perform specific functions, users need the following authorization:

Item	Description
aix.mls.pdir.rmdir	Required to remove directories in a path ignoring the DAC, MAC and MIC permissions.

Exit Status

The **pdrmdir** command returns the following exit values:

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

Examples

1. To delete a partitioned directory, enter:

```
pdrmdir partdir
```

Files

Item	Description
/usr/sbin/pdrmdir	Contains the pdrmdir command.

pdset Command

Purpose

Converts normal directories to partitioned directories.

Syntax

```
pdset dirname ...
```

Description

The **pdset** command converts normal directories that you specify using the *dirname* parameter to partitioned directories.

The directory names that you specify cannot be a partitioned subdirectory or a partitioned sub-subdirectory. Existing subdirectories or files under this directory can only be accessible in the real mode of the partitioned directory.

Security

Only authorized users can run the **pdset** command.

Item	Description
aix.mls.pdir.set	Required for converting normal directories to partitioned directories.

Exit Status

The **pdset** command returns the following exit values:

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

Examples

1. To convert a directory to a partitioned directory, enter:

```
pdset testdir
```

Files

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

penable Command

Purpose

Enables or reports the availability of login ports.

Syntax

```
penable [ -a ] [ Device ]
```

Description

The **penable** command enables normal ports. Normal ports are asynchronous and only allow users to log in. No outgoing use of the port is allowed while it is enabled. The system enables a port by updating an entry in the `/etc/inittab` file and then sending a signal to the **init** process. After receiving the signal and reading the updated status entry, the process takes the appropriate action.

Use the *Device* parameter to specify the ports to be enabled. Permitted values include:

- Full device name, such as the `/dev/tty1` device
- Simple device name, such as the `tty1` device
- A number (for example, 1 to indicate the `/dev/tty1` device).

If you do not specify a *Device* parameter, the **penable** command reports the names of the currently enabled normal ports.

Note: You must have root user authority to run this command.

Flags

Item	Description
<code>-a</code>	Enables all normal ports.

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 enable all normal ports listed in the **/etc/inittab** file, enter:

```
penable -a
```

Files

Item	Description
/etc/locks	Contains lock files for the pshare and pdelay commands.
/usr/sbin/penable	Contains the penable command.

perfwb Command

Purpose

Starts the Performance Workbench to monitor system activity

Syntax

perfwb

Note: The DISPLAY environment variable must be set.

Description

The **perfwb** command is used to start the Performance Workbench. It is a graphical interface to monitor the system activity and processes.

A panel shows the partition configuration and the CPU and memory consumptions.

Another panel lists the top processes that can be sorted on the different provided metrics. A filtering device is also provided to restrict the list to particular processes.

Exit Status

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

Location

/usr/bin/perfwb

Files

Item	Description
/usr/bin/perfwb	Contains the perfwb command.

Item	Description
<code>\$HOME/workspace</code>	Contains the perfwb working directory that contains preferences.

pg Command

Purpose

Formats files to the display.

Syntax

```
pg [ - Number ] [ -c ] [ -e ] [ -f ] [ -n ] [ -p String ] [ -r ] [ -s ] [ +LineNumber ] [ +/Pattern/ ] [ File ... ]
```

Description

The **pg** command reads a file name from the *File* parameter and writes the file to standard output one screen at a time. If you specify a - (dash) as the *File* parameter, or run the **pg** command without options, the **pg** command reads standard input. Each screen is followed by a prompt. If you press the Enter key, another page is displayed. Subcommands used with the **pg** command let you review or search in the file.

To determine workstation attributes, the **pg** command scans the file for the workstation type specified by the **TERM** environment variable. The default type is **dumb**.

When the **pg** command pauses and issues a prompt, you can issue a subcommand. Some of these subcommands change the display to a particular place in the file, some search for specific patterns in the text, and others change the environment in which the **pg** command works.

Changing Location Within the File

The following subcommands display a selected place in the file:

Item	Description
<i>Page</i>	Displays the page specified by the <i>Page</i> parameter.
+Number	Displays the page obtained by adding the <i>Number</i> value to the current page.
-Number	Displays the page as specified by the <i>Number</i> value before the current page.
l	(Lowercase L) Scrolls the display one line forward.
<i>Number</i> l	Displays at the top of the screen the line specified by the <i>Number</i> parameter.
+Number l	Scrolls the display forward for the specified number of lines.
-Number l	Scrolls the display backward for the specified number of lines.
d	Scrolls half a screen forward. Pressing the Ctrl-D key sequence functions the same as the d subcommand.
-d	Scrolls half a screen backward. Pressing the -Ctrl-D key sequence functions the same as the -d subcommand.
Ctrl-L	Displays the current page again. A single . (dot) functions the same as the Ctrl-L key sequence subcommand.
\$	Displays the last page in the file. Do not use this when the input is from a pipeline.

Searching for Text Patterns

The following subcommands search for text patterns in the text. (You can also use the patterns described in the **ed** command.) They must always end with a new-line character, even if the **-n** flag is used.

In an expression such as `[k . a - z] k .`, the minus implies a range, as in a through z, according to the current collating sequence. A collating sequence defines equivalence classes for use in character ranges.

Item	Description
[Number]/Pattern/	Searches for the occurrence of the <i>Pattern</i> value as specified by the <i>Number</i> variable. The search begins immediately after the current page and continues to the end of the current file, without wraparound. The default for the <i>Number</i> variable is 1.

Number?Pattern?

Item	Description
Number^Pattern^	Searches backward for the occurrence of the <i>Pattern</i> value as specified by the <i>Number</i> variable. The searching begins immediately before the current page and continues to the beginning of the current file, without wraparound. The default for the <i>Number</i> variable is 1. The ^ notation is useful for Adds 100 terminals which will not properly handle the ? notation.

After searching, the **pg** command displays the line with the matching pattern at the top of the screen. You can change the position of the display by adding the **m** or **b** suffix to the search command. The **m** suffix displays the line with the matching pattern in the middle of the screen for all succeeding subcommands. The **b** suffix displays the line with the matching pattern at the bottom of the screen for all succeeding subcommands. The **t** suffix displays the line with the matching pattern at the top of the screen again.

Changing the pg Environment

You can change the **pg** command environment with the following subcommands:

Item	Description
[Number]n	Begins examining the next file in the command line, as specified by the <i>Number</i> variable. The default for the <i>Number</i> variable is first.
[Number]p	Begins examining the previous file on the command line, as specified by the <i>Number</i> variable. The default for the <i>Number</i> variable is first.
[Number]w	Displays another window of text. If the <i>Number</i> variable is specified, sets the window size to the number of lines it specifies. This subcommand is the same as the [Number]z subcommand.
[Number]z	Displays another window of text. If the <i>Number</i> variable is specified, sets the window size to the number of lines it specifies. This subcommand is the same as the [Number]w subcommand.
s File	Saves the input in the specified file. Only the current file being examined is saved. This command must always end with a new-line character, even if you specify the -n flag.
h	Displays an abbreviated summary of available subcommands.
q or Q	Quits the pg command.
!Command	Sends the specified command to the shell named in the SHELL environment variable. If this is not available, the default shell is used. This command must always end with a new-line character, even if the -n flag is used.

Attention:

1. Some output is lost when you press the QUIT WITH DUMP (Ctrl-\) or INTERRUPT (Ctrl-C) key sequence because any characters waiting in the output queue are purged when the **QUIT** signal is received.
2. If workstation tabs are not set every eight positions, unpredictable results can occur.

At any time output is being sent to the workstation, you can press the QUIT WITH DUMP or INTERRUPT key sequence. This causes the **pg** command to stop sending output and displays the prompt. Then you can enter one of the preceding subcommands at the command prompt.

If standard output is not a workstation, the **pg** command acts like the **cat** command, except that a header is displayed before each file.

While waiting for workstation input, the **pg** command stops running when you press the INTERRUPT key sequence. Between prompts these signals interrupt the current task and place you in the prompt mode.

Flags

Item	Description
-c	Moves the cursor to the home position and clears the screen before each page. This flag is ignored if the <code>clear_screen</code> field is not defined for your workstation type in the terminfo file.
-e	Does not pause at the end of each file.
-f	Does not split lines. Normally, the pg command splits lines longer than the screen width.
-n	Stops processing when a pg command letter is entered. Normally, commands must end with a new-line character.
-p <i>String</i>	Uses the specified string as the prompt. If the <i>String</i> contains a %d value, that value is replaced by the current page number in the prompt. The default prompt is : (colon). If the specified string contains spaces, you must enclose the string in quotation marks.
-r	Prevents shell escape when the !" subcommand is used.
-s	Highlights all messages and prompts.
+<i>LineNumber</i>	Starts at the specified line number.
-<i>Number</i>	Specifies the number of lines in the window. On workstations that contain 24 lines, the default is 23.
+/<i>Pattern</i>/	Starts at the first line that contains the specified pattern.

Exit Status

This command returns the following exit values:

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

Example

To look at the contents of a file one page at a time, enter:

```
pg filename
```

Files

Item	Description
/usr/bin/pg	Contains the pg command.
/usr/share/lib/terminfo/*	Contains the terminfo file that defines terminal types.

Item	Description
<code>/tmp/pg*</code>	Contains the temporary file created when using <code>pg</code> command.

phold Command

Purpose

Disables or reports the availability of login ports on hold.

Syntax

```
phold [ -a ] [ Device ]
```

Description

The **phold** command disables a set of login ports. The **phold** command allows logged-in users to continue, but does not allow any more users to log in. A user cannot log in on a disabled port. The system disables a port by updating an entry in the `/etc/inittab` file and then sending a signal to the **init** process. When the **init** process receives the signal and reads the updated status entry, it takes the appropriate action.

Use the *Device* parameter to specify the ports to be disabled. Permitted values include:

- A full device name, such as the `/dev/tty1` device
- A simple device name, such as the `tty1` device
- A number (e.g., 1 to indicate the `/dev/tty1` device)

If you do not specify a *Device* parameter, the **phold** command reports the names of currently disabled ports in its set.

Note: You must have root user authority to run this command.

Flags

Item	Description
<code>m</code>	

<code>-a</code>	Holds all ports that are currently enabled.
-----------------	---

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 list the ports that are currently on hold, enter:

```
phold
```


Files

Item	Description
<code>/etc/locks</code>	Contains lock files for the pshare and pdelay commands.
<code>/etc/phold</code>	Contains the phold command.

pic Command

Purpose

Preprocesses **troff** command input for the purpose of drawing pictures.

Syntax

```
pic [ -T Name ] [ - | File ... ]
```

Description

The **pic** command is a **troff** command preprocessor for drawing simple figures on a typesetter. The basic objects are a box, circle, ellipse, line, spline, arrow, arc, and the text specified by the *Text* variable. The top-level object is the picture.

Item Description

File Specifies the output from a **troff** command that is processed by the **pic** command to draw pictures.

Pictures

The top-level object in the **pic** command is the picture.

.PS *OptionalWidth OptionalHeight*

ElementList

.PE

If the **.PF** macro is used instead of the **.PE** macro, the position after printing is restored to what it was upon entry.

Item	Description
<i>OptionalWidth</i>	Specifies the width of the picture (in inches), if present, regardless of any dimensions used internally. The maximum value is 8.5.
<i>OptionalHeight</i>	Specifies a height value, in inches, different from the default, which is scaled to the same proportion. The maximum value is 14.

Item	Description
<i>ElementList</i>	Represents the following list of elements: Shape AttributeList <i>For Statement</i> Placename: Element <i>If Statement</i> Placename: Position <i>Copy Statement</i> Variable = Expression <i>Print Statement</i> Direction <i>Plot Statement</i> { List of Elements } <i>sh X Commandline X</i> [List of Elements] <i>troff-command</i>

Variable names begin with a lowercase letter, followed by zero or more letters or numbers. Place names begin with an uppercase letter, followed by zero or more letters or numbers. Place and variable names retain their values from one picture to the next.

Elements in a list must be separated by new-line characters or ; (semicolon); a long element can be continued by ending the line with a \ (backslash). Comments are introduced by a # character and ended by a new-line character.

Primitives

The primitive objects are as follows:

- box**
- circle**
- ellipse**
- arc**
- line**
- arrow**
- spline**
- move**

Text-List

The **arrow** object is the same as the **line** object with the -> attribute.

Attributes

An *AttributeList* element is a sequence of zero or more attributes; each attribute consists of a keyword, perhaps followed by a value.

Attribute	Attribute
h(eigh)t <i>Expression</i>	wid(th) <i>Expression</i>
rad(ius) <i>Expression</i>	diam(eter) <i>Expression</i>
up <i>OptionalExpression</i>	down <i>OptionalExpression</i>
right <i>OptionalExpression</i>	left <i>OptionalExpression</i>
from <i>Position</i>	to <i>Position</i>

Attribute	Attribute
at <i>Position</i>	with <i>Corner</i>
by <i>Expression, Expression</i>	then
dotted <i>OptionalExpression</i>	dashed <i>OptionalExpression</i>
chop <i>OptionalExpression</i>	-> <- <->
invis	same
Text-list	

Missing attributes and values are filled in from defaults. Not all attributes make sense for all primitives; irrelevant ones are not processed. The following are the currently meaningful attributes:

Item	Description
Primitives	Attributes
box	h(eigh)t, wid(th), at, same, dotted, dashed, invis, Text
circle, ellipse	rad(ius), diam(eter), h(eigh)t, wid(th), at, same, invis, Text
arc	up, down, left, right, h(eigh)t, wid(th), from, to, at, rad(ius), invis, ccw, cw, <-, ->, <->, Text
line, arrow	up, down, left, right, h(eigh)t, wid(th), from, to, by, then, at, same, dotted, dashed, invis, <-, ->, <->, Text
spline	up, down, left, right, h(eigh)t, wid(th), from, to, by, then, at, same, invis, <-, ->, <->, Text
move	up, down, left, right, to, by, same, Text
<i>Text-list</i>	at, Text-item

The **at** attribute implies placing the geometrical center at the specified place. For lines, splines, and arcs, the **h(eigh)t** and **wid(th)** attributes refer to arrowhead size.

The *Text-item* variable is usually an attribute of some primitive; by default, it is placed at the geometrical center of the object. Stand-alone text is also permitted. A *Text-list* primitive is a list of text items; a text item is a quoted string optionally followed by a positioning request, as follows:

"..."

"..." **center**

"..." **ljust**

"..." **rjust**

"..." **above**

"..." **below**

If there are multiple text items for some primitives, they are centered vertically except as qualified. Positioning requests apply to each item independently.

Text items can contain **troff** commands that control, for example, size and font changes and local motions. Make sure these commands are balanced so that the entering state is restored before exiting.

Item	Description
Positions/Places	<p>A position is ultimately an X,Y coordinate pair, but it can also be specified in the following ways:</p> <p><i>Place</i></p> <p>(<i>Position</i>)</p> <p><i>Expression</i>, <i>Expression</i></p> <p>(<i>Position</i>) [+/- (<i>Expression</i>, <i>Expression</i>)]</p> <p>(<i>Position</i>) [+/- <i>Expression</i>, <i>Expression</i>]</p> <p>(<i>Place1</i>, <i>Place2</i>)</p> <p>(<i>Place1.X</i>, <i>Place2.Y</i>)</p> <p><i>Expression</i> < <i>Position</i>, <i>Position</i> ></p> <p><i>Expression</i> [of the way] between <i>Position</i> and <i>Position</i></p> <p><i>Placename</i> [<i>Corner</i>]</p> <p><i>Corner Placename</i></p> <p>Here</p> <p>Corner of <i>Nth</i> <i>Shape</i></p> <p><i>Nth shape</i> [<i>Corner</i>]</p> <p>Note: A <i>Corner</i> variable designates one of the eight compass points or the center, beginning, or end of a primitive, as follows:</p> <p>.n .e .w .s .ne .se .nw .sw</p> <p>.t .b .r .l</p> <p>c .start .end</p>

Each object in a picture has an ordinal number; *Nth* refers to this, as follows:

- *Nth*
- *Nth* last

The **pic** command is flexible enough to accept names like **1th** and **3th**. Usage like **1st** and **3st** are accepted as well.

Variables

The built-in variables and their default values are as follows:

Item	Description
boxwid	0.75
boxht	0.5
circlerad	0.25
arcrad	0.25
ellipsewid	0.75
ellipseht	0.5
linewid	0.5
lineht	0.5
movewid	0.5

Item	Description
moveht	0.5
arrowwid	0.05
arrowht	0.1
textwid	0
textht	0
dashwid	0.5
scale	1

These default values can be changed at any time, and the new values remain in force from picture to picture until changed again.

The **textht** and **textwid** variables can be set to any value to control positioning. The width and height of the generated picture can be set independently from the **.PS** macro line. Variables changed within the [(left bracket) delimiter and the] (right bracket) delimiter revert to their previous value upon exit from the block. Dimensions are divided by **scale** during output.

Note: The **pic** command has an eight inch by eight inch limitation on picture sizes generated and sent to the **troff** command, even when the **.ps** (size) line specifies a size greater than eight inches.

Expressions

The following **pic** command expressions are evaluated in floating point. All numbers representing dimensions are taken to be in inches.

Expression + Expression

Expression - Expression

*Expression * Expression*

Expression / Expression

Expression % Expression (modulus)

- Expression

(Expression)

variable

number

Place .x

Place .y

Place .ht

Place .wid

Place .rad

sin(Expression) **cos(Expression)** **atan2(Expression, Expression)** **log(Expression)** **sqrt(Expression)**
int(Expression) **max(Expression, Expression)** **min(Expression, Expression)** **rand(Expression)**

Logical Operators

The **pic** command provides the following operators for logical evaluation:

Item	Description
!	Not
>	Greater than

Item	Description
<	Less than
>/=	Greater than or equal to
</=	Less than or equal to
&&	And
	Or
==	Equal to
!=	Not equal to

Definitions

The following **define** statement is not part of the grammar:

```
define Name X Replacement text X
```

Occurrences of values such as **\$1** and **\$2** in the *Replacement text* variable are replaced by the corresponding options if the *Name* variable is called, as follows:

```
Name(Option1, Option2, ...)
```

Non-existent options are replaced by null strings. The *Replacement text* variable can contain newline characters.

copy and copy thru Statements

The **copy** statement includes data from a file or values that immediately follow, such as:

```
copy File
```

```
copy thru Macro
```

```
copy File thru Macro
```

```
copy File thru Macro until String
```

The *Macro* parameter value can be either the name of a defined macro or the body of a macro enclosed in some character not part of the body. If no file name is given, the **copy** statement copies the input until the next **.PE** macro line.

for Loops and if Statements

The **for** and **if** statements provide for loops and decision-making, as follows:

```
Variable=Expression to Expression by Expression do X anything X
```

```
if Expression then X anything X else X anything X
```

The **by** and **else** clauses are optional. The *Expression* variable in an **if** statement can use the usual relational operators or the *String1* == (or !=) *String2* string tests.

Miscellaneous Information

The **sh** command runs a command line, as follows:

```
sh X Commandline X
```

It is possible to plot the value of an expression, as follows:

```
plot Expression OptionalFormat Attributes
```

The *Expression* variable value is evaluated and converted to a string (using the format specification, if provided).

The state of fill or no-fill mode is preserved with respect to pictures.

Input numbers can be expressed in **E** (exponential) notation.

Flags

Item	Description
-TName	Prepares the output for the specified printing device. Possible values for <i>Name</i> are: ibm3812 3812 Pageprinter. ibm3816 3816 Pageprinter. hplj Hewlett-Packard LaserJet II. ibm5587G 5587-G01 Kanji Printer multi-byte language support. psc PostScript printer. X100 AIXwindows display. X100K AIXwindows display for multi-byte character support.

The default is **ibm3816**.

Note: It is possible to set the **TYPESETTER** environment variable to one of the preceding values instead of using the **-T Name** flag of the **troff** command.

- Reverts to standard input.

pick Command

Purpose

Selects messages by content and creates and modifies sequences.

Syntax

```
pick [ +Folder ] [ Messages ] [ -datefield Field ] [ -not ] [ -lbrace ] [ -after Date ] [ -before Date ]  
[ -cc "Pattern" ] [ -date "Pattern" ] [ -from "Pattern" ] [ -search "Pattern" ] [ -to "Pattern" ] [ -Component  
"Pattern" ] [ -rbrace ] [ -and ] [ -or ] [ -sequence Name ] [ -zero | -nozero ] [ -public | -npublic ] [ -list |  
-nolist ]
```

Description

The **pick** command selects messages containing particular character patterns or particular dates. You can use the **-and**, **-or**, **-not**, **-lbrace**, and **-rbrace** flags to construct compound conditions for selecting messages.

Flags

Item

-after *Date*

Description

Selects messages with dates later than that specified by the *Date* variable. Use the following specifications for the *Date* variable:

yesterday	today	tomorrow
sunday	monday	tuesday
wednesday	thursday	friday
saturday	-Days	SystemDate

The **pick** command treats the days of the week as days in the past. For example, **monday** means last Monday, not today or next Monday. You can use the *-Days* argument to specify a number of days in the past. For example, *-31* means 31 days ago. For the *SystemDate* argument, you can specify any valid format defined for your system.

-and

Forms a logical AND operation between two message-selecting flags; for example, `pick -after Sunday -and -from mark`. The **-and** flag has precedence over the **-or** flag, but the **-not** flag has precedence over the **-and** flag. Use the **-lbrace** and **-rbrace** flags to override this precedence.

-before *Date*

Selects messages with dates earlier than the specified date. See the **-after** flag on how to specify *Date*.

-cc "*Pattern*"

Selects messages that contain the character string specified by the "*Pattern*" variable in the `cc :` field.

-date "*Pattern*"

Selects messages that contain the character string specified by the "*Pattern*" variable in the `Date :` field.

-datefield *Field*

Specifies which dated field is parsed when the **-after** and **-before** flags are given. By default, the **pick** command uses the `Date :` field.

+Folder

Identifies the folder that contains the messages you wish to pick. By default, the system uses the current folder.

-from "*Pattern*"

Selects messages that contain the character string specified by the "*Pattern*" variable in the `From :` field.

-help

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

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

-lbrace

Groups **-and**, **-or**, and **-not** operations. Operations between the **-lbrace** and **-rbrace** flags are evaluated as one operation. You can nest the **-lbrace** and **-rbrace** flags.

-list

Sends a list of selected message numbers to standard output. This allows you to use the **pick** command to generate message numbers to use as input for other commands. For example, to scan all messages in the current folder that were sent after Tuesday, you would enter the following:

```
scan 'pick -after tuesday -list'
```

If you do not specify a sequence, the **-list** flag is the default.

Item	Description
<i>Messages</i>	<p>Specifies the messages to search. You can specify several messages, a range of messages, or a single message. Use the following to specify messages:</p> <p>Number Number of the message.</p> <p>Sequence A group of messages specified by the user. Recognized values include:</p> <p>all All of the messages in the folder. This is the default.</p> <p>cur or . (period) Current message.</p> <p>first First message in a folder.</p> <p>last Last message in a folder.</p> <p>new New message that is created.</p> <p>next Message following the current message.</p> <p>prev Message preceding the current message.</p>
-nolist	Prevents the pick command from generating a list of the selected message numbers. If a sequence is specified, the -nolist flag is the default.
-nopublic	Restricts a sequence to your usage. The -nopublic flag does not restrict the messages in a sequence, only the sequence itself. This option is the default if the folder is write-protected from other users.
-not	Forms a logical NOT operation on a message-selecting flag; for example, <code>pick -not -from george</code> . This construction evaluates all messages not chosen by the message-selecting flag. The -not flag has precedence over the -and flag, and the -and flag has precedence over the -or flag. Use the -lbrace and -rbrace flags to override this precedence.
-nozero	Appends the selected messages to the specified sequence.
-or	Forms a logical OR operation on two message-selecting flags; for example, <code>pick -from amy -or -from mark</code> . The -not flag has precedence over the -and flag, and the -and flag has precedence over the -or flag. Use the -lbrace and -rbrace flags to override this precedence.
-public	Allows other users access to a sequence. The -public flag does not make protected messages available, only the sequence itself. This option is the default if the folder is not write-protected from other users.
-rbrace	Groups -and , -or , and -not operations. Operations between the -lbrace and -rbrace flags are evaluated as one operation. You can nest the -lbrace and -rbrace flags.
-search "Pattern"	Selects messages that contain the character string specified by the "Pattern" variable anywhere in the message.

Item	Description
-sequence <i>Name</i>	Stores the messages selected by the pick command in the sequence specified by the <i>Name</i> variable.
-to " <i>Pattern</i> "	Selects messages that contain the character string specified by the " <i>Pattern</i> " variable in the To : field.
-zero	Clears the specified sequence before placing the selected messages into the sequence. This flag is the default.
-Component " <i>Pattern</i> "	Selects messages that contain the character string specified by the " <i>Pattern</i> " variable in the heading field specified by the <i>Component</i> variable; for example, <code>pick -reply-to amy</code> .

Profile Entries

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

Item	Description
Current-Folder:	Sets the default current folder.
Path:	Specifies the user's MH 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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To get a list of message numbers in the current folder that are from user jones, enter:

```
pick -from jones
```

The system responds with a message similar to the following:

```
12
15
19
```

2. To see a list of message numbers in the schedule folder received within the last 30 days, enter:

```
pick + schedule -after -30
```

The system responds with a message similar to the following:

```
5
8
21
30
```

Files

Item	Description
\$HOME/.mh_profile	Contains the user's MH profile.
/usr/bin/pick	Contains the pick command.

ping Command

Purpose

Sends an echo request to a network host.

Syntax

```
ping [ -d ] [ -D ] [ -n ] [ -q ] [ -r ] [ -v ] [ -R ] [ -a addr_family ] [ -c Count ] [ -w timeout ] [ -f |  
-i Wait ] [ -l Preload ] [ -p Pattern ] [ -s PacketSize ] [ -S hostname/IP addr ] [ -L ] [ -I a.b.c.d. ]  
[ -o interface ] [ -T tll ] [ -E Interface_name | Source_Hostname | Source_IP_address ] Host [ PacketSize ]  
[ Count ]
```

Description

The `/usr/sbin/ping` command sends an Internet Control Message Protocol (ICMP) ECHO_REQUEST to obtain an ICMP ECHO_RESPONSE from a host or gateway. The **ping** command is useful for the following functions:

- Determining the status of the network and various foreign hosts.
- Tracking and isolating hardware and software problems.
- Testing, measuring, and managing networks.

If the host is operational and on the network, it responds to the echo. Each echo request contains an Internet Protocol (IP) and ICMP header, followed by a ping PID and a **timeval** structure, and enough bytes to complete the packet. The default is to continuously send echo requests until an Interrupt is received (Ctrl-C).

The **ping** command sends one datagram per second and prints one line of output for every response received. The **ping** command calculates round-trip times and packet loss statistics, and displays a brief summary on completion. The **ping** command completes when the program times out or on receipt of a **SIGINT** signal. The *Host* parameter is either a valid hostname or Internet address.

By default, the **ping** command continues to send echo requests to the display until an Interrupt is received (Ctrl-C). The Interrupt key can be changed by using the **stty** command.

Because of the load that continuous echo requests can place on the system, repeated requests must be used primarily for problem isolation.

Flags

Item	Description
-c <i>Count</i>	Specifies the number of echo requests, as indicated by the <i>Count</i> variable, to be sent (and received).
-w <i>timeout</i>	This option works only with the -c option. It causes ping to wait for a maximum of 'timeout' seconds for a reply (after sending the last packet).
-d	Starts socket-level debugging.
-D	This option causes a hex dump to standard output of ICMP ECHO_REPLY packets.

Item	Description
> > -E <i>Interface_name</i> <i>Source_Hostname</i> <i>Source_IP_address</i>	<p>Sends or receives ping packets over specific interface for both IPv4 and IPv6 protocols. When you specify an interface name, a source hostname, or a source IP address that is not one of the machine's interface addresses, an error is returned and no ping packets are sent. If you specify an interface name with the -E flag, by default, the number of echo requests (-c Count) is set to a value of 5.</p> <p>Notes:</p> <ul style="list-style-type: none"> • The -E flag ensures to send or receive ping packet over specific interface. • The -E flag works only on destination media access control (MAC) address, and not on the saved routes. Therefore, the -E flag works in the IPv4 protocol only if the address resolution protocol (ARP) is resolved. The -E flag works in the IPv6 protocol only if the neighbor discovery protocol (NDP) resolves an IPv6 address into a MAC address. • You cannot run the -R flag with the -E option because the -E flag does not work on saved routes.
	<p><<<</p>
-f	<p>Specifies flood-ping option. The -f flag "floods" or outputs packets as fast as they come back or 100 times per second, whichever is more. For every ECHO_REQUEST sent, a . (period) is printed, while for every ECHO_REPLY received, a backspace is printed. This provides a rapid display of how many packets are being dropped. Only the root user can use this option.</p> <p>Note: Use this flag with caution. Flood pinging is only permitted by the root user. The -f flag is incompatible with the -i Wait flag.</p>
-I a.b.c.d	<p>Specifies that the interface specified by <i>a.b.c.d</i> is to be used for outgoing IPv4 multicasts. The -I flag is an uppercase i.</p>
-o interface	<p>Specifies that <i>interface</i> is to be used for outgoing IPv6 multicasts. The interface is specified in the form 'en0', 'tr0', and so on.</p>
-i Wait	<p>Waits the number of seconds specified by the <i>Wait</i> variable between the sending of each packet. The default is to wait for one second between each packet. This option is incompatible with the -f flag.</p>
-L	<p>Disables local loopback for multicast pings.</p>
-l Preload	<p>Sends the number of packets specified by the <i>Preload</i> variable as fast as possible before falling into normal mode of behavior (one per second). The -l flag is a lowercase l.</p>
-n	<p>Specifies numeric output only. No attempt is made to look up symbolic names for host addresses.</p>
-p Pattern	<p>Specifies up to 16 'pad' bytes to complete the packet you send. This is useful for diagnosing data-dependent problems in a network. For example, -p ff fills the packet with all 1's.</p>
-q	<p>Specifies quiet output. Nothing is displayed except the summary lines at startup time and when finished.</p>
-r	<p>Bypasses the routing tables and sends directly to a host on an attached network. If the <i>Host</i> is not on a directly connected network, the ping command generates an error message. This option can be used to ping a local host through an interface that no longer has a route through it.</p>

Item	Description
-R	Specifies record route option. The -R flag includes the RECORD_ROUTE option in the ECHO_REQUEST packet and displays the route buffer on returned packets. Note: The IP header is only large enough for nine such routes. Also, many hosts and gateways ignore this option.
-a <i>addr_family</i>	Maps the destination address of the ICMP packets to IPv6 format if <i>addr_family</i> is equal to "inet6".
-s <i>PacketSize</i>	Specifies the number of data bytes to be sent. The default is 56, which converts into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.
-S <i>hostname/IP addr</i>	Uses the IP address as the source address in outgoing ping packets. On hosts with more than one IP address, the -S flag can be used to send or receive ping packets by using a specific source IP address. If the IP address is not one of the machine's interface addresses, an error is returned and nothing is sent. Note: You can use the -S flag to send or receive the ping packet by using a specific source IP address, but the -S flag does not guarantee to send or receive ping packet over specific interface. Instead, use -E flag that ensures to send or receive ping packet over specific interface.
-T <i>tll</i>	Specifies that the time to live for a multicast packet is <i>tll</i> seconds.
-v	Requests verbose output, which lists ICMP packets that are received in addition to echo responses.

Parameters

Item	Description
<i>PacketSize</i>	Specifies the number of data bytes to be sent. The default is 56, which converts into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data. This parameter is included for compatibility with previous versions of the ping command.
<i>Count</i>	Specifies the number of echo requests to be sent (and received). This parameter is included for compatibility with previous versions of the ping 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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To check the network connection to host canopus and specify the number of echo requests to send, enter:

```
ping -c 5 canopus
```

OR

```
ping canopus 56 5
```

Information similar to the following is displayed:

```
PING canopus.austin.century.com: (128.116.1.5): 56 data bytes
64 bytes from 128.116.1.5: icmp_seq=0 ttl=255 time=2 ms
64 bytes from 128.116.1.5: icmp_seq=1 ttl=255 time=2 ms
```

```
64 bytes from 128.116.1.5: icmp_seq=2 ttl=255 time=3 ms
64 bytes from 128.116.1.5: icmp_seq=3 ttl=255 time=2 ms
64 bytes from 128.116.1.5: icmp_seq=4 ttl=255 time=2 ms

----canopus.austin.century.com PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 2/2/3 ms
```

2. To get information about host lear and start socket-level debugging, enter:

```
ping -d lear
```

Information similar to the following is displayed:

```
PING lear.austin.century.com: (128.114.4.18) 56 data bytes
64 bytes from 128.114.4.18: icmp_seq=0 ttl=255 time=6 ms
64 bytes from 128.114.4.18: icmp_seq=1 ttl=255 time=17 ms
64 bytes from 128.114.4.18: icmp_seq=2 ttl=255 time=6 ms
64 bytes from 128.114.4.18: icmp_seq=3 ttl=255 time=6 ms
64 bytes from 128.114.4.18: icmp_seq=4 ttl=255 time=6 ms
^C
----lear.austin.century.com PING Statistics ----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 6/8/17 ms
```

Note: The output is repeated until an Interrupt (Ctrl-C) is received.

3. To obtain information about host opus and specify the number of data bytes to be sent, enter:

```
ping -s 2000 opus
```

OR

```
ping opus 2000
```

Information similar to the following is displayed:

```
PING opus.austin.century.com: (129.35.34.234): 2000 data bytes
2008 bytes from 129.35.34.234: icmp_seq=0 ttl=255 time=20 ms
2008 bytes from 129.35.34.234: icmp_seq=1 ttl=255 time=19 ms
2008 bytes from 129.35.34.234: icmp_seq=2 ttl=255 time=20 ms
2008 bytes from 129.35.34.234: icmp_seq=3 ttl=255 time=20 ms
2008 bytes from 129.35.34.234: icmp_seq=4 ttl=255 time=20 ms
2008 bytes from 129.35.34.234: icmp_seq=5 ttl=255 time=19 ms
2008 bytes from 129.35.34.234: icmp_seq=6 ttl=255 time=19 ms
^C
----opus.austin.century.com PING Statistics----
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 19/19/20 ms
```

Note: The output is repeated until an Interrupt (Ctrl-C) is received.

4. To invoke the flood-ping option to host stlopnor, enter:

```
ping -f stlopnor
```

Information similar to the following is displayed:

```
Ping stlopnor.austin.century.com: (129.35.34.234): 56 data bytes
.^C
----stlopnor.austin.century.com PING Statistics ----
1098 packets transmitted, 1097 packets received, 0% packet loss
round-trip min/avg/max = 4/4/11
```

Note: The output is repeated until an Interrupt (Ctrl-C) is received.

5. To specify an interval of five seconds between packets sent to host opus, enter:

```
ping -i5 opus
```

Information similar to the following is displayed:

```
PING opus.austin.century.com: (129.35.34.234): 56 data bytes
64 bytes from 129.35.34.234: icmp_seq=0 ttl=255 time=5 ms
64 bytes from 129.35.34.234: icmp_seq=1 ttl=255 time=5 ms
64 bytes from 129.35.34.234: icmp_seq=2 ttl=255 time=6 ms
^C
----opus.austin.century.com PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 5/5/6 ms
```

Note: The output is repeated until an Interrupt (Ctrl-C) is received.

6. To send the number of packets specified by the *Preload* variable as fast as possible before falling into normal mode of behavior to host opus, enter:

```
ping -l 10 opus
```

Information similar to the following is displayed:

```
PING opus.austin.century.com: (129.35.34.234): 56 data bytes
64 bytes from 129.35.34.234: icmp_seq=0 ttl=255 time=9 ms
64 bytes from 129.35.34.234: icmp_seq=1 ttl=255 time=11 ms
64 bytes from 129.35.34.234: icmp_seq=2 ttl=255 time=16 ms
64 bytes from 129.35.34.234: icmp_seq=3 ttl=255 time=22 ms
64 bytes from 129.35.34.234: icmp_seq=4 ttl=255 time=26 ms
64 bytes from 129.35.34.234: icmp_seq=5 ttl=255 time=27 ms
64 bytes from 129.35.34.234: icmp_seq=6 ttl=255 time=30 ms
64 bytes from 129.35.34.234: icmp_seq=7 ttl=255 time=31 ms
64 bytes from 129.35.34.234: icmp_seq=8 ttl=255 time=33 ms
64 bytes from 129.35.34.234: icmp_seq=9 ttl=255 time=35 ms
64 bytes from 129.35.34.234: icmp_seq=10 ttl=255 time=36 ms
64 bytes from 129.35.34.234: icmp_seq=11 ttl=255 time=6 ms
64 bytes from 129.35.34.234: icmp_seq=12 ttl=255 time=6 ms
64 bytes from 129.35.34.234: icmp_seq=13 ttl=255 time=6 ms
64 bytes from 129.35.34.234: icmp_seq=14 ttl=255 time=7 ms
64 bytes from 129.35.34.234: icmp_seq=15 ttl=255 time=6 ms
^C
----opus.austin.century.com PING Statistics----
16 packets transmitted, 16 packets received, 0% packet loss
round-trip min/avg/max = 6/19/36 ms
```

Note: The output is repeated until an Interrupt (Ctrl-C) is received.

7. To diagnose data-dependent problems in a network, enter:

```
ping -p ff opus
```

This command sends packets with a pad-pattern of all 1's to host opus. Information similar to the following is displayed:

```
PATTERN: 0xff
PING opus.austin.century.com: (129.35.34.234): 56 data bytes
64 bytes from 129.35.34.234: icmp_seq=0 ttl=255 time=5 ms
64 bytes from 129.35.34.234: icmp_seq=1 ttl=255 time=5 ms
64 bytes from 129.35.34.234: icmp_seq=2 ttl=255 time=5 ms
64 bytes from 129.35.34.234: icmp_seq=3 ttl=255 time=6 ms
64 bytes from 129.35.34.234: icmp_seq=4 ttl=255 time=5 ms
^C
----opus.austin.century.com PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 5/5/6 ms
```

Note: The output is repeated until an Interrupt (Ctrl-C) is received.

8. To specify quiet output, enter:

```
ping -q bach
```

Only summary information similar to the following sample is displayed:

```
PING bach.austin.century.com: (129.35.34.234): 56 data bytes
^C
----bach.austin.century.com PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 5/5/8 ms
```

Note: The output is repeated until an Interrupt (Ctrl-C) is received.

9. **>** To send or receive ping packets over a specific interface, enter the following command:

```
ping -E en1 2020:1234:5678:9122:2c6f:57ff:fe6b:8b02
```

An output similar to the following example is displayed:

```
Using Source Address 2020:1234:5678:9122:2c6f:57ff:3e6b:58
PING 2020:1234:5678:9122:2c6f:57ff:fe6b:8b02: (2020:1234:5678:9122:2c6f:57ff:fe6b:8b02): 56
data bytes
64 bytes from 2020:1234:5678:9122:2c6f:57ff:fe6b:8b02: icmp_seq=0 ttl=64 time=1.462 ms
64 bytes from 2020:1234:5678:9122:2c6f:57ff:fe6b:8b02: icmp_seq=1 ttl=64 time=0.917 ms
64 bytes from 2020:1234:5678:9122:2c6f:57ff:fe6b:8b02: icmp_seq=2 ttl=64 time=1.187 ms
64 bytes from 2020:1234:5678:9122:2c6f:57ff:fe6b:8b02: icmp_seq=3 ttl=64 time=1.105 ms
64 bytes from 2020:1234:5678:9122:2c6f:57ff:fe6b:8b02: icmp_seq=4 ttl=64 time=1.114 ms

--- 2020:1234:5678:9122:2c6f:57ff:fe6b:8b02 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0/0/1 ms
```

<

pioattred Command

Purpose

Provides a way to format and edit attributes in a virtual printer.

Syntax

```
pioattred -q PrintQueueName -d QueueDeviceName [ -o Action ] [ -a Attribute ]
```

Description

The **pioattred** command provides a way to format virtual printer attributes and to edit the attributes. Specifically, attributes in the printer definition file can be formatted and/or edited according to the action specified with the **-o** flag. Formatted attributes are written to standard output **stdout**. Attributes are edited with the editor specified in the **VISUAL** environment variable. The virtual printer definition file is assumed to be in the **/var/spool/lpd/pio/@local/custom/*** directory.

Flags

Item	Description
-a <i>Attribute</i>	Specifies the name of the attribute in the virtual printer definition file to format or edit. This flag may be specified many times.
-d <i>QueueDeviceName</i>	Specifies the <i>QueueDeviceName</i> spooler of the virtual printer definition to format or edit.

Item	Description
-o Action	<p>Specifies the action that the pioattred command should take on the virtual printer definition. If this flag is omitted, the pioattred command assumes a value of 0 (zero).</p> <p>0 Format the attributes specified. The result goes to stdout.</p> <p>1 Format and edit the attribute(s) specified; use the editor specified in the VISUAL environment variable. If no editor is specified in the VISUAL environment variable, use the vi editor. If an error is made in editing the attributes, save the erroneous attributes in a temporary file, and return a return code indicating an error.</p> <p>The following values are used in the event that an error return code was returned after editing the attributes.</p> <p>2 Edit the attributes again. The virtual printer definition will be the state it was left in when the error occurred.</p> <p>3 Ignore the error and save the edited attributes in the virtual printer definition.</p> <p>4 Clean up and leave things in the state they were before the pioattred command was started.</p>
-q PrintQueueName	<p>Specifies the <i>PrintQueueName</i> spooler of the virtual printer definition to format or edit.</p>

Examples

1. To format the **ci** and **sh** attributes in the queue: quedev virtual printer definition, enter:

```
pioattred -q queue -d quedev -o 0 -a ci -a sh
```

OR

```
pioattred -q queue -d quedev -a ci -a sh
```

2. To format all attributes in the queue: quedev virtual printer definition, enter:

```
pioattred -q queue -d quedev -o 0
```

OR

```
pioattred -q queue -d quedev
```

3. To edit the **st** attribute in the queue: quedev virtual printer definition, enter:

```
pioattred -q queue -d quedev -o 1 -a st
```

Files

Item	Description
/usr/sbin/pioattred	Contains the pioattred command.

piobe Command

Purpose

Print job manager for the printer backend.

Syntax

```
/usr/lpd/piobe [ -a PreviewLevel ] [ -A DiagnosticLevel ] [ -d InputDataStream ] [ -f FilterName ]  
[ FormatterFlags ] [ File ... ]
```

Description

The **piobe** command is a spooler backend program called by the **qdaemon** program to process a print job. The **piobe** command serves as the print job manager.

Based on the argument of the **-d** flag (or its default value in the database), which specifies the data stream type of the print files, the **piobe** command retrieves a pipeline from the database and passes it to a shell. The pipeline contains a string of filters to convert the print files as necessary and send them to a printer. If requested, the **piobe** command also retrieves and runs pipelines from the database to generate header and trailer pages.

The *FormatterFlags* argument (flags other than the flags listed in this topic) is assumed to be referenced by the filter commands in the pipelines. If a flag is specified but not referenced by the pipelines, an error message is issued and the print job ended.

Note: The **piobe** command should not be typed directly on the command line. This command is invoked by the **qdaemon** process and is dependent on the various services provided by the **qdaemon** process.

Flags

Item

-a *PreviewOption*

Description

Provides a way to preview parameter values that would be used for a print job without actually printing any files. Values that can be specified for the *PreviewOption* variable are:

0

Specifies normal print processing

1

Returns a list of flag values and the pipeline of filters that would be used to convert the input data type to the data type expected by the printer, but does not actually invoke the pipeline of filters or send the file to the printer.

The list of flag values returned are the default command line flag values from the configuration database. These values are overridden by any flag arguments specified on the command line. Please note that:

- Only flags that are valid for the *InputDataType* variable specified (or defaulted) for the **-d** flag are shown.
- Flag values related only to the spooling of your print job, instead of the actual printing, are not shown. The default values for the spooling flags are included with the descriptions of the flags for the **qprt** command.
- The flag values may not have been checked to verify that they are valid.

The pipeline of filters shows the filter commands (and the flag values passed to the filter commands) that would process the data from your print file before it is sent to the printer. You can review the description for each of the filter commands to determine the type of filtering that would be performed.

-A *Value*

Specifies the level of diagnostic output. Diagnostic output is useful for diagnosing errors encountered by a pipeline of filters that is processing a print file, a header page, or a trailer page. Diagnostic output is mailed to the user who submitted the print job. The *Value* variable can be one of the following:

0

Discards any standard error output that is produced by the header, trailer, or print file pipelines.

1

If any standard error output is produced, returns the standard error output and the pipeline that produced it and ends the print job.

2

Returns the flag values, standard error output (if any), and completes pipelines, regardless of whether an error is detected. If an error is detected, the print job is ended.

3

Similar to a value of **2**, except that the file is not printed.

A value of **1** is recommended. A value of **0** is used if a filter in a pipeline produces output to standard error, even if no error is encountered, such as for status information. A value of **2** or **3** is used for diagnosing a problem even if the problem does not cause any output to standard error.

Item	Description
-d <i>InputDataType</i>	<p>Specifies the type of data that is in the file to be printed. This flag is a one-character identifier. Based on the data type for the print file and the data type expected by the printer, the print files are passed through filters (if necessary) before being sent to the printer. Examples of data type identifiers are:</p> <ul style="list-style-type: none"> a IBM extended ASCII p Pass-through (sent to the printer unmodified) s PostScript c PCL d Diablo 630 k Kanji. <p>If the printer you select does not support the <i>InputDataType</i> variable and filters are not available to convert the data type of your print file to a data type supported by the printer, the print job will be ended with an error message.</p>
-f <i>FilterType</i>	<p>Specifies a type of filter through which your print file is passed before being sent to the printer. This flag is a one-character identifier. The identifiers are similar to the filter flags available with the html</p>

pioburst Command

Purpose

Generates burst pages (header and trailer pages) for printer output.

Syntax

/usr/lpd/pio/etc/pioburst [**-H** *HostName*] *TextFile*

Description

The **pioburst** command retrieves prototype text for a burst page from the file specified by the *TextFile* variable, fills in the variable fields identified by **%** escape sequences in the prototype text, and writes the constructed text to standard output. It is invoked as a filter in a pipeline by the print job manager, the **pio** command.

The **%** escape sequences, which are replaced by corresponding values, are:

Item	Description
%A	Specifies the formatting flag values.
%D	Specifies the user to whom the print output is to be delivered.
%H	Specifies the name of the host machine printing the job.
%P	Specifies the time the print job was printed.
%Q	Specifies the time the print job was queued.

Item	Description
%S	Specifies the user who submitted the print job.
%T	Specifies the title of the print job.
%%	Specifies the % (percent sign).

Labels (20 characters long) for each of the variable fields can be specified by using the same escape sequence as for the variable field, except using lowercase letters. For example, to generate a label for the variable field specifying the print job was queued (**%Q**), use **%q**. The **%e** variable represents the label END OF OUTPUT FOR:.

The **pioburst** command requires the following environment variables to be initialized:

Item	Description
PIOTITLE	Title of the print job (for %T)
PIOQDATE	Time the print job was queued (for %Q)
PIOFROM	User who submitted the print job (for %S)
PIOTO	User to whom the print output is to be delivered (for %D)
PIOFLAGS	Flag values (for %A).

Flags

Item	Description
-H <i>HostName</i>	Specifies that the host name designated by the <i>HostName</i> variable override the default host name (the name of the host machine printing the job).

Example

To generate a header page and send it to standard output, enter:

```
pioburst /usr/lpd/pio/burst/H.ascii
```

Files

Item	Description
/usr/lpd/pio/etc/pioburst	Contains the pioburst command.

piocnvt Command

Purpose

Expands or contracts a predefined printer definition or a virtual printer definition.

Syntax

```
piocnvt [ -s State ] -i SourceFile [ -o TargetFile ]
```

Description

The **piocnvt** command takes either a predefined printer definition or a virtual printer definition and expands or contracts the file. An expanded printer definition file contains all the attributes associated with

that printer definition. A contracted printer definition contains only the printer specific attributes for that printer definition.

Printer definition files are arranged in a hierarchical parent-child relationship. For example the predefined printer definition `4201-3.asc` has the parent `master`. An expanded printer definition for `4201-3.asc` would contain all the attributes from `4201-3.asc` as well as those from `master`. A contracted printer definition for `4201-3.asc` would contain only the attributes not found in `master`. The **piocnvt** command simply provides a way to move back and forth between the expanded and contracted states of a printer definition file.

Flags

Item	Description
-i <i>SourceFile</i>	Specifies the complete path and name of the input file.
-o <i>TargetFile</i>	Specifies the complete path and name of the output file. If the -o flag is omitted, the <i>SourceFile</i> will be used for output.
-s <i>State</i>	Specifies whether the state of the <i>TargetFile</i> parameter should be expanded or contracted. If the -s flag is omitted, the piocnvt command attempts to determine the state by examining the zD attribute in the <i>SourceFile</i> . If a determination cannot be made the <i>TargetFile</i> parameter will be left in an expanded state. + Indicates that the state of the <i>TargetFile</i> parameter should be expanded. ! Indicates that the state of the <i>TargetFile</i> parameter should be contracted.

Examples

1. To expand the virtual printer definition `lp0:lp0` into the file `new:lp0`; enter:

```
piocnvt -s+ -i lp0:lp0 -o new:lp0
```

2. To contract the virtual printer definition `lp0:lp0` in place; enter:

```
piocnvt -s! -i lp0:lp0
```

Files

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

piodigest Command

Purpose

Digests attribute values for a virtual printer definition into a memory image and stores the memory image in a file.

Syntax

```
/usr/lpd/pio/etc/piodigest [ -s DataStreamType ] [ -n DeviceName ] [ -p DirectoryPath ] [ -q PrintQueueName ] [ -t PrinterType ] [ -d QueueDeviceName ] { ColonFileName | - }
```

Description

The **piodigest** command retrieves virtual printer attribute values from a colon file, builds a memory image of the attribute values and their lookup tables, and writes the constructed memory image to a file. The memory image in the file is then available for access by a print formatter and by the **pioibe** command (the print job manager). The command also creates Object Data Manager (ODM) stanzas for the specified queue and queue devices. The ODM stanzas are used in System Management Interface Tool (SMIT) dialogs. If an attribute called **zV** is specified and the attribute contains a value of **+**, the **piodigest** command performs syntax, reference, and limits validation on all attributes specified in the colon file.

The **piodidigest** command should be invoked whenever a customized version of a virtual printer definition is initially generated or is later modified. Each invocation of the **piodigest** command digests the attribute values for one virtual printer definition.

The *ColonFileName* parameter is the name of the input file in colon format. A colon file contains the attribute values for one virtual printer. A value of - (dash) for the *ColonFileName* parameter indicates that the colon file should be read from standard input.

The name of the output file that is generated will be of the form:

```
PrinterType.DataStreamType.DeviceName.PrintQueueName:QueueDeviceName
```

Flags

Item	Description
-d <i>QueueDeviceName</i>	Specifies the name of the virtual printer (queue device). If this flag is not specified, the virtual printer name specified by the mv attribute from the input colon file is assumed.
-n <i>DeviceName</i>	Specifies the name of the printer device, such as <code>lp0</code> for line printer 0, or <code>lp1</code> for line printer 1. If this flag is not specified, the device name specified by the mn attribute from the input colon file is assumed.
-p <i>DirectoryPath</i>	Specifies the path name of the directory where the output file is to be generated. If this flag is not specified, the <code>/var/spool/lpd/pio/@local/ddi</code> directory is assumed.
-q <i>PrintQueueName</i>	Specifies the name of the print queue to which the virtual printer is assigned. If this flag is not specified, the print queue name specified by the mq attribute from the input colon file is assumed.
-s <i>DataStreamType</i>	Specifies the printer data stream type. Example data stream types are asc (IBM extended ASCII), ps (PostScript), pcl (HP PCL), and 630 (Diablo 630). If this flag is not specified, the data stream type specified by the md attribute from the input colon file is assumed.
-t <i>PrinterType</i>	Specifies the printer type. Examples are <code>4201-3</code> and <code>ti2115</code> . If this flag is not specified, the printer type specified by the mt attribute from the input colon file is assumed.

Example

To generate a digested virtual printer definition, enter:

```
piodigest -d mypro -n lp0 -q proq -s asc -t 4201-3
```

The attribute values for the virtual printer assigned to the mypro queue device on the proq print queue are digested and stored in the file named **4201-3.asc.lp0.proq:mypro** in the **/var/spool/lpd/pio/@local/ddi** directory.

Files

Item	Description
/var/spool/lpd/pio/@local/ddi/*	Contains the digested, virtual printer definitions.
/usr/lpd/pio/etc/piodigest	Contains the piodigest command.

piodmgr Command

Purpose

Compacts the Object Data Manager (ODM) database in the **/var/spool/lpd/pio/@local/smit** directory.

Syntax

```
piodmgr { -c | -h }
```

Description

The **piodmgr** command extracts existing printer definitions from the ODM database in the **/var/spool/lpd/pio/@local/smit** directory, recreates the ODM database, compacts the database, and reloads the compacted database.

The **-c** and **-h** flags are mutually exclusive. The **-h** flag only compacts the database when the host name has been changed. The **-c** flag always compacts the database.

Note: Root user authority is needed to run this command.

Flags

Item	Description
-c	Extracts existing printer definitions from the ODM database, recreates the database, compacts the information, and replaces the database.
-h	Performs exactly like the -c flag, but the -h flag compacts the information only if the host name has been changed. If the host name has been changed, the -h flag extracts the new name and updates the host name information in the database. If the host name has not been changed, the -h flag does not compact the information. This flag is an optional compactor rather than an automatic compactor as with the -c flag.

Examples

1. To compact and update the ODM printer definition database, enter:

```
piodmgr -c
```


- To perform compaction of the information depending on whether the host name has been changed or not, enter:

```
piodmgr -h
```

Files

Item	Description
<code>/usr/lib/lpd/pio/etc/piodmgr</code>	Contains the piodmgr command.
<code>/var/spool/lpd/pio/@local/smit/*</code>	Contains predefined printer definitions used by the command.

piofontin Command

Purpose

Copies fonts from a multilingual font diskette.

Syntax

```
piofontin -t PrinterType -c Codepage [ -d Device ]
```

Description

The **piofontin** command copies font files from a multilingual font diskette to a directory one level beneath the `/usr/lib/lpd/pio/fonts` label. The directory to which the font files are copied has the name specified by the *PrinterType* parameter. The font files are named according to the naming convention for files. Names are of the form:

```
codepage.typeface.pitch*10.quality
```

Only the root user can use the **piofontin** command.

Flags

Item	Description
-c <i>Codepage</i>	Specifies the code page for the fonts. For Greek fonts the value is 851, and for Turkish fonts the value is 853.
-d <i>Device</i>	Specifies the diskette-drive device name. This defaults to the -d/dev/fd0 label, the standard 3.5-inch diskette drive.
-t <i>PrinterType</i>	Specifies the type of printer for the fonts. Supported printer types are 4201-3, 4202-3, 4207-2, 4208-2, 2380, 2381, 2390, and 2391.

Example

To read a diskette containing 4201-3 fonts in code page 851 from diskette drive `/dev/fd1`; enter:

```
piofontin 4201-3 851 /dev/fd1
```

The font files are copied to the `/usr/lib/lpd/pio/fonts/4201-3` directory.

File

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

pioformat Command

Purpose

Drives a printer formatter.

Syntax

```
/usr/lpd/pio/etc/pioformat -@ DataBaseFile [ -! FormatterName ] [ -# + PassThroughOption ]
```

Description

The **pioformat** command initiates the printer formatter driver. The formatter driver establishes access to the database values, loads and links a printer formatter, and then drives the formatter by calling its **setup** function, **initialize** function, **lineout** function, **passthru** function, and **restore** function as appropriate. The formatter driver also provides the **piogetopt** subroutine, **piogetstr** subroutine, **pioexit** subroutine used by the formatter.

The flags listed below are processed by the formatter driver and are not passed on to the formatter. However, all flags NOT listed below are assumed to be formatting flags and are passed on to the formatter.

Flags

Item	Description
<code>-@ DataBaseFile</code>	<p>Specifies either of the following:</p> <ul style="list-style-type: none">• The full path name of the (digested) database file to be accessed• The print queue and queue device names, separated by a colon <p>If the argument string begins with a / (slash) character, it is assumed to be a full path name.</p> <p>The combination of the queue name and the queue device name results in a unique string that is a part of the database file name and is used to search for the database file name in the <code>/var/spool/lpd/pio/@local/ddi</code> directory. This short form alternative is provided as a convenience when the formatter driver and formatter are run as standalone devices, instead of by the spooler.</p>
<code>-! FormatterName</code>	<p>Specifies the full path name of the formatter to be loaded, linked, and driven.</p> <p>If the <code>-!</code> flag is not specified, the default formatter name defined by the mf attribute name in the database is used. A default formatter name is provided as a convenience when the formatter driver and formatter are run as standalone devices, instead of by the spooler.</p>

Item	Description
<code>-# + PassThroughOption</code>	Specifies that the print file should be passed through unmodified. If the <code>-# +</code> flag is not specified, the print file will be formatted. The parameter that is passed to the formatter's setup routine contains a value of 1 instead of 0, indicating that the file should be passed through instead of being formatted.

Examples

1. To format the `myfile` file according to the database file (virtual printer description) for the queue device named `std` associated with the print queue named `pro`, overriding the page width to 132 characters, and using the **pioformat** command and a formatter as a standalone filter, enter:

```
cat myfile | pioformat -@ pro:std -w 132 >/dev/lp0
```

2. To use the **pioformat** command and a formatter in a pipeline running under the spooler, enter:

```
%Ide/pioformat -@ %Idd/%Imm -! %Idf/piof420x %Fbb %Fee ...
```

For this example, assume that:

- The printer is a 4207 Model 2 Proprinter.
- The print queue name is `pro`.
- There is only one queue device (virtual printer) defined for the print queue and its name is `std` and its output data stream type is `asc` (extended ASCII).
- The printer device name is `/dev/lp0`.
- The print job submitter specified the flag and argument `-i 5`.

Before the print job manager (the **piobe** command) passes the pipeline to a shell to format the file, it resolves the pipeline's embedded references to attribute values. Based on the assumptions listed above for this example, the attribute references would be resolved as:

Item	Description
<code>%Ide -> /usr/lpd/pio/etc</code>	Directory where the pioformat command resides
<code>%Idd -> /var/spool/lpd/pio/@local/ddi</code>	Directory for database files
<code>%Imm -> 4207-2.asc.lp0.pro:std</code>	Database file name
<code>%Idf -> /usr/lpd/pio/fmtrs</code>	Directory for formatters
<code>%Fbb -></code>	Null string, since submitter did not specify the -b flag
<code>%Fee -> -i 5</code>	Submitter specified this flag and argument.

The resulting pipeline shown below would be passed to a shell to format the file (shown on multiple lines for readability):

```
/usr/lpd/pio/etc/pioformat      # initiate the formatter driver
-@/usr/lpd/pio/ddi/4207-2.asc.lp0.pro:std
                                # (digested) database file
-!/usr/lpd/pio/fmtrs/piof420x  # loadable formatter
-i5                             # formatting option
                                # (indent 5 characters)
```

Files

Item	Description
<code>/usr/lpd/pio/etc/pioformat</code>	Contains the formatter driver.
<code>/usr/lpd/pio/fmtrs/*</code>	Contains the formatters.
<code>/var/spool/lpd/pio/ @local/ddi/*</code>	Contains the digested database files.

piofquote Command

Purpose

Converts certain control characters destined for PostScript printers.

Syntax

`/usr/lpd/pio/etc/piofquote`

Description

The **piofquote** command is a filter that converts certain control characters destined for PostScript printers that can emulate other printers. The command reads data from standard input, checks for control characters, and modifies them as needed. It then writes the data to standard output.

If a least 1 byte of data appears on standard input, the **piofquote** command writes a hex 04 control character to standard output before the first input data byte is written to standard output. The command also writes a hex 04 to standard output when end-of-file is recognized on standard input.

If a hex 01, 03, 04, 05, 11, 13, 14, or 1c control character is found in the input data read from standard input, the hex 40 bit in the control character is turned on and a hex 01 character is prefixed to the control character before it is written to standard output.

Files

Item	Description
standard input	Input data stream to be processed.
standard output	Output data stream containing converted control characters.

piolsvp Command

Purpose

Lists virtual printers on a system.

Syntax

`piolsvp { -q | -v | -Q | -p | -A } [-nAttachmentField]`

`piolsvp -P Queue [: QueueDevice] -nAttachmentField`

`piolsvp -P Queue -d`

`piolsvp -N AttachmentType -nAttachmentField`

Description

The **piolsvp** command lists the virtual printers and attachment types on the system. The **piolsvp** command displays either the queues or the queues plus the queue-device pairs for virtual printers.

The order of the list of queues and queue-device pairs is the same as the order used by the **/etc/qconfig** file.

Flags

Item	Description
-A	Displays all attachment types and descriptions for the attachment types. The .attach and .config files in the /usr/lib/lpd/pio/etc directory define all attachment types.
-d	Displays the queue devices associated with a given queue.
-nAttachmentField	<p>Specifies a field name for an attachment. The field name is typically a SMIT selector name. Possible values for the <i>AttachmentField</i> variable are:</p> <ul style="list-style-type: none">submit_jobadd_queueadd_printerremove_queueprinter_connchange_queuechange_filters <p>When the -n and -A flags are specified, only the attachment types that have a value for the specified attachment field in their attachment files are displayed. Attachment definitions are kept in the files with the <i>AttachmentType.attach</i> naming convention. The .attach files reside in the /usr/lib/lpd/pio/etc directory.</p> <p>When the -n flag is specified with either the -q or -v flags, only queues and queue-device pairs that belong to defined attachment types are displayed. A defined attachment type has an assigned field value in the definition files.</p> <p>When the -n flag is specified with the -P flag, the SMIT selector name is displayed. The -n and -P flag combination also displays the queue device name and attachment type.</p> <p>When the -n flag is specified with the -N flag, the SMIT selector name is displayed for the specified attachment field and attachment type.</p>
-N	Specifies an attachment type. The SMIT selector name associated with a given attachment field is displayed.
-p	Displays all the queue and queue-device pairs on the system and provides a description of each queue and queue-device pair. Only the queue name for the first queue-queue is displayed if there are queues with multiple queue devices.
-P	Specifies the queue name or queue device name for which information is displayed. The information consists of queue device name, attachment type, and SMIT selector value name.
-q	Displays all queues on the system. The -q flag also displays the queue-device pairs for queues that have more than one device.

Item	Description
-Q	Displays all the queues on the system. The -Q flag does not list queue-device pairs. Use the -q flag to list queue-device pairs.
-v	Displays all queue-device pairs for the queues that have virtual printers.

Examples

1. To display all the print queues on the system, enter:

```
piolsvp -q
```

The output of this command is:

```
e4019a      4019 (IBM ASCII)
d3816      IBM 3816 Page Printer
ena_asc    4029 (IBM ASCII)
ena_gl     4029 (Plotter Emulation)
ena_pcl    4029 (HP LaserJet II Emulation)
ena_ps     4029 (PostScript)
hplj2     Hewlett-Packard LaserJet II
tstx      4216-31 (Proprinter XL Emulation)
e4019ps   4019 (PostScript)
4019lxxa  4029 (PostScript)
4019lxxa:lxx 4029 (PostScript)
4019lxxa:rkmlxx 4019 (IBM ASCII)
4019lxxa:rkmlxxl 4019 (IBM ASCII)
```

2. To display all the virtual printers in the system, enter:

```
piolsvp -v
```

The output of this command is:

```
#QUEUE      DEVICE      DESCRIPTION
e4019a      e4019      4019 (IBM ASCII)
d3816      ena3816    IBM 3816 Page Printer
ena_asc    ena        4029 (IBM ASCII)
ena_gl     ena        4029 (Plotter Emulation)
ena_pcl    ena        4029 (HP LaserJet II Emulation)
ena_ps     ena        4029 (PostScript)
hplj2     lxx        Hewlett-Packard LaserJet II
tstx      lxx        4216-31 (Proprinter XL Emulation)
e4019ps   e4019      4019 (PostScript)
4019lxxa  lxx        4029 (PostScript)
4019lxxa  rkmlxx     4019 (IBM ASCII)
4019lxxa  rkmlxx     4019 (IBM ASCII)
```

3. To list all the queues on the system, enter:

```
piolsvp -Q
```

The output of this command is:

```
e4019a      4019 (IBM ASCII)
d3816      IBM 3816 Page Printer
ena_asc    4029 (IBM ASCII)
ena_gl     4029 (Plotter Emulation)
ena_pcl    4029 (HP LaserJet II Emulation)
ena_ps     4019 (PostScript)
hplj2     Hewlett-Packard LaserJet II
tstx      4216-31 (Proprinter XL Emulation)
e4019ps   4019 (PostScript)
4019lxxa  4029 (PostScript)
```

4. To list all the attachment types that have a SMIT selector value specified for the add_queue SMIT selector, enter:

```
piolsvp -A -nadd_queue
```

The output from this command is:

```
#ATTACHMENT TYPE      DESCRIPTION
local                 Local Attached
remote               Remote Attached
ascii                 ASCII Terminal Attached
other                 Generic Backend Attached
```

5. To list information for the 40191xxa queue, enter:

```
piolsvp -P40191xxa -n add_queue
```

The output from this command is:

```
lxx      xsta      sm_xsta_addq_sel
```

6. To list the SMIT selector value for the remote attachment, enter:

```
piolsvp -Axst -nadd_queue
```

The output from this command is:

```
sm_xsta_addq_sel
```

Files

Item

/usr/lib/lpd/pio/etc/piolsvp

/etc/qconfig

/var/spool/lpd/pio/@local/custom/*

/usr/lib/lpd/pio/etc/*.attach

Description

Contains the **piolsvp** command.

Contains the configuration files.

Contains the customized virtual printer attribute files.

Contains the attachment type files

piomgpdev Command

Purpose

Manages printer pseudo-devices.

Syntax

```
piomgpdev -p PseudoDevice -t AttachmentType { -A | -C | -R | -D } [ -a Clause ... ]
```

Description

The **piomgpdev** command changes and removes pseudo-devices for printer attachments. The **piomgpdev** command stores information about the pseudo-devices in files in the **/var/spool/lpd/pio/@local/dev** directory. The file contains stanzas in the following form:

```
key_word = value
```

The information stored in these files pertains to connection characteristics for a given attachment and a printer.

Flags

Item	Description
-a <i>Clause</i>	Specifies a clause to be added or changed in the file for a pseudo-device. The clause is in the following form: <pre>key_word = value</pre> If the -D flag is specified, the clause can contain only the keyword.
-A	Adds a pseudo-device.
-C	Changes a pseudo-device.
-D	Displays information for a specified clause of a pseudo-device definition.
-p <i>PseudoDevice</i>	Specifies the name of a pseudo-device for a printer attachment.
-R	Removes a pseudo-device.

Files

Item	Description
<code>/usr/lib/lpd/pio/etc/piomgpdev</code>	Contains the piomgpdev command.
<code>/var/spool/lpd/pio/@local/dev/*</code>	Contains the printer pseudo-device files.

piomkapqd Command

Purpose

Builds a SMIT dialog to create print queues and printers.

Syntax

To Create a Print Queue for an Existing Printer

```
piomkapqd -A AttachmentType -p Printer -d DeviceName -h Header [ -e ]
```

To Create a Printer and a Print Queue

```
piomkapqd -A AttachmentType -p Printer -v Device -s Subclass -r Adapter -h Header [ -e ]
```

To Create a Printer Attached to a TTY or to Assign Printer Output to a File and Create a New Queue

```
piomkapqd -A AttachmentType -p Printer { -T TTYName | -f FileName } -h Header [ -e ]
```

To Use a User-Defined Attachment for a New Printer and Print Queue

```
piomkapqd -A AttachmentType -p Printer [ -d DeviceName ] -c CmdExec -i DiscCmd -o ObjectID -h Header [ -e ]
```

Description

The **piomkapqd** command creates a System Management Interface Tool (SMIT) dialog that allows the user to create new printers and print queues. The **piomkapqd** command also allows users to add their user-defined attachment types to a SMIT printer or queue definition dialog.

Flags

Item	Description
-A <i>AttachmentType</i>	Specifies the type of attachment used to connect the printer to the data source. Common values for the <i>AttachmentType</i> variable are: local Specifies a local attachment type. ascii Specifies an ASCII attachment type. file Specifies a file where the data is stored.
-c <i>CmdExec</i>	Specifies the value for the cmd_to_execute SMIT command. This flag is used when creating a user-defined attachment dialog. If this flag is not included, the piomkppq command is used as the default.
-d <i>DeviceName</i>	Specifies the name of the device, pseudo-device, or file where the output is directed, for example <code>lp0</code> or <code>tty1</code> .
-e	Specifies that an existing print queue is to be used for printer output. The -e prevents the piomkapqd command from creating a new queue.
-f <i>FileName</i>	Indicates the name of the file where output is stored.
-h <i>Header</i>	Specifies the title or header of the SMIT dialog that is being created.
-i <i>DiscCmd</i>	Specifies the value of the cmd_to_discover SMIT command. This flag is used when creating a user-defined attachment dialog. If this flag is not included, the piomkapqd command default value is used to create the dialog.
-o <i>ObjectID</i>	Specifies the SMIT object whose ID matches the value of the <i>ObjectID</i> variable.
-p <i>Printer</i>	Specifies the printer type as defined in the <code>/usr/lib/lpd/pio/predef</code> directory, for example <code>ibm4019</code> .
-r <i>ParentAdapter</i>	Specifies the parent adapter for the printer.
-s <i>Subclass</i>	Specifies the subclass type to which the printer belongs. The possible values for the <i>Subclass</i> variable are: <ul style="list-style-type: none">• parallel• rs232• rs422
-T <i>TTYName</i>	Specifies the name of the TTY attached to the new printer or queue.
-v <i>Device</i>	Specifies the device type as defined in the ODM database. The -v flag retrieves printer definitions that are not stored in the <code>/usr/lib/lpd/pio/predef</code> directory.

Examples

1. To create a SMIT dialog that adds a print queue to an existing local printer, enter:

```
piomkapqd -A local -p ibm4019 -d lp0 -h 'Add a New Queue'
```

2. To create a SMIT dialog that adds a new printer named `lp2` and new print queue attached locally, enter:

```
piomkapqd -A local -p ibm4019 -v ibm4019 -s rs232 -r sa0 -h 'Add New Printer'
```

3. To create a SMIT dialog that adds a printer attached to a TTY and create a new queue for the printer, enter:

```
piomkapqd -A tty -p ibm4039 -T tty12 -h 'Add TTY Printer'
```

4. To create a SMIT dialog that directs output to a file name `stuff` and to create a new queue, enter:

```
piomkapqd -A file -p ibm4039 -f stuff -h 'Add Output File' -e
```

5. To create a SMIT dialog that adds a user-defined printer attachment type and creates a new queue, enter:

```
piomkapqd -A hpJetDirect -p hplj-4 [-d lp0] -c /usr/sbin/mkjetd -i /usr/bin/ljsd -o  
JetDirect -h  
'Add New Attachment Type'
```

File

Item	Description
<code>/usr/lib/lpd/pio/etc/piomkapqd</code>	Contains the piomkapqd command.

piomkpq Command

Purpose

Creates a print queue.

Syntax

To add a new printer

```
piomkpq -A AttachmentType -p PrinterType -Q QueueName -D DataStream -v DeviceType -s Subclass  
-r ParentAdapter -w PortNumber [ -a { interface | ptop | autoconfig | speed | parity | bpc | stops | xon | dtr |  
tbc=DescValue } ] ...
```

To create a new print queue

```
piomkpq -A AttachmentType -p PrinterType { -D DataStream / -q QueueName } -s Subclass  
-r ParentAdapter -w PortNumber -v DeviceType [ -a { interface | ptop | autoconfig | speed | parity | bpc |  
stops | xon | dtr | tbc=DescValue } ] ...
```

To create print queues for an existing printer

```
piomkpq -A AttachmentType -p PrinterType -d DeviceName { -D DataStream / -q QueueName }
```

To add an existing printer to an existing print queue

```
piomkpq -A AttachmentType -p PrinterType -d DeviceName -D DataStream -q QueueName
```

Description

The **piomkpq** command creates print queues and printers. This command is used by SMIT dialogs created with the **piomkapqd** command. The **piomkpq** command performs the following functions:

- Creates printer devices with various attachment types.
- Creates print queues.
- Creates queue devices.
- Creates virtual printers.

- Creates pseudo-devices.

Flags

Item	Description
-a	Specifies a device attribute. This takes the form <i>Attribute=Value</i> , for example: <code>-a speed=9600</code> . The valid attributes are: Interface ptop autoconfic speed parity bpc stops xon dtr tbc
-A AttachmentType	Specifies the type of attachment used to connect the printer to the data source. Common values for the <i>AttachmentType</i> variable are: local Specifies a local attachment type. ascii Specifies an ASCII attachment type. file Specifies a file where the data is stored.
-d DeviceName	Specifies the name of the device, pseudo-device, or file where the output is directed, for example <code>lp0</code> or <code>tty1</code> .
-D DataStream	Specifies the datastream of a print queue to be created or an existing print queue.
-p PrinterType	Specifies the printer type as defined in the <code>/usr/lib/lpd/pio/predef</code> directory, for example <code>ibm4019</code> .
-q QueueName	Specifies a new queue name. The -q and -Q flags are exclusive.
-Q QueueName	Specifies an existing queue name. The -q and -Q flags are exclusive.
-s Subclass	Specifies the subclass type to which the printer belongs. The possible values for the <i>Subclass</i> variable are: <ul style="list-style-type: none"> • parallel • rs232 • rs422
-r ParentAdapter	Specifies the parent adapter for the printer.
-w PortNumber	Specifies the port number for the printer attachment.
-v DeviceType	Specifies the device type as defined in the ODM database.

Examples

1. To create a local print queue named `castor` of datastream ASCII for an existing IBM 4019 printer named `lp0`, enter:

```
piomkpg -A local -p ibm4019 -d lp0 -D asc -q castor
```

2. To add an existing local printer to an existing local print queue called `pyrite` for the datastream PostScript, enter:

```
piomkpg -A local -p ibm4019 -d lp0 -Q pyrite -D ps
```

3. To create local print queue called `baker` for a new printer, enter:

```
piomkpg -A local -p ibm4019 -D asc -Q baker -s parallel -r ppa0  
-w p -v ibm4019 [-a ptop=120]
```

4. To create the **clues** file print queue, enter:

```
piomkpg -A file -p ibm4019 -d clues -D asc -q baker
```

Files

Item	Description
<code>/usr/lib/lpd/pio/etc/piomkpg</code>	Contains the piomkpg command.
<code>/usr/lib/lpd/pio/etc/piomgpdev</code>	Creates a pseudo-device.
<code>/usr/sbin/mkdev</code>	Creates a device.
<code>/usr/bin/mkque</code>	Creates a queue.
<code>/usr/bin/mkquedv</code>	Creates a queue device.
<code>/usr/sbin/mkvirprt</code>	Creates a virtual printer.

piomsg Command

Purpose

Sends a printer backend message to the user.

Syntax

```
piomsg [ -u UserList ] [ -c MsgCatalog [ -s MsgSet ] -n MsgNumber ] [ -a MsgArg ] ... [ MsgText ]
```

Description

The **piomsg** command either retrieves a printer backend message from a message catalog or sends a specified message text to one or more users. The **piomsg** command runs when a print job is executed. Typically, the **piomsg** command is used in printer colon files to send a message to the user submitting a print job while the print job is processed by the **pio** command.

When the **-c**, **-s**, or **-n** flags are specified, the **piomsg** command retrieves a message from a message catalog. The command searches for the message in the directory specified in the **NLSPATH** environment variable. If the **NLSPATH** environment variable does not contain a directory path, the **piomsg** command searches the `/usr/lib/lpd/pio/etc` default directory. If no message is found in the `/usr/lib/lpd/pio/etc` directory, the command supplies the text specified in the *MsgText* variable. When the **-c**, **-s**, or **-n** flags are not specified, the **piomsg** command returns the value (if any) of the *MsgText* variable.

Each message is parsed for the **%s** or **%n\$s printf** subroutine conversion specifications. The **printf** conversion specifications are replaced with supplied message strings, if any, before the message is sent to the user. The **piomsg** command processes escape sequences, such as, linefeed **/n** or horizontal tab **/t**, that are embedded in the message.

Flags

Item	Description
-a <i>MsgArg</i>	Specifies the message argument string. The value of the <i>MsgArg</i> variable is substituted into the message, if it contains the %s or %n\$s printf subroutine conversion specifications. The -a flag can be specified up to 10 times to specify multiple arguments. If there are any errors while parsing conversion specifications, the original message is sent.
-c <i>MsgCatalog</i>	Specifies the message catalog that contains the message to be retrieved. The -c flag must be specified with the -n flag.
-n <i>MsgNumber</i>	Specifies the message number. The -n flag must be specified with the -c flag.
-s <i>MsgSet</i>	Specifies an optional message set. The default value for the <i>MsgSet</i> variable is 1. The -s flag must be specified with both the -c and -n flags.
-u <i>UserList</i>	Specifies the list of users who receive the message. The names of users or nodes in the <i>UserList</i> variable are separated by commas. To include a node name in the user list specify the @ character followed by a node name or address. If the -u flag is omitted, the message returns to the user who initiated the print job.

Examples

1. To retrieve message number 100 in message set number 1 from the `piobe.cat` message catalog and send the message to user `joe` on the same node as the print server and `tom` on node `foobar`, enter:

```
piomsg -u joe,tom@foobar -c piobe.cat -n 100
```

2. To send a message with a message argument string to the user who submitted the print job, enter:

```
piomsg -a "/usr/bin/troff" "The specified filter %s is not found\n"
```

3. To retrieve message number 5 in set number2 from the `xyz.cat`, use a dummy message in the event of a failure, and send the message to the printer, enter:

```
piomsg -cxyz.cat -s2 -n5 "xyz.cat is not installed.\n"
```

Note: When the **piomsg** command cannot retrieve messages from the catalog specified with the **NLSPATH** environment variable or the default directory, the supplied message text is sent to the users.

File

Item	Description
<code>/usr/lib/lpd/pio/etc/piomsg</code>	Contains the piomsg command.

pioout Command

Purpose

Printer backend's device driver interface program.

Syntax

```
/usr/lpd/pio/etc/pioout [ -A BytesPrinted ] [ -B TotalBytes ] [ -C NumberCancelStrings ]  
[ -D CancelString ] [ -E Mask ] [ -F FormFeedString ] [ -I InterventionRequiredUser ] [ -K TextString ]  
[ -L TextString ] [ -N NumberFormFeedStrings ] [ -O OutFile ] [ -P PrefixFile ] [ -R ParseRoutine ]  
[ -S SuffixFile ] [ -W+ ]
```

Description

The **pioout** command is at the end of pipelines invoked by the **piobe** command (the print job manager) to print a file or a burst page on a printer. It reads input data from standard input, the prefix file (if the **-P** flag is specified), and the suffix file (if the **-S** flag is specified), and then writes the data to the printer (or *OutFile*, if the **-O** flag is specified). Error conditions and situations where intervention is required (unless the **-I** flag is specified) are reported to the user who submitted the print job.

The values specified with the **-A** flag and the **-B** flag are used to periodically report to the **qdaemon** process the percentage of the print job that has completed. The **-C** flag and the **-D** flag specify the data string sent to the printer if the print job is canceled.

The **-O** flag is used to generate a header page and store it in a temporary file. The **-P** flag is then used to print the header page (that was saved in a temporary file) just prior to printing the print file.

The **pioout** command requires the following environment variables to be initialized:

Item	Description
PIOTITLE	Title of the print job
PIODEVNAME	Device name
PIOQNAME	Print queue name
PIOQDNAME	Queue device name
PIOFROM	User who submitted the print job
PIOMAILONLY	If nonzero, message to user should always be mailed, not displayed.
PIOTERM	Overrides the terminal type assumed from the tty definition. This variable is only used for print jobs submitted to terminal-attached terminals.

Flags

Item	Description
-A BytesPrinted	Specifies the number of bytes already printed for the print job.
-B TotalBytes	Specifies the total number of bytes to be printed for the print job.
-C NumberCancelStrings	Specifies the number of times the string specified by the -D flag is to be sent to the printer when a print job is canceled. If this flag is not specified, the value is assumed to be 3168.
-D CancelString	Specifies the string to be sent to the printer when a print job is canceled. If the -D flag is not specified, the string is assumed to consist of 1 null character.
-E Mask	Specifies, as <i>Mask</i> , one or more device-driver error-flag names, separated by commas. If the mask is one returned by the ioctl subroutine with an LPQUERY command, the error condition indicated by the mask is ignored. Flag names can include LPST_ERROR , LPST_NOSLCT , and LPST_SOFT , and are defined in the /usr/include/sys/lpio.h file.

Item	Description
-F <i>FormFeed String</i>	Specifies the string to be sent to the printer to cause a form feed. If the -F flag is not specified, the string is assumed to be <code>\014</code> .
-I <i>InterventionRequiredUser</i>	Specifies the user to whom a message is to be sent when the printer requires intervention. If this flag is not specified, the message is sent to the user who submitted the print job. The <i>InterventionRequiredUser</i> parameter can be one or more user names, separated by commas. A null string represents the print job submitter. For example, the string <code>,jim@server02</code> causes intervention required messages to be sent to both the print job submitter and to user <code>jim</code> at node <code>server02</code> .
-K <i>TextString</i>	Specifies that messages sent by a PostScript printer will be discarded if they contain the specified text string. For example, if the <i>TextString</i> variable is <code>warming up</code> , messages that include the text <code>warming up</code> will be discarded.
-L <i>TextString</i>	Specifies that if a message received from a PostScript printer includes the specified text string, the text following this text string in the message will be sent to the intervention-required user specified by the -I flag.
-N <i>NumberFormFeedStrings</i>	Specifies the number of form-feed strings to be sent to the printer at the end of the input data stream. If this flag is not specified, the value is assumed to be zero. This flag is normally used only to align continuous forms after the printer has been idle, or to feed forms when the printer goes idle.
-O <i>OutFile</i>	Specifies that the output is sent to the specified file instead of being sent to the printer.
-P <i>PrefixFile</i>	Specifies the file sent to the printer before the first byte of the print file is sent. If the print job terminates before the first byte of the print file arrives, the prefix file is not sent.
-R <i>ParseRoutine</i>	Specifies the full path name of a routine to parse data read from the printer. An example of a parse routine is contained in the <code>/usr/include/piostruct.h</code> file. If the -R flag is not specified, a default parse routine is used.
-S <i>SuffixFile</i>	Specifies the file sent to the printer after the print file has been sent. If the print job terminates before the first byte of the print file arrives, the suffix file is not sent.
-W +	Specifies that EOF (hex 04) must be received from the printer in order to exit.

piopredef Command

Purpose

Creates a predefined printer data-stream definition.

Syntax

piopredef [**-r**] **-d** *QueueDeviceName* **-q** *PrintQueueName* **-s** *DataStreamType* **-t** *PrinterType*

Description

The **piopredef** command creates a predefined printer data-stream definition from a virtual printer definition. It can be thought of as the inverse of the **mkvirprt** command, displayed with the **chvirprt** command, and then specified with the **piopredef** command to create a predefined definition for the unsupported printer.

The new predefined printer definition can then be specified with a **mkvirprt** command to generate additional virtual printers for the unsupported printer type on the same computer, or transported to other computers and used there.

Flags

Item	Description
-d <i>QueueDeviceName</i>	Specifies with the <i>QueueDeviceName</i> variable the spooler of the customized virtual printer definition to be used to create the predefined printer definition.
-q <i>PrintQueueName</i>	Specifies with the <i>PrintQueueName</i> variable the spooler of the virtual printer definition to be used to create the predefined printer definition.
-r	Specifies that if the -s flag and the -t flag specify a predefined printer definition that already exists, the existing one should be replaced.
-s <i>DataStreamType</i>	Specifies with the <i>DataStreamType</i> variable the printer for the predefined printer definition to be created. Example data stream types are: asc IBM extended ASCII gl Hewlett-Packard GL pcl Hewlett-Packard PCL ps PostScript 630 Diablo 630 855 Texas Instruments 855.
-t <i>PrinterType</i>	Specifies the printer type for the predefined printer definition to be created. Examples of existing printer types are: 4201-3, hplj-2, ti2115, and so on.

Note: If no flags are specified, the command syntax is displayed.

Example

To create a new predefined printer definition from an existing virtual printer definition for the virtual printer, enter:

```
piopredef -d mypro -q proq -s asc -t 9234-2
```

The attributes for the virtual printer assigned to the mypro queue device on the proq print queue are copied to create a new predefined printer definition for the 9234-2 printer (asc data stream).

Files

Item	Description
<code>/etc/piopredef</code>	Contains the piopredef command.
<code>/usr/lpd/pio/predef/*</code>	Predefined printer data stream attribute files. File names are in the format: <code>PrinterType.DataStreamType</code> .
<code>/var/spool/lpd/pio/@local/custom/*</code>	Customized virtual printer attribute files. File names are in the format: <code>PrintQueueName:QueueDeviceName</code> .

pkgadd Command

Purpose

Transfers a software package or set to the system.

Syntax

To Install a Software Package

```
pkgadd [ -d Device ] [ -r Response ] [ -n ] [ -a Admin ] [ -P Path ] [ Pkginst1 [ Pkginst2 [ . . . ] ] ]
```

To Copy a Software Package to the Specified Spool Directory

```
pkgadd -s Spool [ -d Device ] [ Pkginst1 [ Pkginst2 [ . . . ] ] ]
```

Description

pkgadd transfers the contents of a software package or set from the distribution medium or directory to install it onto the system. A package is a collection of related files and executables that can be independently installed. A set is made up of a special-purpose package, referred to as a Set Installation Package (SIP), and a collection of one or more packages that are members of the set. The SIP controls the installation of the set.

pkgadd checks that all packages listed on the command line are on the installation medium. If any of the packages listed does not exist, no changes are made to the system, that is, none of the listed packages are installed.

Note: Nonroot users must meet the following conditions to run the **pkgadd** command successfully:

1. Users must have write permission to the paths specified in the `pkgmap` file.
2. The current `user:group` must match the `user:group` specified in the `pkgmap` file.
3. Users must have write permissions on the `/var/sadm/install` and `/var/sadm/pkg` directories.

Used without the **-d** flag, **pkgadd** looks in the default spool directory for the package (`/var/spool/pkg`). Used with the **-s** flag, it writes the package to a spool directory instead of installing it.

Error messages are always logged. In addition, when **pkgadd** terminates, it sends mail (by default, to "root") with all the error messages and a summary of which packages installed completely, partially, or not at all.

Flags

Item	Description
-d <i>Device</i>	Installs or copies a package/set from <i>Device</i> . <i>Device</i> can be the full pathname to a directory, file or named pipe, or "-" which specifies packages in datastream format read from standard input. The default device is the installation spool directory (/var/spool/pkg).
-r <i>Response</i>	Identifies a file or directory, <i>Response</i> , which contains the answers to questions posed by a "request script" during a previous pkgask session conducted in interactive mode [see the pkgask command]. When <i>Pkginst</i> is a package, <i>Response</i> can be a full pathname or a directory; when <i>Pkginst</i> is a SIP, <i>Response</i> must be a directory.
-n	Specifies that installation runs in non-interactive mode. The default mode is interactive.
-a <i>Admin</i>	Defines an installation administration file, <i>Admin</i> , to be used in place of the default administration file to specify whether installation checks (such as the check on the amount of space, the system state, and so on) are done. The token "none" overrides the use of any admin file, and thus forces interaction with the user. Unless a full pathname is given, pkgadd looks in the /var/sadm/install/admin directory for the file. By default, the file default in that directory is used. default specifies that no checking is done, except to see if there is enough room to install the package and if there are dependencies on other packages. The -a flag cannot be used if <i>Pkginst</i> is a SIP.
-P <i>Path</i>	Specifies an alternative root directory path for installation. Files will be installed under this location.
<i>Pkginst</i>	Defines a short string used to designate an abbreviation for the package/set name. (The term "package instance" is used loosely: it refers to all instantiations of <i>Pkginst</i> .) See the pkginfo command and the pkginfo file format. If <i>Pkginst</i> is a SIP, the SIP controls installation of the set by using request scripts and pre-install scripts. The SIP request script, not the package installation tools, is responsible for prompting the user for responses and taking the appropriate actions. If the request script fails, only the SIP is processed. To indicate all instances of a package, specify ' <i>Pkginst</i> .*', enclosing the command line in single quotes, as shown, to prevent the shell from interpreting the "*" character. Use the token "all" to refer to all packages available on the source medium.
-s <i>Spool</i>	Reads the package into the directory <i>Spool</i> instead of installing it.

Special Notes

The **-r** flag can be used to indicate a directory name as well as a filename. The directory can contain numerous *Response* files, each sharing the name of the package with which it should be associated. This would be used, for example, when adding multiple interactive packages with one invocation of **pkgadd**. Each package that had a request script would need a *Response* file. If you create response files with the same name as the package (for example, *Package1* and *Package2*) then, after the **-r** flag, name the directory in which these files reside.

The **-n** flag causes the installation to halt if any interaction is needed to complete it.

When invoked with no *Pkginst* specified on the command line, **pkgadd** only displays the names of sets if at least one SIP exists on the media. Because of this, you shouldn't include packages on the same media if some are members of sets and some are not. If you do, the packages which are not members of sets can be installed only if their **pkginst** names are provided on the command line.

The **pkgadd** command checks to see if any of the files in *Pkginst* are already installed on the system and, if any are, saves this fact before continuing with installation. Later, **pkgadd** does not reinstall these files

on the system. If one of the packages installation scripts removes such a file, the result is that the file will no longer be on the system when package installation completes.

The **pkgadd** command does not uncompress any files that were already compressed (that is, only those in ".Z" form) before being processed by **pkgmk**.

Exit Status

This command returns the following exit values:

Item	Description
------	-------------

- | | |
|----|--|
| 0 | Successful completion of script. |
| 1 | Fatal error. Installation process is terminated at this point. |
| 2 | Warning or possible error condition. Installation continues. A warning message is displayed at the time of completion. |
| 3 | Script was interrupted and possibly left unfinished. Installation terminates at this point. |
| 4 | Script was suspended (administration). Installation terminates at this point. |
| 5 | Script was suspended (interaction was required). Installation terminates at this point. |
| 10 | System should be rebooted when installation of all selected packages is completed. (This value should be added to one of the single-digit exit codes described above.) |
| 20 | The system should be rebooted immediately upon completing installation of the current package. (This value should be added to one of the single-digit exit codes described above.) |
| 77 | No package was selected for the set. |
| 99 | Internal error. |

Files

Item	Description
<code>/var/sadm/install/admin/default</code>	default package administration file
<code>/var/sadm/install/logs/pkginst.log</code>	error message log
<code>/var/spool/pkg</code>	default spool directory

pkgask Command

Purpose

Stores answers to a request script.

Syntax

```
pkgask [ -d Device] -r Response [ Pkginst [ Pkginst [ . . . ] ] ]
```

Description

pkgask enables an administrator to store answers to an interactive package (one with a request script) or a set of packages. A set is made up of a special-purpose package, referred to as a Set Installation Package (SIP), and a collection of one or more packages that are members of the set. The SIP controls the installation of the set.

Invoking **pkgask** generates a *Response* file that is then used as input at installation time. The use of this *Response* file prevents any interaction from occurring during installation since the file already contains all of the information the package needs.

When **pkgask** runs, it creates the response file as well as the following directories:

Item	Description
/ptfvars	Contains variables pertaining to the package.
/fileinfo	Contains checksum information about the package.
/oldfiles	Contains backups of previous versions of the package.

To install the package on another system non-interactively, you must copy all of these files and directories to the target system.

Note: If you overwrite any of these directories, for example, to install another package non-interactively, you will not be able to successfully remove the first package unless you restore the original directory contents first.

You can use the **-r** flag to indicate a directory name as well as a filename. The directory name is used to create numerous *Response* files, each sharing the name of the package with which it should be associated. This is useful, for example, when you add multiple interactive packages with one invocation of **pkgadd**. Each package needs a *Response* file. To create multiple response files with the same name as the package instance, name the directory in which the files should be created and supply multiple instance names with the **pkgask** command. When installing the packages, you can identify this directory to the **pkgadd** command.

Flags

Item	Description
-d Device	Runs the request script for a package on <i>Device</i> . <i>Device</i> can be the full pathname to a directory (such as /var/tmp), or "-" which specifies packages in datastream format read from standard input. The default device is the installation spool directory (/var/spool/pkg).
-r Response	Identifies a file or directory, <i>Response</i> , which should be created to contain the responses to interactions with the packages request script. The file, or directory of files, can later be used as input to the pkgadd command [see the pkgadd command]. When <i>Pkginst</i> is a package, <i>Response</i> can be a full pathname or a directory; when <i>Pkginst</i> is a SIP, <i>Response</i> must be a directory.

Item	Description
<i>Pkginst</i>	<p>Defines a short string used to designate an abbreviated package/set name. (The term "package instance" is used loosely: it refers to all instantiations of <i>Pkginst</i>, even those that do not include instance identifiers.)</p> <p>To create a package name abbreviation, assign it with the "PKG" parameter. For example, to assign the abbreviation "cmds" to the Advanced Commands package, enter PKG=cmds.</p> <p>If <i>Pkginst</i> specifies a SIP, all request scripts for packages which are members of that set are run (if any) and the resulting response files are placed in the directory provided to the -r flag.</p> <p>To indicate all instances of a package, specify '<i>Pkginst.*</i>', enclosing the command line in single quotes, as shown, to prevent the shell from interpreting the "*" character. Use the token "all" to refer to all packages available on the source medium.</p> <p>Note: When invoked with no <i>Pkginst</i> specified on the command line, pkgask only displays the names of sets if at least one SIP exists on the device. Thus, if you have packages which are not members of sets, they can be referenced only if their <i>Pkginst</i> names are provided on the command line.</p>

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion of script.
1	Fatal error. Installation process is terminated at this point.
2	Warning or possible error condition. Installation continues. A warning message is displayed at the time of completion.
3	Script was interrupted and possibly left unfinished. Installation terminates at this point.
4	Script was suspended (administration). Installation terminates at this point.
5	Script was suspended (interaction was required). Installation terminates at this point.
10	System should be rebooted when installation of all selected packages is completed. (This value should be added to one of the single-digit exit codes described above.)
20	The system should be rebooted immediately upon completing installation of the current package. (This value should be added to one of the single-digit exit codes described above.)
77	No package was selected for the set.
99	Internal error.

Files

Item	Description
/var/spool/pkg	default spool directory

pkgchk Command

Purpose

Checks the accuracy of an installation.

Syntax

To Check the Contents of Installed Objects

```
pkgchk [-l | -a -c -f -q -v] [-n -x] [-P path] [-p Path1[,Path2 ...] [-i File] [Pkginst ...]
```

To Check the Contents of a Package Spooled on a Specified Device

```
pkgchk -d Device [-l | -v] [-p Path1[,Path2 ...] [-i File] [Pkginst ...]
```

To Check the Contents of a Package Described in the Specified pkgmap

```
pkgchk -m Pkgmap [-e Envfile] [-l | -a -c -f -q -v] [-n -x] [-i File] [-p Path1[,Path2 ...]]
```

Description

pkgchk checks the accuracy of installed files or, by use of the **-l** flag, displays information about package files. The command checks the integrity of directory structures and the files. Discrepancies are reported on **stderr** along with a detailed explanation of the problem.

The first synopsis defined above is used to list or check the contents and/or attributes of objects that are currently installed on the system. Package names can be listed on the command line, or by default the entire contents of a machine is checked. If packages are installed in an alternative root directory path using the **pkgadd** command with the **-P** option, contents and attributes can be checked or listed using the same alternative root directory path specified with the **-P** option.

The second synopsis is used to list or check the contents of a package which has been spooled on the specified device, but not installed. Note that attributes cannot be checked for spooled packages.

The third synopsis is used to list or check the contents and/or attributes of objects which are described in the indicated *Pkgmap*.

Flags

Item	Description
-l	Lists information on the selected files that make up a package. It is not compatible with the a , c , f , g , and v flags.
-a	Audits the file attributes only, does not check file contents. Default is to check both.
-c	Audits the file contents only, does not check file attributes. Default is to check both.
-f	Corrects file attributes if possible. If used with the -x flag, it removes hidden files. When pkgchk is invoked with this flag it creates directories, named pipes, links, and special devices if they do not already exist.
-q	Enables quiet mode. Does not give messages about missing files.
-v	Enables verbose mode. Files are listed as processed.
-n	Ignores volatile or editable files. This should be used for most post-installation checking.

Item	Description
-x	Searches exclusive directories only, looking for files that exist that are not in the installation software database or the indicated <i>Pkgmap</i> file. (An exclusive directory is a directory created by and for a package; it should contain only files delivered with a package. If any non-package files are found in an exclusive directory, pkgchk reports an error.) If -x is used with the -f flag, hidden files are removed; no other checking is done. Note: To remove hidden files only, use the -f and -x flags together. To remove hidden files and check attributes and contents of files, use the -f , -x , -c , and -a flags together.
-p	Only checks the accuracy of the pathname or pathnames listed. "pathname" can be one or more pathnames separated by commas (or by white space, if the list is quoted).
-i	Reads a list of pathnames from <i>File</i> and compares this list against the installation software database or the indicated <i>Pkgmap</i> file. Pathnames that are not contained in "inputfile" are not checked.
-d	Specifies the device on which a spooled package resides. <i>Device</i> can be a directory pathname, or "-" which specifies packages in datastream format read from standard input.
-m	Requests that the package be checked against the pkgmap file <i>Pkgmap</i> .
-e	Requests that the pkginfo file named as <i>Envfile</i> be used to resolve parameters noted in the specified pkgmap file.
<i>Pkginst</i>	Defines a short string used to designate an abbreviation for the package name. (The term "package instance" is used loosely: it refers to all instantiations of <i>Pkginst</i> , even those that do not include instance identifiers.) To indicate all instances of a package, specify ' <i>Pkginst.*</i> ', enclosing the command line in single quotes, as shown, to prevent the shell from interpreting the "*" character. Use the token "all" to refer to all packages available on the source medium.
-P path	Requests that the package in the alternate root directory path be checked.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion of script.
1	Fatal error. Installation process is terminated at this point.

Files

Item	Description
<i>/usr/sbin/pkgchk</i>	Contains the pkgchk command.

pkginfo Command

Purpose

Displays software package and/or set information.

Syntax

To Display Information on Installed Packages

```
pkginfo [ -q] [ -x | -l] [ -r] [ -p | -i] [ -a Arch] [ -P Path] [ -v Version] [ -c Category1,[Category2 [, . . .]]]  
[ Pkginst [, Pkginst [, . . .]]]
```

To Display Information on Packages Contained in the Specified Device

```
pkginfo [ -d Device] [ -q] [ -x | -l] [ -a Arch] [ -P Path] [ -v Version] [ -c Category1 [,Category2 [, . . .]]]  
[ PkginstPkginst [, Pkginst [, . . .]]]
```

Description

pkginfo displays information about software packages or sets that are installed on the system (as requested in the first synopsis) or that reside on a directory (as requested in the second synopsis). A package is a collection of related files and executable that can be independently installed. A set is made up of a special-purpose package, referred to as a Set Installation Package (SIP), and a collection of one or more packages that are members of the set. The SIP controls the installation of the set.

When run without flags, **pkginfo** displays one line of information about every installed package (whether installed completely or partially) whose category is not the value "set". The information displayed includes the primary category, package instance, and name of the package. For UNIX software packages produced before UNIX System V Release 4, **pkginfo** displays only the package name and abbreviation.

The **-p** and **-i** flags are meaningless if used in conjunction with the **-d** flag. The **-p** and **-i** flags are mutually exclusive. The **-x** and **-l** flags are mutually exclusive.

Flags

Item	Description
-q	Enables quiet mode - no information is displayed. This flag overrides the -x , -l , -p , and -i flags. (Can be invoked by a program to query whether or not a package has been installed.)
-x	Extracts and displays the following information about the specified package: abbreviation, name, and, if available, architecture and version.
-l	Displays a "long format" report (that is, one that includes all available information) about the specified package(s).
-r	Lists the installation base for the specified package if the package is relocatable.
-p	Displays information only for partially installed packages.
-i	Displays information only for fully installed packages.
-a Arch	Specifies the architecture of the package as <i>Arch</i> .
-P Path	Displays information for packages installed in the alternative root directory path.
-v Version	Specifies the version of the package as <i>Version</i> . All compatible versions can be requested by preceding the version name with a tilde "~".

Item	Description
-c <i>Category</i> . . .	<p>Displays information about packages that belong to category <i>Category</i>. (Categories are defined in the category field of the pkginfo file; see the pkginfo file format for details.) More than one category may be specified in a comma-separated list. A package is required to belong to only one category, even when multiple categories are specified. The package-to-category match is not case-sensitive.</p> <p>If the category specified is "set", pkginfo displays information about Set Installation Packages (SIPs).</p>
<i>Pkginst</i>	<p>Defines a short string used to designate an abbreviation for the package/set name. (The term "package instance" is used loosely: it refers to all instantiations of <i>Pkginst</i>, even those that do not include instance identifiers.)</p> <p>To indicate all instances of a package, specify '<i>Pkginst.*</i>', enclosing the command line in single quotes, as shown, to prevent the shell from interpreting the "*" character. Use the token "all" to refer to all packages available on the source medium.</p> <p>If <i>Pkginst</i> is a SIP, information about the packages with which the SIP is associated is displayed.</p>
-d <i>Device</i>	<p>Displays information from packages/sets that reside on <i>Device</i>. <i>Device</i> can be the full pathname to a directory (such as /var/tmp), or "-" which specifies packages in datastream format read from standard input. The default device is the installation spool directory (/var/spool/pkg).</p>

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion of script.
1	Fatal error. Installation process is terminated at this point.

Files

Item	Description
/var/spool/pkg	Default spool directory

pkgmk Command

Purpose

Produces an installable package.

Syntax

```
pkgmk [ -c ] [ -o ] [ -a Arch ] [ -b BaseDir ] [ -d Directory ] [ -f Prototype ] [ -l Limit ] [ -p PStamp ] [ -r RootPath ] [ -v Version ] [ Variable=Value ... ] [ PkgInst ]
```

Description

pkgmk produces an installable package to be used as input to the **pkgadd** command. A package is a collection of related files and executables that can be independently installed. The package contents will be in directory structure format.

The **pkgmk** command uses the package prototype file as input and creates a **pkgmap** file. The contents for each entry in the prototype file is copied to the appropriate output location. Information concerning the contents (checksum, file size, modification date) is computed and stored in the **pkgmap** file, along with attribute information specified in the prototype file.

Flags

Item	Description
-a <i>Arch</i>	Overrides the architecture information provided in the pkginfo file with <i>Arch</i> .
-b <i>BaseDir</i>	Prepends the indicated <i>BaseDir</i> to locate relocatable objects on the source machine.
-c	Compresses non-information files. You must also specify the -r option when using -c . Entries in the <i>Prototype</i> file that reference relative paths above the <i>RootPath</i> specification will not be compressed. Any files that were already compressed (that is, only those in ".Z" form) before being processed by pkgmk will not be uncompressed by the pkgadd command.
-d <i>Directory</i>	Creates the package in <i>Directory</i> . The directory named must already exist.
-f <i>Prototype</i>	Uses the file <i>Prototype</i> as input to the command. The default name for this file is either Prototype or prototype . You can use pkgproto to create the <i>Prototype</i> file. In this case, you must manually add in the entries for any installation scripts and files you are using in the package. You only need entries for those files and scripts that you use. However, you must always add an entry for the pkginfo file for the package. See pkgproto for more information.
-l <i>Limit</i>	Specifies the maximum size in 512-byte blocks of the output device as <i>Limit</i> . By default, if the output file is a directory or a mountable device, pkgmk will employ the df command to dynamically calculate the amount of available space on the output device. Useful in conjunction with pkgtrans to create a package with datastream format.
-o	Overwrites the same instance. The package instance will be overwritten if it already exists.
-p <i>PStamp</i>	Overrides the production stamp definition in the pkginfo file with <i>PStamp</i> .
-r <i>RootPath</i>	Appends the source pathname in the <i>Prototype</i> file to the indicated <i>RootPath</i> to locate objects on the source machine.

Item	Description
-v <i>Version</i>	Overrides version information provided in the pkginfo file with <i>Version</i> .
<i>Variable=Value</i>	Places the indicated variable in the packaging environment.
<i>PkgInst</i>	A short string used to designate an abbreviation for the package name. pkgmk will automatically create a new instance if the version and/or architecture is different. A user should specify only a package abbreviation; a particular instance should not be specified unless the user is overwriting it.

Examples

1. If you want to create a package named mypkgA containing the **lsps** and **lsuser** commands, you must first create the contents of the package. For example:

```
mkdir -p /home/myuser/example/pkgmk/sbin
cp /usr/sbin/lsps /home/myuser/example/pkgmk/sbin
cp /usr/sbin/lsuser /home/myuser/example/pkgmk/sbin
```

Then, create the **pkginfo** file. In this example the **pkginfo** file is /home/myuser/example/pkgmk/pkginfo, which contains the following:

```
PKG="mypkgA"
NAME="My Package A"
ARCH="PPC"
RELEASE="1.0"
VERSION="2"
CATEGORY="Application"
PSTAMP="AIX 2001/02/05"
```

Then, create the *Prototype* file, /home/myuser/example/pkgmk/prototype file which contains the following:

```
!search /home/myuser/example/pkgmk/sbin
i pkginfo=/home/myuser/example/pkgmk/pkginfo
d example /example 1777 bin bin
d example /example/pkgmk 1777 bin bin
d example /example/pkgmk/sbin 1777 bin bin
f example /example/pkgmk/sbin/lsps 555 bin bin
f example /example/pkgmk/sbin/lsuser 555 bin bin
```

Then, create the package with the above *Prototype* and **pkginfo** files using the **pkgmk** command:

```
pkgmk -d /tmp -f /home/myuser/example/pkgmk/prototype
```

This produces the following output:

```
Building pkgmap from package prototype file
### Processing pkginfo file
    WARNING:parameter <CLASSES> set to "example"

### Attempting to volumize 5 entries in pkgmap
Part 1 -- 218 blocks, 10 entries
/tmp/mypkgA/pkgmap
/tmp/mypkgA/pkginfo
/tmp/mypkgA/root/example/pkgmk/sbin/lsps
/tmp/mypkgA/root/example/pkgmk/sbin/lsuser
### Packaging complete
```

The newly created package named mypkgA now exists in /tmp/mypkgA.

Exit Status

Item	Description
0	Successful completion of script.
1	Fatal error. Installation process is terminated at this point.
99	Internal error.

Files

Item	Description
/usr/sbin/pkgmk	Contains the pkgmk command.

pkgparam Command

Purpose

Displays package parameter values.

Syntax

To Display the Value of a Parameter Contained in pkginfo

```
pkgparam [ -v ] [ -d Device ] [ -P path ] Pkginst [ Param ...]
```

To Display the Value of a Parameter Contained in a Device

```
pkgparam -d Device [ -v ] [ Param ...]
```

To Display the Value of a Parameter Contained in a File

```
pkgparam -f File [ -v ] [ Param ...]
```

Description

pkgparam displays the value associated with the parameter or parameters requested on the command line. The values are located in one of the following places: in the **pkginfo** file for *Pkginst*, on the *Device* named with the **-d** flag, or on the specific file named with the **-f** flag. When a *Device* is given, but a *Pkginst* is not (as shown in the second synopsis), parameter information for all packages residing on *Device* is shown.

If packages are installed in an alternative root directory path using the **pkgadd** command with the **-P** option, package parameters can be requested using the same alternative root directory path specified with the **-P** option.

One parameter value is shown per line. Only the value of a parameter is given unless the **-v** flag is used. With this flag, the output of the command is in this format:

```
Parameter1='Value1'  
Parameter2='Value2'  
Parameter3='Value3'
```

If no parameters are specified on the command line, values for all parameters associated with the package are shown.

Flags

Item	Description
-v	Specifies verbose mode. Displays name of parameter and its value.

Item	Description
-d <i>Device</i>	Specifies the <i>Device</i> on which a <i>Pkginst</i> is stored. <i>Device</i> can be the full pathname to a directory (such as /var/tmp), or "-" which specifies packages in datastream format read from standard input.
-f	Requests that the command read <i>File</i> for parameter values. This file should be in the same format as a pkginfo file. As an example, such a file might be created during package development and used while testing software during this stage.
<i>Pkginst</i>	Defines a specific package for which parameter values should be displayed. The format <i>Pkginst.*</i> can be used to indicate all instances of a package. When using this format, enclose the command line in single quotes to prevent the shell from interpreting the "*" character.
<i>Param</i>	Defines a specific parameter whose value should be displayed.
-P <i>path</i>	Searches for the pkginfo file in the alternate root directory path.

Exit Status

If parameter information is not available for the indicated package, the command exits with a non-zero status.

Item	Description
0	Successful completion of script.
1	Fatal error. Installation process is terminated at this point.

Files

Item	Description
/var/spool/pkg	default spool directory
/usr/sbin/pkgparam	Contains the pkgparam command.

pkgproto Command

Purpose

Generates a prototype file.

Syntax

```
pkgproto [ -i ] [ -c Class ] [Path1 [=Path2] ...]
```

Description

The **pkgproto** commands scans the indicated paths and generates a prototype file that may be used as input to the **pkgmk** command. To do this, the standard output of this command must be redirected to a file. The file can then be used when invoking **pkgmk**.

If no *Paths* are specified on the command line, standard input is assumed to be a list of *Paths*. If the *Path* listed on the command line is a directory, the contents of the directory are searched. However, if input is read from stdin, a directory specified as a path is not searched.

The prototype file attributes *mac*, *fixed*, and *inherited*, cannot be determined by **pkgproto** and must be manually added to the file.

By default, **pkgproto** creates symbolic link entries for any symbolic link encountered (ftype=s). When you use the **-i** flag, **pkgproto** creates a file entry for symbolic links (ftype=f). The prototype file must be edited to assign file types such as v (volatile), e (editable), or x (exclusive directory). **pkgproto** detects linked files. If multiple files are linked together, the first path encountered is considered the source of the link.

The output from this command is sent to standard output. You must redirect standard output to a file if you wish to use the result as a prototype file when invoking **pkgmk**. Since **pkgmk** uses prototype as the default filename for the prototype file, we suggest you direct the output of **pkgproto** to the file name **prototype**.

You must add entries to the prototype file produced by this command for any installation scripts and files your package may need. At minimum, you will need an entry for the **pkginfo** file. You may also need entries for any of the following files you use in your package: **copyright**, **compver**, **depend**, **setinfo**, **space**, any installation or removal scripts you define for the package, and/or any classes you define.

Note:

1. By default, **pkgproto** creates symbolic link entries for any symbolic link encountered (ftype=s). When you use the **-i** option, **pkgproto** creates a file entry for symbolic links (ftype=f). The prototype file must be edited to assign file types such as v (volatile), e (editable), or x (exclusive directory). **pkgproto** detects linked files. If multiple files are linked together, the first path encountered is considered the source of the link.
2. The output from this command is sent to standard output. You must redirect standard output to a file if you wish to use the result as a prototype file when invoking **pkgmk**. Since **pkgmk** uses prototype as the default filename for the prototype file, we suggest you direct the output of **pkgproto** to the file name **prototype**.
3. Note that you must add entries to the **prototype** file produced by this command for any installation scripts and files your package may need. At minimum, you will need an entry for the **pkginfo** file; see **pkginfo** for more information. You may also need entries for any of the following files you use in your package: **copyright**, **compver**, **depend**, **setinfo**, **space**, any installation or removal scripts you define for the package, and/or any classes you define, (e.g., postinstall).

Flags

Item	Description
-i	Ignores symbolic links and records the paths as ftype=f (a file) versus ftype=s (symbolic link).
-c Class	Maps the class of all paths to <i>Class</i> .
<i>Path1</i>	Path of directory where objects are located.
<i>Path2</i>	Path that should be substituted on output for <i>Path1</i> .

Examples

The following examples show uses of **pkgproto** and a partial listing of the output produced.

```

1. $ pkgproto /usr/bin=bin /usr/usr/bin=usrbin /etc=etc
f none bin/sed=/bin/sed 0775 bin bin
f none bin/sh=/bin/sh 0755 bin daemon
f none bin/sort=/bin/sort 0755 bin bin
d none etc/master.d 0755 root daemon
f none etc/master.d/kernel=/etc/master.d/kernel 0644 root daemon
f none etc/rc=/etc/rc 0744 root daemon

2. $ find / -type d -print | pkgproto
d none / 755 root root
d none /usr/bin 755 bin bin
d none /usr 755 root root
d none /usr/bin 775 bin bin
d none /etc 755 root root
d none /tmp 777 root root

```

- Identical to the previous example, but with the output captured in a file for later processing with **pkgmk**. Entries added for the required **pkginfo** file, and, for instance, a postinstall script that might be executed after the files are copied into the correct locations.

```
$ find / -type d -print | pkgproto >prototype
$ (edit the file to add entries for pkginfo and postinstall)
$ cat prototype
i pkginfo
i postinstall
d none / 755 root root
d none /usr/bin 755 bin bin
d none /usr 755 root root
d none /usr/bin 775 bin bin
d none /etc 755 root root
d none /tmp 777 root root
```

Return Codes

Item	Description
0	Successful completion of script.
1	Fatal error. Installation process is terminated at this point.

Files

Item	Description
<code>/usr/sbin/pkgproto</code>	Contains the pkgproto command

pkgrm Command

Purpose

Removes a package or set from the system.

Syntax

To Remove an Installed Software Package

```
pkgrm [ -n] [ -a Admin] [ -P Path] [ Pkginst1 [ Pkginst2 [ . . . ] ] ]
```

To Remove a Software Package from a Spool Device

```
pkgrm -s Spool [ Pkginst ]
```

Description

pkgrm removes a previously installed or partially installed package/set from the system. A package is a collection of related files and executables that can be independently installed. A set is made up of a special-purpose package, referred to as a Set Installation Package (SIP), and a collection of one or more packages that are members of the set.

pkgrm checks that all packages listed on the command line are on the system. If any of the packages listed does not exist, no changes are made to the system, that is, none of the listed packages are removed.

A check is also made to determine if any other packages depend on the one being removed. The action taken if a dependency exists is defined in the *Admin* file (see the **-a** flag, below).

The default state for the command is interactive mode, meaning that prompt messages are given during processing to allow the administrator to confirm the actions being taken. Non-interactive mode can be requested with the **-n** flag.

The **-s** flag can be used to specify the directory from which spooled packages should be removed.

Flags

Item	Description
-n	Enables non-interactive mode. If there is a need for interaction, the command exits. Use of this flag requires that at least one package instance be named upon invocation of the command.
-a Admin	Defines an installation administration file, <i>Admin</i> , to be used in place of the default administration file. [For a description of the format of an <i>Admin</i> file, see the admin file format.] The token "none" overrides the use of any <i>Admin</i> file, and thus forces interaction with the user. Unless a full pathname is given, pkgrm looks in the /var/sadm/install/admin directory for the file. By default, the file default in that directory is used.
-P Path	Removes the specified packages from the alternative root directory path.
-s Spool	Removes the specified package(s) from the directory <i>Spool</i> .
<i>Pkginst</i>	Defines a short string used to designate an abbreviation for the package/set name. (The term "package instance" is used loosely: it refers to all instantiations of <i>Pkginst</i> , even those that do not include instance identifiers.) If <i>Pkginst</i> specifies a SIP, all installed packages which are members of the set, and the SIP itself, are removed in reverse dependency order. To indicate all instances of a package, specify ' <i>Pkginst.*</i> ', enclosing the command line in single quotes, as shown, to prevent the shell from interpreting the "*" character. Use the token "all" to refer to all packages available on the source medium.

Exit Status

This command returns the following exit values:

Item	Description
0	Successful completion of script.
1	Fatal error. Installation process is terminated at this point.
99	Internal error.

Files

Item	Description
/usr/sbin/pkgrm	Contains the pkgrm command

pkgtrans Command

Purpose

Translates package format.

Syntax

pkgtrans [**-i -o -n -s**] [**-z Blocksize**] *Device1 Device2* [*Pkginst1* [*Pkginst2* [...]]]

Description

The **pkgtrans** command translates an installable package from one format to another. It translates the following:

- A file system format to a datastream
- A datastream to a file system format

You cannot run **pkgtrans** from **csH**.

Flags

Item	Description
-i	Copies the pkginfo and <i>Pkgmap</i> files. If the packages category is defined as "set" for Set Installation Packages (SIPs) (see setinfo file format), then that packages' setinfo file is also copied.
-o	Overwrites the same instance on the destination device. The package instance is overwritten if it already exists.
-n	Creates a new instance of the package on the destination device. If the package instance already exists on the destination device, it is left unchanged and a new instance is created. The new instance has a sequence number attached to distinguish it from the existing instance. For example, assume the destination device already contained an instance of package <i>X</i> . If you use pkgtrans with the -n flag to write a new instance of package <i>X</i> to the device, the existing instance of package <i>X</i> remains on the destination device, and a new instance, called <i>X.2</i> , would be created on the device. If you executed pkgtrans again with the -n flag, a third instance, called <i>X.3</i> , would be created.
-s	Indicates that the package should be written to <i>Device2</i> as a datastream rather than as a file system. The default behavior is to write to <i>Device2</i> in the file system format.
-z <i>Blocksize</i>	Indicates the blocksize to be used when transferring to cartridge tape. Packages that have been written to tape using the -z flag and a value not equal to 512 are always read using a blocksize of 32768. Thus, the -z flag is not applicable when reading from cartridge tape.
<i>Device1</i>	Indicates the source device. Can be - (hyphen) which specifies packages in datastream format read from standard input. The package or packages on this device are translated and placed on <i>Device2</i> . If <i>Device1</i> is a regular file or directory, you must use the absolute pathname, rather than a relative pathname.
<i>Device2</i>	Indicates the destination device. Can be - (hyphen) which specifies packages written to standard output in datastream format. Translated packages are placed on this device. If <i>Device2</i> is a regular file or directory, you must specify it as an absolute pathname, rather than a relative pathname.
<i>Pkginst</i>	Specifies which package on <i>Device1</i> should be translated. The token "all" may be used to indicate all packages. <i>Pkginst.*</i> can be used to indicate all instances of a package. If no packages are defined, a prompt shows all packages on the device and asks which to translate. If a set is being transferred to datastream format, the <i>Pkginst</i> arguments should begin with the SIP and be followed by the packages listed in the SIP's setinfo file, in the order in which they appear in that file.

Note: By default, **pkgtrans** does not transfer any instance of a package if any instance of that package already exists on the destination device. Use of the **-n** flag creates a new instance if an instance of this package already exists. Use of the **-o** flag overwrites the same instance if it already exists. Neither of these flags are useful if the destination device is a datastream, because the entire datastream is overwritten anyway.

Exit Status

This command returns the following exit values:

Item	Description
------	-------------

- | | |
|---|--|
| 0 | Successful completion of script. |
| 1 | Fatal error. Installation process is terminated at this point. |

Files

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

Examples

1. To translate all packages located on drive *Device* and place the translations in **/tmp**, type:

```
pkgtrans Device /tmp all
```

2. To translate "pkg1" and "pkg2" in **tmp** and place them on *Device* in a datastream format, type:

```
pkgtrans -s /tmp Device pkg1 pkg2
```

platform_dump Command

Purpose

Perform platform (hardware and firmware) dump related actions.

Syntax

```
platform_dump { -c | -d | -e | -f fstype | -F flag | -l | -q | -S | -s seq_no } [ -L ] [ -N flag ]
```

Description

The `platform_dump` command supports hardware and firmware problem determination for POWER5 platforms. You can use this command to help the operating system save firmware-related and hardware-related dumps. This command is supported only on partitions which have service authority enabled, except for Hardware Management Console (HMC) managed systems. On an HMC managed system, the dumps go to the HMC. The `platform_dump` command is normally run by operating system functions such as base installation and `dumpcheck`. Platform dumps contain:

- The hardware state
- The hypervisor state
- The FSP (Flexible Service Processor) state information

The disk space for platform dump files is reserved using the `platform_dump` command. A dedicated logical volume, **/dev/fwdump**, is created in the rootvg volume group and mounted on the **/var/adm/ras/platform** directory. The **fwdump_dev** device and **fwdump_dir** mount point are both saved in ODM in the SWservAt object class. During installation, AIX utilizes the `platform_dump` command to reserve the necessary disk space. The disk space is reserved only if the partition is designated to be a service partition. The maximum possible size for the platform dumps is indicated to AIX so that enough space can be allocated beforehand for the platform dumps. Note that this size can change dynamically. The operating system detects this and informs you about the extra requirements and automatically expands the logical volume if possible.

Note: If you assign service partition authority to an AIX partition after the partition was installed, run the `platform_dump -f <fstype>` command to create the `/dev/fwdump` rootvg logical volume. The `fstype` argument can have the `jfs2` or `jfs` value.

The `-L` flag is provided to record command output to the error log.

Flags

Item	Description
<code>-c</code>	Performs a check on the estimated platform dump size (as indicated by the firmware) and the disk space allocated for the platform dumps. It will report the following: If estimated size is less than or equal to allocated space, will return 0. If estimated size is greater than allocated space, will return 1.
<code>-d</code>	Deletes the file system space reserved for platform dumps and free up the same for other uses. Any existing dump files on the reserved disk space will be lost.
<code>-e</code>	Provides an estimate of disk space required to save the platform dumps when they occur. This option will interact with the firmware to provide this estimate. It is expected that, based on this space information, the user will have enough disk space allocated for platform dumps to be saved. The value output will be the required size in bytes.
<code>-f fstype</code>	Reserves enough disk space on the system for platforming dumps. The <code>-f</code> option will create a file system (if one does not exist) exclusively for platform dumps. If a file system already exists and the size is not enough, the file system size will be increased. The <code>fstype</code> must be a valid file system type. If the file system already exists, any may be specified.
<code>-F flag</code>	Enables or disables platform dumps. If flag is 0, platform dumps are disabled, if 1, platform dumps are enabled.
<code>-l</code>	Lists the current configuration of platform dump.
<code>-L</code>	Tells <code>platform_dump</code> to log its output as well as displaying it. This does not apply to the size output by the <code>-e</code> option.
<code>> -N flag <</code>	<code>> </code> Enables or disables platform dump compression. If the <code>flag</code> parameter is 0, platform dump compression is disabled, if the <code>flag</code> parameter is 1, platform dump compression is enabled. <code> <</code>
<code>-q</code>	Checks whether the platform supports platform dumps or not. Will return 0 if platform dump is supported.
<code>-s seq_no</code>	Saves the platform dump from the firmware as identified in the dump notification event. <code>seq_no</code> indicates the sequence number of the dump notification event as stored in the AIX error log file. This sequence number will be used by this command to parse the detailed data area and obtain dump tag and dump type information needed to obtain the dump data from firmware.
<code>-S</code>	Saves the scan dumps on systems which support scan data. When this option is specified, the command will check for the existence of a scan dump, and if so will read and save the scandump data from firmware using the existing scan dump interface.

Exit Status

0

On successful completion.

1

Returned if `-c` was specified, and there is insufficient space to save platform dumps.

255

Returned if platform dumps are not supported on the system.

3

Returned if platform dumps has been disabled.

2

Returned in an error is encountered.

Security

The `platform_dump` may only be executed by the root user.

Example

1. To get an estimate of the platform dumps size, type the following:

```
platform_dump -e
```

This will report the estimated platform dump size in bytes.

plotgbe Command

Purpose

Plots HP-GL files to a plotter device.

Syntax

```
/usr/lpd/plotgbe [ -fr=X ] [ -noin ] File
```

Description

The **plotgbe** command is a backend program which plots HP-GL files to a plotter device. The plotter device must be attached to a 5085/5086 workstation via the 5080 Attachment Adapter. To use the **plotgbe** command, you must define a print queue for the **plotgbe** backend program. See **enq** command, use the **-o** flag to pass options to the **plotgbe** backend for processing.

The **plotgbe** backend command also generates the appropriate HP-GL commands for plotter initialization and plot scaling. This data is sent to the plotter before the user-specified HP-GL file is sent. Thus, any scaling or initialization commands included in the HP-GL file override those generated by the **plotgbe** backend command.

Note: The user must have read access to the file sent to the **plotgbe** command with the print request command.

Flags

Item	Description
-fr=X	Provides for plotting multi-frame drawings. This option causes <i>X</i> number of frames to be plotted, where <i>X</i> is a number in the range 1 through 9. For example, plotting a 20' drawing on E-size role media may require 5 frames. Thus, the option <code>fr=5</code> would be passed to the plotgbe backend.
-noin	Allows plotter front panel settings to remain in effect for the current plot without being reset to default values. Normally, the P1 and P2 positions which define the plot page on the plotter are set by the plotgbe command to their default location. Use the -noin no-initialization option to override the default locations.

Examples

1. To send the file `longaxis.gl` to the `plt` plotter queue and specify to the backend that the file requires five frames to print, enter:

```
enq -Pplt -o -fr=5 longaxis.gl
```

2. To send the file `plotdata.gl` to the `plt` plotter queue, specifying that the plot page positions are not to be reset to default for this file, enter:

```
enq -Pplt -o -noin plotdata.gl
```

3. To send the file `twoplot.gl` to the `plt` plotter queue, specifying no plot page initialization and that the plotter print the drawing in two frames, enter:

```
enq -Pplt -o -noin -o fr=2 twoplot.gl
```

Files

Item	Description
<code>/usr/lpd/plotgbe</code>	Contains the plotgbe command.

plotlbe Command

Purpose

Plots HP-GL files to a plotter device.

Syntax

```
/usr/lpd/plotlbe [ -fr=X ] [ -noin ] File
```

Description

The **plotlbe** command is a backend program which plots HP-GL files to a plotter attached to a serial port defined as a TTY device. To use the **plotlbe** command, you must define a TTY device for the serial port and define a print queue for the **plotlbe** backend program.

When configuring the TTY serial port, set the baud-rate, parity, and stop bits to the appropriate settings for your plotter. You must also set XON/XOFF to FALSE for your TTY port.

The **plotlbe** command is called by the **qdaemon** process. It should not be entered on the command line. Any options needed for a specific print request to a plotter should be passed to the **plotlbe** command with the command used to request a print job (usually the **enq** command). With the **enq** command, use the **-o** flag to pass options to the **plotlbe** backend for processing.

The **plotlbe** backend command supports the following plotters: 7731, 7372, 7374, 7375-1, 7375-2, 6180, 6182, 6184, 6186-1, and 6186-2.

The **plotlbe** command supports ENQ/ACK handshaking. Refer to your plotter programming manual for more information on handshaking.

The **plotlbe** backend command also generates the appropriate HP-GL commands for plotter initialization and plot scaling. This data is sent to the plotter before the user-specified HP-GL file is sent. Thus, any scaling or initialization commands included in the HP-GL file override those generated by the **plotlbe** backend command.

Note: The user must have read access to the file sent to the **plotlbe** command with the print request command.

Flags

Item	Description
-fr=X	Provides for plotting multi-frame drawings. This option causes <i>X</i> number of frames to be plotted, where <i>X</i> is a number in the range 1 through 9. For example, plotting a 20' drawing on E-size roll media may require 5 frames. Thus, the option -fr=5 would be passed to the plotlbe backend.
-noin	Allows plotter front panel settings to remain in effect for the current plot without being reset to default values. Normally, the P1 and P2 positions which define the plot page on the plotter are set by the plotlbe command to their default locations. Use the -noin no-initialization option to override the default locations.

Examples

1. To send the file `longaxis.gl` to the `plt` plotter queue and specify to the backend that the file requires five frames to plot, enter:

```
enq -Pplt -o -fr=5 longaxis.gl
```

2. To send the file `plotdata.gl` to the `plt` plotter queue, specifying that the plot page positions are not to be reset to default for this file, enter:

```
enq -Pplt -o -noin plotdata.gl
```

3. To send the file `twoplot.gl` to the `plt` plotter queue, specifying no plot page initialization and that the plotter print the drawing in two frames, enter:

```
enq -Pplt -o -noin -o fr=2 twoplot.gl
```

Files

Item	Description
<code>/usr/lpd/plotlbe</code>	Contains the plotlbe command.

pmctl Command

Purpose

Starts, resets, or stops generating Performance Monitor events.

Syntax

```
pmctl [ { [ -E mode ] ] [ -f interval ] { [ -y command ] } | [ -O Options ] | [ -h ] | [ -r ] | [ -S ] | [ -s ] { [ -a y command ] [ -f interval ] }
```

Description

The **pmctl** command starts, stops, or resets the generation of Performance Monitor events in the PMAPI subsystem to support manual offline mode with the **tprof -E** command. It also reports the current status of the PMAPI subsystem.

Flags

Item	Description
-a	Turns on large page analysis.

Item	Description
-E [<i>mode</i>]	<p>Enables event-based profiling. You can specify one of the following modes:</p> <p>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).</p> <p>EMULATION Enables the emulation profiling mode.</p> <p>ALIGNMENT Enables the alignment profiling mode.</p> <p>ISLBMISS Enables the Instruction Segment Lookaside Buffer miss profiling mode.</p> <p>DSLBMISS Enables the Data Segment Lookaside Buffer miss profiling mode.</p>
-f <i>interval</i>	<p>Specifies the sampling interval to use.</p> <ul style="list-style-type: none"> • For processor cycle, EMULATION, ALIGNMENT, ISLBMISS, and DSLBMISS events, specify 1 to 500 milliseconds (default = 10). • For other Performance Monitor events, specify 10000 up to MAXINT occurrences (default = 10000). <p>If you use the -f flag with the -y flag, specify 1 up to MAXINT occurrences for other Performance Monitor events (default = 10000).</p>

Item

>|**o** options|<

Description

This **-O** option can have the following values:

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.

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.

Item**Description**

>|**-o options** (continued)|<

threshold_eve_start_sel=name

Specifies the event to start the threshold counting. For valid values of this option, see [threshold_eve_stop_sel](#).

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	Description
> -o options (continued) <	<p>random_samp_ele_crit=<i>name</i> Specifies the random criteria for selecting the instructions for sampling. Valid values for this option are as follows:</p> <p>ALL_INSTR All instructions are eligible. This value is the default setting.</p> <p>LOAD_STORE The operation is routed to the Load Store Unit (LSU); for example, load, store.</p> <p>PROB_NOP Sample only special no-operation instructions, which are called Probe NOP events.</p> <p>LARX_STCX Sample only load and reserve indexed (LARX) instructions and store conditional indexed (STCX) instructions.</p> <p>LOAD_SAMPLING Sample only load instructions.</p> <p>LONG_LATENCY_OP Sample only long latency instructions (div/sqrt/mul).</p> <p>STORE_SAMPLING Sample only store instructions.</p> <p>LOAD_MISSES Sample only load-miss instructions.</p> <p>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.</p> <p>BRANCH_MISPREDICTS Sample only branch-mispredict instructions.</p> <p>BRANCH_MISPREDICTS_DIR_CR_CTR Sample only branch-mispredict events (Direction, CR, or CTR).</p> <p>BRANCH_MISPREDICTS_TA Sample only branch-mispredict events (TA).</p> <p>BRANCHES_RESOLVED_TAKEN Sample branches with resolution.</p> <p>NON_REPEATING_BRANCHES Sample only non-repeating branches.</p> <p>ALL_BRANCHES_REQ_PRED Sample all branches that require prediction.</p> <p>group_num=<i>event_group</i> Specifies a group number that must be used for the pmctl 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.</p>
-h	Prints man page information.
-r	Releases and resets the PMAPI subsystem.
-S	Stops generating Performance Monitor events.

Item	Description
-s	Prints the current status of the PMAPI subsystem.
-y command	Turns on the event-based profiling only for the specified command and its descendents.

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 stop generating Performance Monitoring events, enter the following command:

```
pmctl -S
```

2. To reset generating Performance Monitoring events, enter the following command:

```
pmctl -r
```

3. To report the current status of the PMAPI subsystem, enter the following command:

```
pmctl -s
```

4. To start generating Performance Monitoring events, enter the following command:

```
pmctl -E
```

5. To start generating Performance Monitoring events only for the specified **workload** command and its descendents, enter the following command:

```
pmctl -E -y workload
```

6. To support the **tprof -E** command in manual offline mode, enter the following command:

```
trace -adf -o mydata.trc
trcon
pmctl -E
sleep 10; trcstop
gensyms > mydata.syms
tprof -suker mydata
```

7. To support the **tprof -E** command in manual offline mode profiling for the specified **workload** command and its descendents, enter the following command:

```
trace -adf -o mydata.trc
trcon
pmctl -E -y workload
trcstop
gensyms > mydata.syms
tprof -suker mydata
```

8. **>|**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:

```
trace -adf -o mydata.trc
trcon
pmctl -E PM_THRESH_MET-0 threshold_cmp_val=10000 -0
threshold_event_sel=THRESHOLD_COUNT_CYCLES -0
threshold_eve_start_sel=PM_MRK_L1_RELOAD_VALID -0
threshold_eve_stop_sel=PM_MRK_ST_DONE_L2 -0
random_samp_ele_crit=ALL_INSTR -0 group_num=607
```

```
sleep 10; trcstop
gensyms > mydata.syms
tprof -suker mydata
```

|<

pmcycles Command

Purpose

Measures processor clock speed.

Syntax

```
pmcycles [ -d ] [ -m ] [ >|M|< ]
```

Description

The **pmcycles** command displays the *nominal* and measured processor speed for the system in MHz. The nominal processor speed is the maximum frequency at which the system can run across all environments and workload conditions. Depending on system conditions, the nominal processor frequency might not represent the minimum or maximum achievable processor speed. The measured frequency is the current running frequency of the processor.

The **lparstat -E 1 1** and **mpstat -E 1 1** commands can also be used to determine the current processor speed.

Flags

Item	Description
-d	Displays the decremter in MHz and nanoseconds per tick.
-m	Displays the speed of each of the processors.
> M <	> Displays the measured frequency of the processor. <

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 display the processor speed, enter the following command:

```
pmcycles
```

The output might be similar to the following example:

```
This machine runs at 133 MHz
```

2. To display each processor speed, enter the following command:

```
pmcycles -m
```

The output might be similar to the following example:

```
Cpu 0 runs at 200 MHz
CPU 1 runs at 200 MHz
```

3. >| To display the measured processor speed, enter the following command:

```
pmcycles -M
```

The output might be similar to the following example:

```
The measured frequency is 2294 MHz
```

|<

pmlist Command

Purpose

Lists information about supported processors.

Syntax

```
pmlist [ -h ]
```

```
pmlist -l [ -o t | c | x ]
```

```
pmlist { -s | -e ShortName | -c Counter [ ,event ] | -g Group | -S Set | -D DerivedMetricID | -m MetricGroup |  
-V Variable } [ -p ProcessorType ] [ -s ] [ -d ] [ -o t | c | x ] [ -f Filter ]
```

Description

The **pmlist** command performs the following functions:

- List the supported processors.
- List the information summary for a specified processor.
- List the event table for a specified processor.
- List any existing event groups for a specified processor.
- List any existing event sets for a specified processor.
- List the event set and formula for a specified derived metric.
- List the variables in the derived metric files.

Flags

Item	Description
-c -1	Lists all events for all counters.
-c Counter	Lists all events for the specified <i>Counter</i> .
-c Counter,Event	Lists the specified <i>Event</i> for the specified <i>Counter</i> .
-d	Displays event detailed description.
-D -1	Displays all the derived metrics supported.
-D DerivedMetricID	Displays the specified <i>DerivedMetricID</i> .
-e ShortName	Lists the description of the specified <i>ShortName</i> for all Counters.

Item	Description
-f v,u,c	Specifies the event filter as a comma-separated list of filters. The valid filters are: v (verified) , u (unverified) , and c (caveat). These filters represent the testing status of an event. The default filter is v,u,c .
-g -1	Lists all event groups.
-g Group	Lists the specified event <i>Group</i> .
-h	displays help information for the pmlist command.
-l	Lists all supported processor types.
-m -1	Lists all derived metrics by metric group.
-m MetricGroup	Displays all the derived metrics that pertain to the specified <i>MetricGroup</i> .
-o t c x	Specifies the output format for the pmlist command. The valid output formats are specified as one of: t (text format), c (CSV format) and x (XML format). The default output format is text.
-p ProcessorType	Specifies the processor type.
-s	Displays the processor information summary.
-S -1	Displays all the event sets supported.
-S Set	Displays the specified event <i>Set</i> .
-V -1	Displays all the variables that are used to calculate derived metrics.
-V Variable	Displays the specified variable.

Examples

1. To display the list of all supported processors, type:

```
pmlist -l
```

2. To display a summary information for the current processor, type:

```
pmlist -s
```

3. To display a summary information for the current processor in CSV format, type:

```
pmlist -s -o c
```

4. To display group number 62 for the current processor (if the current processor supports event groups), type:

```
pmlist -g 62
```

5. To display detailed information for event 3 of counter 1 of POWER4 processor, type:

```
pmlist -p POWER4 -c 1,3 -d
```

6. To display set number 2 for the current processor (if the current processor supports event sets), type:

```
pmlist -S 2
```

pmtu Command

Purpose

Displays and deletes Path MTU discovery related information.

Syntax

```
pmtu [-inet6] display/[delete [-dst destination] [-gw gateway] ]
```

Description

The **pmtu** command is provided to manage the Path MTU information. The command can be used to display the Path MTU table. By default the Ipv4 pmtu entries are displayed. Ipv6 pmtu entries can be displayed using the **-inet6** flag. This command also enables a root user to delete a pmtu entry with the **pmtu delete** command. The delete can be based on destination, gateway, or both.

A pmtu entry gets added into the PMTU table when a route add occurs with an MTU value.

A network option, **pmtu_expire**, is provided to expire unused pmtu entries. The default value of **pmtu_expire** is 10 minutes.

Flags

Item	Description
-dst	Specifies the destination of the pmtu entry to be deleted.
-gw	Specifies the gateway of the pmtu entry to be deleted.
-inet6	Specifies to display or delete Ipv6 pmtu entry.

Exit Status

Item	Description
0	The command completed successfully.
1	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 display Ipv4 pmtu entries, type:

```
pmtu display
```

The output will look similar to the following:

```
dst          gw          If    pmtu    refcnt  redisc_t  exp
-----
192.168.5.5  192.168.10.33  en2   1500    1       0         0
```

The reference count signifies the number of current TCP and UDP applications using this pmtu entry.

The `redisc_t` entry signifies the amount of time that is elapsed since the last Path MTU discovery attempt. The PMTU is rediscovered after every `pmtu_rediscover_interval` minutes. Its default value is 30 minutes and can be changed using the **no** command.

The PMTU entry expiry is controlled by the network option `pmtu_expire`. Its default value is 10 minutes. This value can be changed using the **no** command. A value of 0 does not expire any entries. The `exp` entry signifies the expiry time. PMTU entries having more than zero `refcnt` have `exp` of 0. When the `refcnt` becomes zero, the `exp` time increases every minute and the entry gets deleted when the **exp** variable becomes equal to `pmtu_expire`.

2. To delete an entry based on destination, type:

```
pmtu delete -dst 192.168.5.5
```

3. To display Ipv6 , type:

```
pmtu -inet6 display
```

Output will look similar to the following:

```
dst                gw      If    pmtu    refcnt  redisc_t  exp
-----
fe80::204:acff:fee4:ab3b  ::    lo0   16896    2        2         0
```

Location

`/usr/sbin/pmtu`

Files

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

pop3d Daemon

Purpose

Starts the Post Office Protocol Version 3 (POP3) server process.

Syntax

```
pop3d [-c]
```

Description

The **pop3d** command is a POP3 server. It supports the POP3 remote mail access protocol. Also, it accepts commands on its standard input and responds on its standard output. You normally invoke the **pop3d** command with the **inetd** daemon with those descriptors attached to a remote client connection.

The **pop3d** command works with the existing mail infrastructure consisting of **sendmail** and **bellmail**.

Flags

Item	Description
<code>-c</code>	Suppresses the reverse host name lookup.

Parameters

Item Description

None

Exit Status

All error and status information is written to a logfile if **syslogd** is configured for logging.

Security

The **pop3d** daemon is a PAM-enabled application with a service name of *imap*. 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 **imap** service in **/etc/pam.conf**. The **pop3d** daemon requires **/etc/pam.conf** entries for the **auth** and **session** module types. Listed below is a recommended configuration in **/etc/pam.conf** for the **imap** service:

```
#
# AIX imap configuration
#
imap auth      required    /usr/lib/security/pam_aix
imap session   required    /usr/lib/security/pam_aix
```

Note: Because the **pop3d** daemon uses the **imap** library for authentication, the **imap** service is used for both the **imapd** and **pop3d** daemons.

Files

Item	Description
/usr/sbin/pop3d	Contains the pop3d command.
html	

pop3ds Daemon

Purpose

Starts the Post Office Protocol Version 3 (POP3) server process over TLS/SSL.

Syntax

pop3ds [-c]

Description

The **pop3ds** command is a POP3 server. It supports the POP3 remote mail access protocol. Also, it accepts commands on its standard input and responds on its standard output. You normally invoke the **pop3d3** command with the **inetd** daemon with those descriptors attached to a remote client connection.

The **pop3ds** command works with the existing mail infrastructure consisting of **sendmail** and **bellmail**.

Flags

Item	Description
-c	Suppresses the reverse host name lookup.

Parameters

Item Description

None

Exit Status

All error and status information is written to a logfile if **syslogd** is configured for logging.

Security

The **pop3ds** daemon is a PAM-enabled application with a service name of *imap*. 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 **imap** service in **/etc/pam.conf**. The **pop3ds** daemon requires **/etc/pam.conf** entries for the **auth** and **session** module types. Listed below is a recommended configuration in **/etc/pam.conf** for the **imap** service:

```
#
# AIX imap configuration
#
imap auth      required    /usr/lib/security/pam_aix
imap session   required    /usr/lib/security/pam_aix
```

Note: Because the **pop3ds** daemon uses the **imap** library for authentication, the **imap** service is used for both the **imapds** and **pop3ds** daemons.

Files

Item	Description
/usr/sbin/pop3ds	Contains the pop3ds command.
html	

portmap Daemon

Purpose

Converts RPC program numbers into Internet port numbers.

Syntax

/usr/sbin/portmap

Description

The **portmap** daemon converts RPC program numbers into Internet port numbers.

When an RPC server starts up, it registers with the **portmap** daemon. The server tells the daemon which port number it is listening to and which RPC program numbers it serves. Thus, the **portmap** daemon knows the location of every registered port on the host and which programs are available on each of these ports.

A client consults the **portmap** daemon only once for each program the client tries to call. The **portmap** daemon tells the client which port to send the call to. The client stores this information for future reference.

Since standard RPC servers are normally started by the **inetd** daemon, the **portmap** daemon must be started before the **inetd** daemon is invoked.

Note: If the **portmap** daemon is stopped or comes to an abnormal end, all RPC servers on the host must be restarted.

Flags

None

Examples

1. To start the **portmap** daemon, enter the following command:

```
startsrc -s portmap
```

2. To stop the **portmap** daemon enter the following command:

```
stopsrc -s portmap
```

Files

Item	Description
inetd.conf	Starts RPC daemons and other TCP/IP daemons.
/etc/rpc	Contains a list of server names and their corresponding rpc program numbers and aliases.

portmir Command

Purpose

Allows one TTY stream (monitor) to attach to another TTY stream (target) and monitor the user session that is taking place on that stream.

Syntax

```
portmir { -d mir_modem -t target [ -m monitor ] | -t target [ -m monitor ] | { -o | -c monitor | -q }
```

Description

The **portmir** command allows one TTY stream (monitor) to attach to another TTY stream (target) and monitor the user session that is taking place on that stream. This is accomplished by pushing a special "mirror" module into both the target and monitor TTY streams.

Both the target and monitor TTYs receive a printed message on their respective displays when a monitoring session begins. The monitoring session can be terminated from either the target TTY, monitor TTY, or a third TTY not involved in the monitoring session. When the monitor is used in a non-service mode, both streams must be in the open state (that is, either a getty or active session must be taking place on each TTY) in order for the command to work. This is necessary to allow the pushing of the "mirror" streams module. The **portmir** command is supported for use with TTY devices only (PTS, TTY, LFT).

The terminal type, as defined in the TERM environment variable, must be the same for both the monitor and target TTY. The value of this environment variable must correspond to a valid entry in the **terminfo** database. An example terminal type would be **ibm3151** or **vt100**. The LFT is similar to the **vt100**. Terminal emulators such as **aixterm** are usually similar in function to **vt100**.

Although the console can be used as either the target TTY or the monitor TTY, using the console as the monitor TTY is not recommended. However, if the console is used as the monitor TTY, note that the console is first automatically redirected to the target TTY for the duration of the monitoring session. When the monitoring session is terminated, the console is redirected back to the device specified in the CuAt ODM database attribute **syscons**. If the console had been previously redirected, the redirection is not preserved.

Async devices that provide offloading of character processing may have problems if they are mirroring devices that rely on the line discipline (**ldterm**) to provide this function. An example of this would be the 128-port async adapter. Use the **chdev** command to disable the fastcook attribute if a port of a dissimilar adapter is monitored. Run the command as follows:

```
chdev -l tty1 -a fastcook -disable
```

Flags

Item	Description
-c <i>monitor</i>	Configures port for service boot by creating CuAt ODM database attribute portmir_monitor , which contains the device parameter as the value field. This device is used later as the default monitoring device when the portmir command is invoked in service mode (-s flag). Mirroring must be configured by the system administrator to execute at service boot time using the -c option. The target defaults to the device defined in the portmir_monitor attribute.
-d <i>mir_modem</i>	Sets monitoring port for dial-in purposes. Only the root user can issue the command with this flag. Ensure that /usr/share/mir_modem is linked to the correct modem setup file. /usr/share/mir_modem contains example files; you may need to create your own, depending on your type of modem.
-m <i>monitor</i>	Specifies monitoring device. If neither the -m option nor the -s option are specified, then the monitoring device defaults to the port on which the portmir command was run.
-o	Turns off monitoring and terminates the command.
-q	Queries the value set with the -c option.
-t <i>target</i>	Specifies target device to be monitored.

Security

Only a single mirror session may be running at any one time.

To mirror a port in the nonservice mode, place a list of users who may monitor them in a **.mir** file in your home directory (not required for the root user). When the **mirror** daemon begins running, the daemon checks to see who is on that port. It then checks to see if the user of the monitoring port is authorized to monitor that port.

The **.mir** file must have the format of a single user ID per line.



Attention: Running the **su** command to change to root user during a mirror session gives root authority to *both* 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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. After **user1** has placed **user2**'s login ID into `/u/user2/.mir` file, to mirror **user1** on target **tty1** from **user2** on monitor **tty2**, enter:

```
portmir -t tty1 -m tty2
```

2. To mirror target **tty1** to user on monitor **tty2** who is dialing in, enter:

```
portmir -t tty1 -m tty2 -d mir_modem
```

3. To set up mirroring for service boot, specifying the monitoring device during the service boot, enter:

```
portmir -c tty
```

4. To disable mirroring during the service boot, enter:

```
portmir -c off
```

5. To query the service boot mirroring device, enter:

```
portmir -q
```

Files

Item	Description
<code>/usr/share/modems/mir_modem</code>	Modem configuration file examples for setting up dial-in.
<code>/usr/sbin/portmir</code>	Contains the command file.

post Command

Purpose

Routes a message.

Syntax

```
post [ -alias File ... ] [ -format | -noformat ] [ -msgid | -nomsgid ] [ -filter File | -nofilter ] [ -width Number ] [ -verbose | -noverbose ] [ -watch | -nowatch ] File
```

Description

The **post** command routes messages to the correct destinations. The **post** command cannot be started by the user. The **post** command can be called only by other programs.

The **post** command searches a message for all components that specify a recipient's address and parses each address to check for the proper format. The **post** command then puts addresses into the standard format and calls the **sendmail** command. The **post** command also performs header operations, such as appending the `Date:` and `From:` components and processing the `Bcc:` component. The **post** command uses the *File* parameter to specify the name of the file to be posted.

Note: The **post** command may report errors when parsing complex addresses (for example, `@A:harold@B.UUCP`). If you use complex addresses, use the **spost** command instead of the **post** command.

Flags

Item	Description
-alias <i>File</i>	Searches the specified mail alias file for addresses. This flag may be repeated to specify multiple mail alias files. The post command automatically searches the /etc/mh/MailAliases file.
-filter <i>File</i>	Uses the header components in the specified file to copy messages sent to Bcc : recipients.
-format	Puts all recipient addresses into 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.
-msgid	Adds a message-identification component (such as Message-ID:) to the message.
-nofilter	Strips the Bcc : header from the message for the To : and cc : recipients. Sends the message with minimal headers to the Bcc : recipients. This flag is the default.
-noformat	Does not alter the format of the recipient addresses.
-nomsgid	Does not add a message-identification component to the message. This flag is the default.
-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.
-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.

Files

Item	Description
/etc/mh/MailAliases	Contains the default mail aliases.
/etc/mh/mtstailor	Contains MH command definitions.

pppattachd Daemon

Purpose

Attaches an asynchronous device stream to the PPP (Point to Point Protocol) subsystem. Can be invoked as a daemon or a normal process.

Syntax

To Use a Specific tty Port as a Connection (Runs as a Daemon):

```
pppattachd /dev/ttyPortNumber { client | server | demand } { ip | ipv6 | ip ipv6 } [ multilink ] [ connect "ConnectorProgram" ] [ inactive Seconds ] [ authenticate pap | chap ] [ peer pap | chap ] [ user Name ] [ remote HostName ] [ nodaemon ]
```

To Use Standard In and Standard Out as the tty Device (Runs as a Process):

```
pppattachd { client | server | demand } { ip | ipv6 | ip ipv6 } [ multilink ] [ inactive Seconds ] [ authenticate pap | chap ] [ peer pap | chap ] [ user Name ] [ remote HostName ] [ nodaemon ]
```

Description

The **pppattachd** daemon provides the mechanism to bind an asynchronous stream to the PPP subsystem. When placing an out going connection on a specific tty port, **pppattachd** becomes a daemon. When using stdin (standard in) and stdout (standard out) as the tty device for PPP communications **pppattachd** does not become a daemon. (It would be executed from the **\$HOME/.profile** upon login on a tty device.)

You can activate PAP or CHAP authentication with the **authenticate** and **peer** options. Use the **smit** command to create entries in either the **/etc/ppp/pap-secrets** or **/etc/ppp/chap-secrets** file. The **pppattachd** daemon uses the passwords in these files to authenticate the connection. It searches only the **/etc/ppp/pap-secrets** file for PAP authentication and the **/etc/ppp/chap-secrets** file for CHAP authentication.

The multilink option is to used to identify the PPP link as having several attachments between the two PPP peers. PPP packets are fragmented at one peer, sent over the multiple attachments, and then reconnected on the remote peer that must also support multilink. The maximum receive reconstruction unit (MMRU) and endpoint descriptor are set through SMIT on the PPP Link Configuration menu. MRRU is the maximum data size before fragmentation. The endpoint discriminator uniquely identifies the local system.

Errors and messages are logged using the **syslog** facility.

Options

Item	Description
authenticate pap chap	Defines the current system as the authenticator of either PAP or CHAP.
client server demand	Defines the type of subsystem connection to be bound to on the system running the daemon.
ip ipv6 ip ipv6	Specifies protocol types.
connect "ConnectorProgram"	Specifies the program to use to place an outgoing connection. The tty device opened is passed as stdin and stdout to the program. The /usr/sbin/pppdial command is a connector program that can be used.
inactive Seconds	Specifies the number (unsigned integer) of seconds to wait for inactivity on the link before terminating the connection. The default value is 0 (no timeout).
multilink	Identifies the PPP link as having a group of attachments connecting to two PPP peers.
nodaemon	Specifies to the attachment process that it is not to become a daemon. You must use this option for attachment processes that are invoked with demand connections.
peer pap chap	Defines the current system as the peer of either PAP or CHAP.

Item	Description
remote <i>HostName</i>	Defines the remote host name to be used for PAP authentication. An entry for <i>UserName RemoteHostName Password</i> must exist in /etc/ppp/pap-secrets file for a successful connection. This option only has meaning for PAP authentication on both the authenticator and peer.
user <i>Name</i>	Defines the user entry to use for PAP authentication. An entry for <i>UserName RemoteHostName Password</i> must exist in /etc/ppp/pap-secrets file for a successful connection. This option only has meaning for PAP authentication on the peer.

Exit Status

This command returns the following exit values:

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

Security

Access Control: Any User

Auditing Events: N/A

Examples

1. You want System A to act as a client to server System B. From System A enter:

```
/usr/sbin/pppattachd /dev/tty0 client ip connect "sysbconnector"
```

where `sysbconnector` is the connector program.

On System B, the user that logged in would have invoked from **\$HOME/.profile**:

```
exec /usr/sbin/pppattachd server ip 2>/dev/null
```

2. You want server System B to contact client System A. From System B enter:

```
/usr/sbin/pppattachd /dev/tty0 server ipv6 connect "sysaconnector"
```

where `sysaconnector` is the connector program.

On System A, the user that logged in would have invoked from **\$HOME/.profile**:

```
exec /usr/sbin/pppattachd client ipv6 2>/dev/null
```

3. You want System A to act as a client to server System B using PAP authentication. System B acts as the authenticator and System A is the peer to be authenticated. From System A enter:

```
/usr/sbin/pppattachd /dev/tty0 client ip ipv6 peer pap user username \  
connect "sysbconnector"
```

where `sysbconnector` is the connector program.

On System A, the **/etc/ppp/pap-secrets** file contains: `username * ppppassword`. On System B, the user that logged in would have invoked from **\$HOME/.profile**:

```
exec /usr/sbin/pppattachd server ip ipv6 authenticate pap 2>/dev/null
```


On System B, the **/etc/ppp/pap-secrets** file contains: `username * ppppassword`.

Files

Item	Description
/usr/sbin/pppattachd	Contains the pppattachd daemon.
/etc/ppp/attXXX.pid	Contains the process id. XXX is the pid, the content of the file is the network layer ID to which the attachment was bound. The user must belong to uucp group for the pid file to be created.

pppcontrold Daemon

Purpose

Controls startup and management of the PPP (Point to Point Protocol) subsystem.

Syntax

To Start and Stop by Using the System Resource Controller:

startsrc -s pppcontrold

stopsrc -s pppcontrold

Description

The **pppcontrold** daemon reads in the **/etc/ppp/lcp_config** and **/etc/ppp/if_conf** files to install and configure the PPP subsystem. SMIT should be used to generate both **/etc/ppp/lcp_config** and **/etc/ppp/if_conf**. To modify these files the user must have root authority or be a member of the uucp group. The configuration files are read at initialization where the appropriate streams modules are configured and loaded, and the tcpip network interface layers are installed into the system. After configuring the subsystem, the **pppcontrold** daemon monitors the streams associated with the IP and IPv6 interfaces to perform operations such as setting IP addresses, and the flags of the IP and IPv6 interface. The **pppcontrold** daemon terminates upon receipt of SIGTERM or when the **stopsrc** command is invoked. The preferred method of starting and stopping the **pppcontrold** daemon is with SRC (System Resource Controller). You must have root authority to run the src commands.

Errors and messages are logged using the **syslog** facility.

The **pppcontrold** daemon creates the **/etc/ppp/pppcontrold.pid** file, which contains a single line with the command process ID used to terminate the **pppcontrold** daemon.

Flags

None

/etc/ppp/lcp_config File

This file provides the configuration information required for the subsystem. These values are used to ensure proper allocation of storage at the time the subsystem is configured. It is important to configure just what is needed since these values define storage that is allocated within the kernel. Blank lines and lines beginning with a # (pound sign) are ignored in the configuration file. Do not use blank lines or lines beginning with # (pound sign) within the interface definition. Only use these lines between interface definitions.

Required Keywords

server_name *name* Name of this system. This name should be unique to the system. Ensure that the first 20 bytes of the name are unique.

Required Keywords

lcp_server #	Number of server connections. Represents the number of server connections that the subsystem will allow. Storage for all specified connections is allocated at the time the subsystem is configured. The minimum value is 0 and the maximum value is gated by the amount of memory in the system.
lcp_demand #	Specify the maximum number of demand links that you want the PPP LCP multiplexor to support. Set this value to the number of demand interfaces that you expect to configure. The default value is 0.
lcp_client #	Number of client connections. The minimum value is 0 and the maximum value is gated by the amount of memory in the system. Client connections are IP and IPv6 interfaces configured without addresses.
num_if #	Number of IP and IPv6 interfaces to configure. Must be less than or equal to lcp_server + lcp_client.
num_if6 #	Maximum number of TCP/IPv6 interfaces to allow. The value is a decimal number. This number, along with "max ip interfaces" and "max ip & ipv6 interfaces", cannot be greater than total maximum number of server, client and demand links (max server links + max client links + max demand links = max ip interfaces + max ipv6 interfaces + max ip & ipv6 interfaces). When a machine is used solely as a client connecting up to one server, this field would be set to 1. On a server, this field would be set to the maximum number of IPv6 clients that can simultaneously connect to the server. In this case, make sure that you have enough IPv6 interfaces defined.
num_if_and_if6 #	Maximum number of TCP/IP and IPv6 interfaces to allow. The value is a decimal number. This number along with "max ip interfaces" and "max ipv6 interfaces" cannot be greater than total maximum number of server, client and demand links (max server links + max client links + max demand links = max ip interfaces + max ipv6 interfaces + max ip & ipv6 interfaces). When a machine is used solely as a client connecting up to one server, this field would be set to 1. On a server, this field would be set to the maximum number of IP and IPv6 clients that can simultaneously connect to the server. In this case, make sure that you have enough IP and IPv6 interfaces defined.
num_hdlc #	Maximum number of concurrent asynchronous PPP sessions (server, client and demand) that can be active. This field is a decimal number. The value can not be greater than the total maximum number of server, client and demand links ([max server connections + max client connections + max demand connections] = max async hdlc attachments = [max ip interfaces + max ipv6 interfaces + max ip & ipv6 interfaces]).

Optional Keywords

These keywords will override the global default LCP options.

Item	Description
txacm <i>0xXXXXXXXX</i>	Transmit Asynchronous Character Map.
-negacm	Do not negotiate async character mapping. Rejects the peers configuration information frames that contains this option.

Item	Description
-negmru	Do not negotiate MRU (Maximum Receive Unit). Rejects the peers configuration information frames that contains this option.
mru #	MRU desired. A default is 1500.
-negacf	Do not negotiate ACF (address control field) compression. ACF will not be compressed. Rejects the peers configuration information frames that contain this option.
-negprotocolcompress	Do not negotiate protocol compression. Normally, the PPP protocol field will be compressed by one byte for Network protocols. This disables negotiation of this option for both receiving and sending frames.

/etc/ppp/if_conf File

This file defines all the server TCP/IP interfaces. Blank lines and lines beginning with a # (pound sign) are ignored in the configuration file. Do not use blank lines or lines beginning with # (pound sign) within the interface definition . Only use these lines between interface definitions.

Keywords

interface	Indicates that a new interface definition is being started.
ip and ipv6	Specifies the protocol or protocols used for this interface and will coincide with the local_ip, local_ip6, remote_ip, and remote_ip6 keywords. These keywords can be used alone or in combination.
server	Indicates that the interface is a server connection. Requires the following keywords: local_ip xxx.yyy.zzz.qqq remote_ip xxx.yyy.zzz.qqq local_ip6 ::XXXX:XXXX:XXXX:XXXX remote_ip6 ::XXXX:XXXX:XXXX:XXXX These addresses MUST be different on the pair basis, however the local IP and IPv6 address can be the same for all PPP interfaces. On a given server, the remote address must be unique. The "interface" "server" entry will contain only local_ip and remote_ip addresses if the smitty PPP IP Interfaces menu is used to configure the interface. remote_ip6 and local_ip6 will be seen in the entry if the smitty PPP IPv6 Interfaces menu is used. Finally, all four will be found if smitty PPP IP and IPv6 Interfaces is used.
client	This is an IPv6 option only. A client interface is required for all IPv6 connections. The address will be randomly generated based on the system model and ID. You can elect to zero out the address, (::0:0:0:0 or simply ::) and have the server assign an IPv6 address to the client. An example if_conf file entry follows:

```
interface
client
ipv6
local_ip6 ::0000:0000:0000:0000

interface
client
ip
ipv6
local_ip6 ::0007:0000:0000:4445
```

Keywords

demand There is a local_XXX and remote_XXX that are dependant on the protocol type (IP, IPv6 or both). A quoted command string is also required to establish connection with the authenticating host (server). An example **if_conf** file entry follows:

```
interface
demand
ipv6
local_ip6 ::0007:0000:0000:4444
remote_ip6 ::0009:0000:0000:5555
dcmd "exec /usr/sbin/pppattachd /dev/tty3 demand ipv6 >/dev/tty3 nodaemon"

interface
demand
ip
ipv6
local_ip 44.44.44.46
remote_ip 66.66.66.66
netmask 255.255.255.0
local_ip6 ::0007:0000:0000:4446
remote_ip6 ::0009:0000:0000:6666
dcmd "exec /usr/sbin/pppattachd /dev/tty4 demand ip ipv6 >/dev/tty4 nodaemon"
```

Optional Keywords

netmask xxx.xxx.xxx.xxx Specifies a netmask for an IPv4 interface.

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.

Examples

Example **/ect/ppp/lcp_config** File:

```
# Comment line
server_name pppclient
lcp_server 0
lcp_client 3
lcp_demand 2
num_if 1
num_if6 2
num_if_and_if6 2
num_hdlc 5
```

Example **/ect/ppp/if_conf** File:

```
# Sample ip server configuration information.
# Note that the complete stanza does not contain
# comments or blank lines
interface
server
ip
local_ip 129.35.130.45
remote_ip 129.35.131.191
netmask 255.255.240.0

# Sample ipv6 server configuration information.
# Note that the complete stanza does not contain
```

```

# comments or blank lines
interface
server
ipv6
local_ip6 ::0009:2313:4C00:3193
remote_ip6 ::0009:2313:4C00:3194

#However between stanzas one can have blank or
# comment lines.

interface
server
ip
local_ip 129.35.130.45
remote_ip 129.35.131.196
netmask 255.255.240.0

interface
server
ip
local_ip 129.35.130.45
remote_ip 129.35.131.197
netmask 255.255.240.0

interface
server
ip
local_ip 129.35.130.45
remote_ip 129.35.131.201
netmask 255.255.240.0

interface
server
ip
local_ip 129.35.130.45
remote_ip 129.35.131.212
netmask 255.255.240.0

```

The above configuration files would result in a subsystem that installs the IP and IPv6 interfaces as follows:

```

pp0: flags=71<UP,POINTOPOINT,NOTRAILERS>
inet 129.35.130.45 --> 129.35.131.191 netmask 0xfffff000
pp1: flags=31<UP,POINTOPOINT,NOTRAILERS>
inet 129.35.130.45 --> 129.35.131.196 netmask 0xfffff000
pp2: flags=31<UP,POINTOPOINT,NOTRAILERS>
inet 129.35.130.45 --> 129.35.131.197 netmask 0xfffff000
pp3: flags=31<UP,POINTOPOINT,NOTRAILERS>
inet 129.35.130.45 --> 129.35.131.201 netmask 0xfffff000
pp4: flags=31<UP,POINTOPOINT,NOTRAILERS>
inet 129.35.130.45 --> 129.35.131.212 netmask 0xfffff000
pp5: flags=30<POINTOPOINT,NOTRAILERS>
inet netmask

```

Note: pp5 is the result of the `lcp_client` keyword in the `/etc/ppp/lcp_config` file (`lcp_client 1`). Both IP and IPv6 client interfaces will have no address associated with them until a connection is established with the server and the IPs are negotiated through IPCP/IPV6CP. The only exception is with demand client interfaces. These interfaces will specify their own address and demand it during negotiation. As such, they will have their IP and IPv6 address associated with their interface as soon as the PPP subsystem is started.

Files

Item	Description
<code>/usr/sbin/pppcontrold</code>	Contains the pppcontrold daemon.
<code>/etc/ppp/lcp_config</code>	Configures the subsystem (lcp_config should be generated by SMIT).
<code>/etc/ppp/if_conf</code>	Configures the TCP/IP interfaces (if_conf should be generated by SMIT).
<code>/etc/ppp/pppcontrold.pid</code>	Contains the pppcontrold process id.
<code>/etc/ppp/ppp.conf</code>	Contains input to the strload command.

pppdial Command

Purpose

Establish an asynchronous connection with a remote system for use by the PPP (Point to Point Protocol) subsystem.

Syntax

```
pppdial [ -t TimeOut ] [ -v ] [ -d VerboseFile ] -f ChatFile
```

Description

The **pppdial** command provides the capability to establish a connection with a remote system over an asynchronous device. It is used with the **pppattachd** daemon as the means for carrying out the dialog with modems and remote systems to the point where PPP frames should be sent. The **pppdial** command uses standard input (stdin) and standard output (stdout) as the devices over which the dialog occurs.

Errors and messages are logged using the **syslog** facility.

Flags

Item	Description
-d <i>VerboseFile</i>	Logs the chat activity to <i>VerboseFile</i> . If <i>VerboseFile</i> does not exist, the pppdial command creates it. If <i>VerboseFile</i> does exist, the pppdial command appends the output to the existing file.
-f <i>ChatFile</i>	Specifies the file which contains the dialog that is to occur over the tty device. The content of <i>ChatFile</i> conforms to the syntax of the Basic Networking Utility (BNU)/UNIX to UNIX Copy Program (UUCP).
-t <i>TimeOut</i>	Specifies the number of seconds to wait before timing out during the Expect phase of the chat activity.
-v	Logs the chat activity using the syslog facility.

Exit Status

This command returns the following exit values:

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

Security

Access Control: Any User

Examples

To establish a connection with a remote system, enter on the command line in one line:

```
/usr/sbin/pppattachd client ip /dev/tty0 connect "/usr/sbin/pppdial  
-v -f /home/pppuser/dialer.file"
```

The *ChatFile* named `/home/pppuser/dialer.file` contains:

```
' '
atdt4311088
CONNECT
\\d\\n
ogin
pppuser
ssword
pppuserpwd
```

with the following meaning:

```
' '
atdt4311088      Expect a nul string
                  Send the modem the dial command
                  4311088 is the phone number to dial
CONNECT         Expect connect from the modem
\\d\\n          Delay for 1 second then send a new line
ogin            Expect the string ogin
pppuser        Send the string pppuser
                pppuser is the user id on the remote system
ssword         Expect the string ssword
pppuserpwd     Send the string pppuserpwd
                pppuserpwd is the password of the user pppuser on the
                remote system
```

The remote system must have a user `pppuser` defined with a password `pppuserpwd` and a **\$HOME/.profile** containing:

```
exec pppattachd server ip ipv6 2>/dev/null
```

This is a very simplistic example. The example requires that the PPP subsystem is running on both the client and server (or remote) system. The example requires that the client system have a modem defined on `/dev/tty0`. The *ChatFile* contains the number 4311088 to dial. The remote system must also have a user defined with a password and a **.profile** which starts a PPP attachment on the remote system. The device (`/dev/tty0`), phone number, user, user password and mechanism starting the PPP attachment are variable and should reflect the current values on the server system.

Files

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

pppstat Command

Purpose

Extracts and displays the RAS (Reliability, Availability, and Serviceability) information of the PPP (Point to Point Protocol) subsystem.

Syntax

pppstat

Description

The **pppstat** command provides the capability to monitor particular characteristics of active links. The following information is displayed for all active links:

LCP Multiplexing Layer

Item	Description
Local MRU	Specifies the Maximum Receive Unit setting for this host. This is maximum length of a packet that the remote host can send to the local host.
Remote MRU	Specifies the Maximum Receive Unit setting for the remote host. This is the maximum length of a packet that we can send to the remote host.
Local To Peer ACCM	Specifies the ASYNC Character Map used in the transmission of packets to the remote host.
Peer To Local ACCM	Specifies the ASYNC Character Map used by the remote host in the transmission of packets to the local host.
Local To Remote Protocol Field Compression	Specifies whether Protocol Compression is used in the transmission of packets to the remote host.
Remote To Local Protocol Field Compression	Specifies whether Protocol Compression is used in the transmission of packets from the remote host to the local host.
Local To Remote Address/Control Field Compression	Specifies whether Address/Control field compression is being used in the transmission of packets to the remote host.
Remote To Local Address/Control Field Compression	Specifies whether Address/Control field compression is being used in the transmission of packets from the remote host to the local host.

LCP Multiplexing Layer prior to PPP negotiating

Item	Description
MRU	Specifies the Maximum Receive Unit for receiving packets. This is the value that this host attempted to negotiate with the remote host.
Receive ACCM	Specifies the initial remote-to-local ASYNC Character Map that was used in the negotiation.
Transmit ACCM	Specifies the initial local-to-remote ASYNC Character Map that was used in the negotiation.
Magic Number	Specifies the magic number attempted in negotiation.
Frame Check Size	Specifies the length of the Frame Check Sequence that this host attempted to negotiate. This is fixed at 16 bits.

HDLC Framing Layer

Item	Description
Bad Address Fields	Specifies the number of times a packet has been received with an incorrect address field.
Bad Controls Fields	Specifies the number of times a packet has been received with an incorrect control field.
Oversized Packets	Specifies the number of times a packet has been received that has a length that exceeds the Maximum Receive Unit length.
Bad Frame Check Sequence	Specifies the number of times a packet has been received with a bad Frame Check Sequence.

Item	Description
Incoming Good Octets	Specifies the number of octets received in valid packets.
Outgoing Good Octets	Specifies the number of octets sent successfully in packets.
Incoming Good Packets	Specifies the number of packets received successfully.
Outgoing Good Packets	Specifies the number of packets sent successfully.

The output is sent to **stdout**. Messages are sent to **stderr**.

Exit Status

This command returns the following exit values:

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

Security

Access Control: Any User

Auditing Events: N/A

Files

Item	Description
/usr/sbin/pppstat	Contains the pppstat command.

pprof Command

Purpose

Reports CPU usage of all kernel threads over a period of time.

Syntax

```
pprof { time | -I pprof.flow | -i tracefile | -d } [ -T bytes ] [ -v ] [ -s ] [ -n ] [ -f ] [ -p ] [ -w ] [ -r PURR ] [ -@
[WparList | ALL ]
```

Description

The **pprof** command reports on all kernel threads running within an interval using the **trace** utility. The raw process information is saved to **pprof.flow**, and five reports are generated. The **pprof** command can also take previously generated **Pprof.flow** to regenerate reports. If no flags are specified, all reports are generated.

Types of Reports

Item	Description
pprof.cpu	<p>Lists all kernel level threads sorted by actual cpu time. Contains: Process Name, Process ID, Parent Process ID, Process State at Beginning and End, Thread ID, Parent Thread ID, Actual CPU Time, Start Time, Stop Time, Stop - Start</p> <p>The WPAR name is also provided when the -@ flag with no argument has been selected.</p>
pprof.start	<p>Lists all kernel threads sorted by start time. Contains: Process Name, Process ID, Parent Process ID, Process State Beginning and End, Thread ID, Parent Thread ID, Actual CPU Time, Start Time, Stop Time, Stop - Start</p> <p>The WPAR name is also provided when the -@ flag with no argument has been selected.</p>
pprof.namecpu	<p>Lists information about each type of kernel thread (all executable with the same name). Contains: Process Name, Number of Threads, CPU Time, % of Total CPU Time</p> <p>The WPAR name is also provided when the -@ flag with no argument has been selected.</p>
pprof.famind	<p>Lists all processes grouped by families (processes with a common ancestor). Child process names are indented with respect to the parent. Contains: Start Time, Stop Time, Actual CPU Time, Process ID, Parent Process ID, Thread ID, Parent Thread ID, Process State at Beginning and End, Level, Process Name.</p> <p>The WPAR name is also provided when the -@ flag with no argument has been selected.</p>
pprof.famcpu	<p>Lists the information for all families (processes with a common ancestor). The Process Name and Process ID for the family is not necessarily the ancestor. Contains: Start Time, Process Name, Process ID, Number of Threads, Total CPU Time.</p> <p>The WPAR name is also provided when the -@ flag with no argument has been selected.</p>

Flags

Item	Description
-d	Waits for the user to execute trcon and trcstop from the command line.
-f	Specifies to only generate the pprof.famcpu and pprof.famind reports.
-i tracefile	Indicates to generate reports from a tracefile . The trace must contain the following hooks: 135,106,10C,134,139,465,467,00A
-I pprof.flow	Indicates to generate reports from a previously generated pprof.flow . Specifies to only generate the pprof.namecpu report.
-n	Specifies to only generate the pprof.namecpu report.
-p	Specifies to only generate the pprof.cpu report.
-r PURR	Uses PURR time instead of TimeBase in percent and CPU time calculation. Elapsed time calculations are unaffected.
-s	Specifies to only generate the pprof.start report.

Item	Description
-T	Sets the trace kernel buffer size in bytes. The default is 32000.
-v	Sets verbose mode (print extra details).
-w	Specifies to only generate pprof.flow .
-@ [<i>WparList</i> ALL]	Displays WPAR information.
	ALL Lists all WPARs.
	WparList Specifies a comma-separated list of WPARs of interest.
<i>time</i>	Specifies the number of seconds to trace the system.

Note: Review the `/usr/lpp/perfagent/README.perfagent.tools` file for the latest on changes to the performance analysis tools.

pr Command

Purpose

Writes a file to standard output.

Syntax

```
pr [ +Page ][ -Column [ -a ] | -m ][ -d ][ -F ][ -r ][ -t ][ -e [ Character ] [ Gap ] ][ -h Header ]
[ -i [ Character ] [ Gap ] ][ -l Lines ][ -n [ Character ] [ Width ] ][ -o Offset ][ -s [ Character ] ][ -w Width ]
[ -x [ Character ] [ Width ] ][ -f ][ -p ] [ File ... | - ]
```

Description

The **pr** command writes the specified file or files to standard output. If you specify the **-** (minus sign) parameter instead of the *File* parameter, or if you specify neither, the **pr** command reads standard input. A heading that contains the page number, date, time, and name of the file separates the output into pages.

Unless specified, columns are of equal width and separated by at least one space. Lines that are too long for the page width are cut off. If standard output is a workstation, the **pr** command does not display error messages until it has ended.

Flags

Item	Description
-Column	Sets the number of columns to the value specified by the <i>Column</i> variable. The default value is 1. This option should not be used with the -m flag. The -e and -i flags are assumed for multicolumn output. A text column should never exceed the length of the page (see the -l flag). When the -Column flag is used with the -t flag, use the minimum number of lines to write the output.
+Page	Begins the display with the page number specified by the <i>Page</i> variable. The default value is 1.
-a	Modifies the effect of the -Column flag so that multiple columns are filled horizontally, from left to right. For example, if there are two columns, the first input line goes in column 1, the second goes in column 2, the third becomes line 2 of column 1, and so forth. If the -a flag is not specified, columns are created vertically.

Item	Description
-d	Produces double-spaced output.
-e [<i>Character</i>][<i>Gap</i>]	Expands tabs to character positions as follows: $Gap+1$, $2*Gap+1$, $3*Gap+1$, and so on. The default value of <i>Gap</i> is 8. Tab characters in the input expand to the appropriate number of spaces in order to line up with the next tab setting. If you specify a value for the <i>Character</i> variable (any character other than a digit), that character becomes the input tab character. The default value of the <i>Character</i> variable is the ASCII TAB character.
-F	Uses a form-feed character to advance to a new page. (Otherwise the pr command issues a sequence of line-feed characters.) Pauses before beginning the first page if the standard output is a workstation. This flag is equivalent to the -f flag.
-f	Uses a form-feed character to advance to a new page. (Otherwise the pr command issues a sequence of line-feed characters.) Pauses before beginning the first page if the standard output is a workstation. This flag is equivalent to the -F flag.
-h <i>Header</i>	Uses the specified header string as the page header. If the -h flag is not used, the page header defaults to the file name specified by the <i>File</i> parameter.
-i [<i>Character</i>][<i>Gap</i>]	Replaces white space wherever possible by inserting tabs to character positions, as follows: $Gap+1$, $2*Gap+1$, and $3*Gap+1$, and so forth. The default value of <i>Gap</i> is 8. If you specify a value for the <i>Character</i> variable (any character other than a digit), that character is used as the output tab character.
-l <i>Lines</i>	Overrides the 66-line default and resets the page length to the number of lines specified by the <i>Lines</i> variable. If the <i>Lines</i> value is smaller than the sum of both the header and trailer depths (in lines), the header and trailer are suppressed (as if the -t flag were in effect).
-m	Merges files. Standard output is formatted so the pr command writes one line from each file specified by the <i>File</i> parameter, side by side into text columns of equal fixed widths, based on the number of column positions. This flag should not be used with the <i>-Column</i> flag.
-n [<i>Character</i>][<i>Width</i>]	Provides line numbering based on the number of digits specified by the <i>Width</i> variable. The default is 5 digits. The line number occupies the first $Width+1$ column positions of each text column of default output, or of each line of output when the -m flag is set. If the <i>Character</i> variable is specified (any non-digit character), it is appended to the line number to separate it from what follows on the line. The default character separator is the tab character.
-o <i>Offset</i>	Indents each line by the number of character positions specified by the <i>Offset</i> variable. The total number of character positions per line is the sum of the width and offset. The default <i>Offset</i> value is 0.
-p	Pauses before beginning each page if the output is directed to a workstation. The pr command sounds the alarm at the workstation and waits for you to press the Enter key.
-r	Does not display diagnostic messages if the system cannot open files.
-s [<i>Character</i>]	Separates columns by the single character specified by the <i>Character</i> variable instead of by the appropriate number of spaces. The default value for the <i>Character</i> variable is an ASCII TAB character.

Item	Description
-t	Does not display the five-line identifying header and the five-line footer. Stops after the last line of each file without spacing to the end of the page.
-w <i>Width</i>	Sets the width of line to width column positions for multiple text-column output only. If the -w option is not specified and the -s option is not specified, the default width is 72. If the -w is not specified and the -s option is specified, the default width is 512. For single column output, input lines will not be truncated.
-x [<i>Character</i>] [<i>Width</i>]	Provides the same line numbering functions as the -n flag.

Exit Status

This command returns the following exit values:

Item	Description
0	All files were successfully written.
>0	An error occurred.

Examples

1. To print a file with headings and page numbers on the printer, type:

```
pr prog.c | qprt
```

This adds page headings to the **prog.c** file and sends it to the **qprt** command. The heading consists of the date the file was last modified, the file name, and the page number.

2. To specify a title, type:

```
pr -h "MAIN PROGRAM" prog.c | qprt
```

This prints the **prog.c** file with the title Main Program in place of the file name. The modification date and page number are still printed.

3. To print a file in multiple columns, type:

```
pr -3 word.lst | qprt
```

This prints the **word.lst** file in three vertical columns.

4. To print several files side by side on the paper:

```
pr -m -h "Members and Visitors" member.lst visitor.lst | qprt
```

This prints the **member.lst** and **visitor.lst** files side by side with the title Members and Visitors.

5. To modify a file for later use, type:

```
pr -t -e prog.c > prog.notab.c
```

This replaces tab characters in the **prog.c** file with spaces and puts the result in **prog.notab.c** file. Tab positions are at every eighth column (that is 9, 17, 25, 33, . . .). The **-e** flag tells the **pr** command to replace the tab characters; the **-t** flag suppresses the page headings.

Files

Item	Description
/usr/bin/pr	Contains the pr command.

Item	Description
<code>/dev/tty*</code>	Suspends messages.

praliases Command

Purpose

Displays mail aliases of the system.

Syntax

praliases [`-C file`] [`-f file`] [`key`]

Description

The **praliases** command displays current aliases of the system for each line, in no particular order. The special internal @: @ alias is displayed, if present.

Flags

Item	Description
<code>-C file</code>	Reads the specified sendmail configuration file instead of the default sendmail configuration file.
<code>-f file</code>	Reads the specified file instead of the configured aliases files of the sendmail file.
<code>key</code>	Displays entries that match the keys, if one or more keys are specified on the command line.

Note: The **praliases** command exits with 0 on success and with >0 if an error occurs.

Files

Item	Description
<code>/etc/mail/sendmail.cf</code>	Contains the default sendmail configuration file.

prctmp Command

Purpose

Displays the session record files.

Syntax

`/usr/sbin/acct/prctmp File...`

Description

A user with administrative authority can enter the **prctmp** command to display the session record file created by the **acctcon1** command, normally the `/var/adm/acct/nite/ctmp` file. The session record file is converted into the connect-time total accounting record by the **acctcon2** command and then incorporated into the daily accounting report.

Security

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

Example

To display the session record file, enter:

```
prctmp /var/adm/acct/nite/ctmp
```

This command displays the session record file created by the **acctcon1** command.

Files

Item	Description
/usr/sbin/acct	The path to the accounting commands.
/var/adm/acct/nite	Contains accounting data files.

prdaily Command

Purpose

Creates an ASCII report of the previous day's accounting data.

Syntax

```
/usr/sbin/acct/prdaily [ -X ] [ -l ] [ mmdd ] [ -c ]
```

Description

The **prdaily** command is called by the **runacct** command to format an ASCII report of the previous day's accounting data. The report resides in the **/var/adm/acct/sum/rprtmmdd** file, where *mmdd* specifies the month and day of the report.

Flags

Item	Description
-c	Reports exceptional resource usage by command. This flag may be used only on the current day's accounting data.
-l [<i>mmdd</i>]	Reports exceptional usage by login ID for the specified date. Use the <i>mmdd</i> variable to specify a date other than the current day.
-X	Processes all available characters for each user name instead of truncating to the first 8 characters. This flag will also cause the prdaily command to use the /var/adm/acct/sumx directory instead of the /var/adm/acct/sum directory.

Security

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

Files

Item	Description
/usr/sbin/acct	The path to the accounting commands.

Item	Description
<code>/usr/sbin/acct/ptelus.awk</code>	Calculates the limits for exceptional usage by login ID. This is a shell procedure.
<code>/usr/sbin/acct/ptecms.awk</code>	Calculates the limits of exceptional usage by command name. This is a shell procedure.
<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 long user name processing is requested.

preparevsd Command

Purpose

Makes a virtual shared disk available.

Syntax

```
preparevsd {-a | vsd_name...}
```

Description

The **preparevsd** command brings the specified virtual shared disks from the stopped state to the suspended state. The virtual shared disks are made available. Open and close requests are honored, while read and write requests are held until the virtual shared disks are brought to the active state. If they are in the suspended state, this command leaves them 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 **Prepare 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 stopped state are to be prepared.

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 **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*.

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 stopped state to the suspended state, enter:

```
preparevsd vsd1vg1n1
```

Location

/opt/rsct/vsd/bin/preparevsd

preprnode Command

Purpose

Prepares a node to be defined to a peer domain.

Syntax

```
preprnode [-k] [-h] [-TV] node_name1 [node_name2 ... ]  
preprnode -f | -F {file_name | "-" } [-k] [-h] [-TV]
```

Description

The `preprnode` command prepares security on the node on which the command is run so it can be defined in a peer domain. It allows for peer domain operations to be performed on this node and must be run before the node can join a peer domain using the `mkprdomain` or `addprnode` command.

Before the `mkprdomain` command is issued on a node, the `preprnode` command must be run on each node to be defined to the new peer domain, using the name of the node that is to run the `mkprdomain` command as the parameter. This gives the `mkprdomain` node the necessary authority to create the peer domain configuration on each new node and set up additional security.

Before the `addprnode` command is issued on a node, the `preprnode` command must be run on each node that is to be added, using the names of all online nodes as the parameters. This gives the online nodes the authority to perform the necessary operations on the new node.

The `preprnode` command performs the following:

1. Establishes trust with the node names specified on the command by adding their public keys to the trusted host list.
2. Modifies the resource monitoring and control (RMC) access control list (ACL) file to enable access to peer domain resources on this node from the other nodes in the peer domain. This allows peer domain operations to occur on the node. The RMC subsystem is refreshed so that these access changes will take effect.
3. RMC remote connections are enabled.

If the nodes that are to be defined to a peer domain are already in a management domain, you do not need to exchange public keys. You can use the `-k` flag to omit this step.

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.

-k

Specifies that the command should not exchange public keys.

-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 node (or nodes) from which peer domain commands can be accepted. Typically, this is the name of the node that will be running the `mkxpdomain` command when forming the peer domain. When adding to the peer domain, it is a list of the nodes that are currently online in the peer domain. The node name is the IP address or the long or short version of the DNS host name. The node name must resolve to an IP address.

Security

The user of the `prexnode` command needs write permission to the access control list (ACL) file. Permissions are specified in the ACL file. 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.

Restrictions

This command must run on a node that will be 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 "-"` 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. Suppose `mkripdomain` will be issued from `nodeA`. To prepare `nodeB`, `nodeC`, and `nodeD` to be defined to a new peer domain, `App1Domain`, run this command on `nodeB`, on `nodeC`, and then on `nodeD`:

```
preprnode nodeA
```

2. Suppose `nodeA` and `nodeB` are online in `App1Domain`. To prepare `nodeC` to be added to the existing domain, run this command on `nodeC`:

```
preprnode nodeA nodeB
```

Alternatively, create a file called `onlineNodes` with these contents:

```
nodeA  
nodeB
```

Then, run this command on `nodeC`:

```
preprnode -f onlineNodes
```

Location

`/opt/rsct/bin/preprnode`

Files

The access control list (ACL) file — `/var/ct/cfg/ctrmc.acls` — is modified. If this file does not exist, it is created.

prev Command

Purpose

Shows the previous message.

Syntax

```
prev [ +Folder ] [ -header | -noheader ] [ -showproc CommandString | -noshowproc ]
```

Description

The **prev** command displays the previous message in a folder. The **prev** command is similar to the **show** command with the **prev** value specified.

The **prev** command passes any flags that it does not recognize to the **showproc** program.

Flags

Item	Description
<code>+Folder</code>	Specifies the folder that contains the message you want to show.
<code>-header</code>	Displays a one-line description of the message being shown. The description includes the folder name and the message number. This flag is the default.
<code>-help</code>	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.
<code>-noheader</code>	Prevents display of a one-line description of each message.
<code>-noshowproc</code>	Uses the <code>/usr/bin/cat</code> command to list the previous command.
<code>-showproc</code> <i>CommandString</i>	Uses the specified command string to perform the listing.

Profile Entries

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

Item	Description
<code>Current-Folder:</code>	Sets the default current folder.
<code>Path:</code>	Specifies the <i>UserMhDirectory</i> .
<code>showproc:</code>	Specifies the program used to show 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 associated with this command, see the `lssecattr` command or the `getcmdattr` subcommand.

Examples

1. To display the previous message in the current folder, enter:

```
prev
```

The system responds with a message similar to the following:

```
(Message schedule: 10)
```

The text of the message is also displayed. In this example, message 10 in the current folder `schedule` is the previous message.

2. To show the previous message in the `meetings` folder, enter:

```
prev +meetings
```

The system responds with a message similar to the following:

```
(Message inbox: 5)
```

In this example, message 5 in the `meetings` folder is the previous message.

Files

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

printenv Command

Purpose

Displays the values of environment variables.

Syntax

```
printenv [ Name ]
```

Description

The **printenv** command displays the values of environment variables. If you specify the *Name* parameter, the system only prints the value associated with the *Name* parameter. If you do not specify the *Name* parameter, the **printenv** command displays the current environment, showing one *Name =Value* sequence per line.

If you specify a *Name* parameter that you have not defined in the environment, the **printenv** command returns an exit status of 1; otherwise it returns a status of 0 (zero).

Examples

1. To find the current setting of the **MAILMSG** environment variable, enter:

```
printenv MAILMSG
```

2. The command returns the value of the **MAILMSG** environment variable. For example:

```
YOU HAVE NEW MAIL
```

printf Command

Purpose

Writes formatted output.

Syntax

```
printf Format [ Argument ... ]
```

Description

The **printf** command converts, formats, and writes its *Argument* parameters to standard output. The *Argument* parameters are formatted under control of the *Format* parameter. The formatted output line cannot exceed **LINE_MAX** bytes in length.

The following environment variables affect the execution of the **printf** command:

Item	Description
LANG	Determines the locale to use for the locale categories when both LC_ALL and the corresponding environment variable (beginning with LC_) do not specify a locale.
LC_ALL	Determines the locale to be used to override any values for locale categories specified by the setting of LANG or any other LC_ environment variable.
LC_CTYPE	Determines the locale for the interpretation of sequences of bytes of text data as characters; for example, single versus multibyte characters in parameters.
LC_MESSAGES	Determines the language in which messages should be written.
LC_NUMERIC	Determines the locale for numeric formatting. This environment variable affects the format of numbers written using the e , E , f , g , and G conversion characters.

The *Format* parameter is a character string that contains three types of objects:

- Plain characters copied to the output stream.
- Conversion specifications, each of which cause 0 or more items to be retrieved from the value parameter list.
- The following escape sequences. When copied to the output stream, these sequences cause their associated action to be displayed on devices capable of the action:

Item	Description
------	-------------

\\	Backslash
-----------	-----------

\a	Alert
-----------	-------

\b	Backspace
-----------	-----------

\f	Form feed
-----------	-----------

\n	New line
-----------	----------

\r	Carriage return
-----------	-----------------

\t	Tab
-----------	-----

\v	Vertical tab
-----------	--------------

\ddd	Where <i>ddd</i> is a one-, two-, or three-digit octal number. These escape sequences are displayed as a byte with the numeric value specified by the octal number.
-------------	---

The *Argument* parameter is a list of one or more strings to be written to standard output under the control of the *Format* parameter.

The *Format* parameter is reused as often as necessary to satisfy the *Argument* parameters. Any extra **c** or **s** conversion specifications are evaluated as if a null string *Argument* were supplied; other extra conversion specifications are evaluated as if a 0 *Argument* were supplied. Where the *Format* parameter contains no conversion specifications and *Argument* parameters are present, the results are unspecified.

Each conversion specification in the *Format* parameter has the following syntax in this order:

1. A **%** (percent sign).
2. Zero or more options, which modify the meaning of the conversion specification. The option characters and their meanings are:

Item	Description
------	-------------

-	The result of the conversion is left-aligned within the field.
----------	--

+	The result of a signed conversion always begins with a sign (+ or -).
----------	---

Item	Description
------	-------------

- | | |
|--------------|--|
| blank | If the first character of a signed conversion is not a sign, a blank is prefixed to the result. If both the blank and + option characters are displayed, then the blank option character is ignored. |
| # | This option specifies that the value is to be converted to an alternate form. For c , d , i , u , and s conversions, the option has no effect. For o conversion, it increases the precision to force the first digit of the result to be a, 0 (zero). For x and X conversions, a nonzero result has 0x, or 0X prefixed to it, respectively. For e , E , f , g , and G conversions, the result always contains a radix character, even if no digits follow the radix character. For g and G conversions, trailing zeros are not removed from the result as they usually are. |
| 0 | For d , i , o , u , x , e , E , f , g , and G conversions, leading zeroes (following any indication of sign or base) are used to pad to the field width, no space padding is performed. If the 0 (zero) and the - (minus sign) options are displayed, the 0 (zero) option is ignored. For d , i , o , u , x , and X conversions, if a precision is specified, the 0 (zero) option is ignored. |

Note: For other conversions, the behavior is undefined.

- An optional decimal digit string that specifies the minimum field width. If the converted value has fewer characters than the field width, the field is padded on the left to the length specified by the field width. If the left-adjustment option is specified, the field is padded on the right. If the result of a conversion is wider than the field width, the field is expanded to contain the converted result. No truncation occurs. However, a small precision may cause truncation on the right.
- An optional precision. The precision is a . (dot) followed by a decimal digit string. If no precision is given, it is treated as 0 (zero). The precision specifies:
 - The minimum number of digits to be displayed for the **d**, **o**, **i**, **u**, **x**, or **X** conversions.
 - The number of digits to be displayed after the radix character for the **e** and **f** conversions.
 - The maximum number of significant digits for the **g** conversion.
 - The maximum number of bytes to be printed from a string in the **s** conversion.
- A character that indicates the type of conversion to be applied, such as:

Item	Description
------	-------------

- | | |
|-------------|--|
| % | Performs no conversion. Prints a % (percent sign). |
| a, A | Accepts a floating-point value and converts the value to a decimal notation that is in the format [-]0xh.hhhhp±d. The decimal notation contains one hexadecimal digit before the decimal point. This hexadecimal digit must be a non-zero value if the specified floating-point value is a normalized floating-point value or is unspecified. The number of hexadecimal digits after the decimal point indicates the precision value. If the precision value is not specified by using the <i>Format</i> parameter and the exponent value of the FLT_RADIX argument is 2, the precision value represents the floating-point value. If the precision value is not specified and the exponent value of the FLT_RADIX argument is not 2, the precision value can distinguish between the different floating-point values in the internal representation format that is used by the a, A conversion specifier. The trailing zeros in the decimal notation can be removed. If the precision value is zero and the # flag is not specified, the decimal point is not displayed. The letters abcdef are used for a conversion specifier and the letters ABCDEF are used for A conversion specifier. The A conversion specifier provides a number with the characters X and P instead of the characters x and p. To represent the decimal exponent of 2, the exponent of the FLT_RADIX argument must contain one digit to as many digits as needed. If the floating-point value is zero, the exponent value is also zero. The floating-point value that represents an infinity or NaN data type is converted to the format of the f, F conversion specifier. |

Item Description

- d, i** Accepts an integer value and converts it to signed decimal notation. The precision specifies the minimum number of digits to be displayed. If the value being converted can be represented in fewer digits, it is expanded with leading zeros. The default precision is 1. The result of converting a zero value with a precision of zero is a null string. Specifying a field width with a zero as a leading character causes the field width value to be padded with leading zeros.
- o** Accepts an integer value and converts it to signed octal notation. The precision specifies the minimum number of digits to be displayed. If the value being converted can be represented in fewer digits, it is expanded with leading zeros. The default precision is 1. The result of converting a zero value with a precision of zero is a null string. Specifying a field width with a zero as a leading character causes the field width value to be padded with leading zeros. An octal value for field width is not implied.
- u** Accepts an integer value and converts it to unsigned decimal notation. The precision specifies the minimum number of digits to be displayed. If the value being converted can be represented in fewer digits, it is expanded with leading zeros. The default precision is 1. The result of converting a zero value with a precision of zero is a null string. Specifying a field width with a zero as a leading character causes the field width value to be padded with leading zeros.
- x, X** Accepts an integer value and converts it to hexadecimal notation. The letters abcdef are used for the **x** conversion and the letters ABCDEF are used for the **X** conversion. The precision specifies the minimum number of digits to be displayed. If the value being converted can be represented in fewer digits, it is expanded with leading zeros. The default precision is 1. The result of converting a zero value with a precision of zero is a null string. Specifying a field width with a zero as a leading character causes the field width value to be padded with leading zeros.
- f, F** Accepts a floating-point value and converts it to a decimal notation in the format **[-]ddd.ddd**. The number of digits after the radix character or the decimal point is equal to the specified precision value. The **LC_NUMERIC** locale category determines the radix character that must be used in this format. If a precision value is removed from the floating-point value, six digits are added in the output after the radix character. If the precision value is 0 (zero), the radix character is not displayed. A floating-point value that represents an infinity data type is converted to the format **[-]inf** or **[-]infinity**. The format **[-]inf** or **[-]infinity** is implementation-dependent. A floating-point value that represents a NaN data type is converted to the format **[-]nan** (*n-char-sequence*) or **[-]nan**. The meaning of any *n-char-sequence* is implementation-dependent. The **F** conversion specifier provides the formats **INF**, **INFINITY**, or **NAN** instead of **inf**, **infinity**, or **nan**.
- e, E** Accepts a float or double value and converts it to the exponential form **[-] d.dde{+|-}dd**. There is one digit before the radix character (shown here as the decimal point) and the number of digits after the radix character is equal to the precision specification. The **LC_NUMERIC** locale category determines the radix character to use in this format. If no precision is specified, then six digits are output. If the precision is 0 (zero), then no radix character will be displayed. The **E** conversion character produces a number with E instead of e before the exponent. The exponent always contains at least two digits. However, if the value to be printed requires an exponent greater than two digits, additional exponent digits are printed as necessary.
- g, G** Accepts a float or double value and converts it in the style of the **f** or **e** conversion characters (or **E** in the case of the **G** conversion), with the precision specifying the number of significant digits. Trailing zeros are removed from the result. A radix character is displayed only if it is followed by a digit. The style used depends on the value converted. Style **g** results only if the exponent resulting from the conversion is less than -4, or if it is greater than or equal to the precision.
- c** Accepts a value as a string and prints the first character in the string.

Item Description

- s** Accepts a value as a string and prints characters from the string until the end of the string is encountered or the number of characters indicated by the precision is reached. If no precision is specified, all characters up to the first null character are printed.
- b** Accepts a value as a string, that may contain backslash-escape sequences. Bytes from the converted string are printed until the end of the string or number of bytes indicated by the precision specification is reached. If the precision is omitted, all bytes until the first null character are printed.

The following backslash-escape sequences are supported:

- The escape sequences previously listed above under the description of the *Format* parameter. These are converted to the individual characters they represented.
- The `\c` (backslash c) sequence, which is not displayed and causes the **printf** command to ignore any remaining characters in the string parameter containing it, any remaining string parameters, and any additional characters in the *Format* parameter.

Exit Status

This command returns the following exit values:

Item Description

m

0 Successful completion.

>0 An error occurred.

Examples

1. Enter the following command:

```
printf "%5d%4d\n" 1 21 321 4321 54321
```

This produces the following output:

```
 1 21
3214321
54321 0
```

The *Format* parameter is used three times to print all of the given strings. The 0 (zero) is supplied by the **printf** command to satisfy the last `%4d` conversion specification.

2. Enter the following command:

```
printf "%c %c\n" 78 79
```

This produces the following output:

```
7 7
```

3. The following example demonstrates how the **%%** format specifier can be used to print the date in an order different from the order of the arguments:

```
printf ("%1$s, %3$d. %2$s, %4$d:%5$.2d", weekday, month, day, hour, min);
Sunday, 3. July, 10:02
(weekday, day. month, hour:min)
```

Files

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

probevctl Command

Purpose

Changes and displays the ProbeVue dynamic tracing parameters and the ProbeVue sessions.

Syntax

```
probevctl [ -c attribute = value ] [ -C ] [ -f { milli|micro } ] [ -d sessionID ] [ -l ] [ -n attribute = value ] [ -p ] [ -s { probevue_session_id } ] [ -t ] [ -u user-list ] [ -T { show|start|stop|reset } ]
```

Description

The **probevctl** command changes and displays the ProbeVue dynamic tracing parameters, the per-processor trace buffer size, the consumed pinned memory, the user owning the session, the identifier of the process that started the session, and the information on whether the session has kernel probes for the ProbeVue sessions.

The following ProbeVue parameters are configurable:

- ProbeVue status (enabled/disabled).
- Maximum pinned memory (MB) allocated for all ProbeVue sessions.
- Maximum pinned memory (KB) allocated for a non-privileged user's ProbeVue session including the memory for the trace buffers.
- Number of concurrent ProbeVue sessions allowed for a regular user.
- Default size of the per-processor trace buffers (KB).
- The minimum period in milliseconds that a regular user can request the trace consumer to read from its trace buffers.
- The default period in milliseconds that the ProbeVue buffers will be read by the trace consumer.
- The size of the per-processor computation stack used by a ProbeVue session (KB).
- The minimum time interval allowed for global root user in interval probes.
- The percentage of memory that is allocated for the dynamic data structure.
- The size of the per-processor local table in KB.
- The number of page fault contexts for handling page faults.
- The maximum number of threads a ProbeVue session should support when it has thread local variables.
- The maximum size of per-CPU buffer, in bytes, used by a **net** probe action.
- The maximum time, in milliseconds, a **systrace** probe action can take when the action is started in interrupt context.
- The maximum time, in milliseconds, a **sysproc** probe action can take when the action is started in interrupt context.
- The maximum time, in milliseconds, an **io** probe action can take when the action is started in interrupt context.
- The maximum time, in milliseconds, a **net** probe action can take when the action is started in interrupt context.
- The maximum time, in milliseconds, a **CPU-bound interval** probe action can take when the action is started in the interrupt context.

Only the root user or the users having the **aix.ras.probevue.manage** authorization can update the ProbeVue parameters and view all the ProbeVue sessions. Otherwise, users can view only the sessions owned by themselves. Each session is displayed in the following format:

Sid	Pid	Uid	Buffer size in bytes	Consumed memory in bytes	Kernel probes	Profiling
<i>sid</i>	<i>pid</i>	<i>uid</i>	<i>bufsize</i>	<i>memory</i>	yes or no	yes or no

By default, the ProbeVue is enabled. Attempt to disable the ProbeVue when the ProbeVue sessions are active will fail.

Flags

-c

Specifies non-user ProbeVue parameters. Arguments to this flag must be separated by commas or enclosed in double quotation marks and separated by commas or spaces. If either the **-p** or the **-t** flag is not specified with this flag, new values will be made effective both in the present boot and next boot sessions. The valid *attribute-value* pairs are as follows:

trace= {on | off}

Specifies whether the ProbeVue must be enabled or disabled.

default_buffer_size=default_buffer_size

Specifies the default size of the per-processor trace buffers in KB. This is rounded to the next 4KB page.

max_total_mem_size=max_total_mem_size

Specifies the maximum pinned memory in MB consumable by the entire ProbeVue framework.

default_read_rate=default_read_rate

Specifies the default period in milliseconds that the ProbeVue buffers will be read by the trace consumer.

stack_size=stack_size_in_4Kpages

Specifies the size of the per-processor computation stack in KB. This will be rounded to the next 4KB page.

local_table_size=number

Specifies the size of the per-processor local table in KB. Half of the space allocated for the local table is used by temporary strings. The default value is set to 4 KB.

min_interval=interval in ms

Specifies the minimum time interval allowed for global root user in interval probes.

num_pagefaults=number

Specifies the number of page fault contexts for handling page faults. The specified number of page fault contexts are preallocated during ProbeVue framework initialization.

num_threads_traced=number

Specifies the maximum number of threads a ProbeVue session can support when it has thread local variables. The ProbeVue framework preallocates all the thread-local variables at the start of a session for the maximum number of threads that are specified with this attribute.

max_net_buf_size=number

Specifies the maximum size of per-CPU buffer, in bytes, used by a **net** probe action.

max_intr_systrcprb_time=number

Specifies the maximum amount of time, in milliseconds, that a probe action can take to run when the **systrace** probe action is started in interrupt context.

max_intr_sysprocprb_time=number

Specifies the maximum amount of time, in milliseconds, that a probe action can take to run when the **sysproc** probe action is started in interrupt context.

max_intr_ioprpb_time=number

Specifies the maximum amount of time, in milliseconds, that a probe action can take to run when the **io** probe action is started in interrupt context.

max_intr_netprpb_time=number

Specifies the maximum amount of time, in milliseconds, that a probe action can take to run when the **net** probe action is started in interrupt context.

async_stats_fetch_interval=number

Specifies the asynchronous fetch interval in milliseconds to fetch the system statistics. This attribute is a global value that is applicable for all ProbeVue sessions. ProbeVue sets the asynchronous fetch interval based on this value. The default value for the tunable is 1000 milliseconds. Changes in the tunable value does not affect the running sessions and the specified value is used only for the new sessions.

fetch_stats_async_only={yes | no}

Specifies that all system statistics must be fetched in the asynchronous mode even if the synchronous fetch is possible. The default value for this parameter is no. Changes in the tunable value does not affect the running sessions and the specified value will be used only for the new sessions.

max_intr_cpuboundprpb_time=number

Specifies the maximum amount of time, in milliseconds, that a probe action can take to run when the **CPU-bound interval** probe action is started in interrupt context.

-C

Sets the ProbeVue session tunable to initial values. The **-C** option uses the current configuration of system to determine configuration initial values (High Config or Low Config) to be set and updates both current and next boot parameters. For more information about high and low configuration values of tunables, see the [ProbeVue dynamic tracing facility](#) topic.

Note: Before using the **-C** flag, ensure there is no active ProbeVue session.

-d sessionId

Displays the list of probes enabled for the specified session. When you specify all as the session ID, then the probes for all the ProbeVue sessions that can be viewed by the user is displayed. A list of ProbeVue sessions and the associated session ID can be obtained using the **probevctrl** command.

-f

Specifies the format in which time consumed data for probe actions needs to be displayed. The **-f** option can be used along with the **-T** option and it shows action on a specific session. The possible formats follow:

milli

Displays the time in milliseconds.

micro

Displays the time in microseconds.

The default format is **milli**.

-l

Lists the present value of the ProbeVue configuration parameters. If the **-p** or the **-t** flag is not specified, parameter values for the present boot session are displayed.

-n

Specifies the configurable parameters for regular users. Arguments to this option must be separated by commas or enclosed in double quotation marks and separated by commas or spaces. If either the **-p** or the **-t** flag is not specified with this flag, new values will be made effective both in the present boot and next boot sessions. The valid *attribute-value* pairs are as follows:

max_mem_size=max_mem_size

Specifies the maximum pinned memory in MB consumable by a ProbeVue session.

max_sessions=max_sessn

Specifies the maximum concurrent sessions allowed.

min_read_rate=min_read_rate

Specifies the minimum period in milliseconds that a regular user can request the trace consumer to read from its trace buffers.

pin_mem_dvar_pc=pin_mem_dvar_pc

Specifies the percentage of memory that can be allocated to the dynamic data structure for dynamic type variables. This memory can be used for stack trace and associative array type dynamic variables. The value of this parameter is set in the range 10-100. The default value is 50.

-p

Specifies that the default values for the next boot must be updated and displayed.

-s probevue_session_id

Specifies the session on which the action is to be performed. This flag must be used with the **-T** flag.

-u

Specifies comma-separated user list whose ProbeVue sessions must be listed. If the **-u** flag is not specified, all of the ProbeVue sessions that the user can view are displayed. A user with the **aix.ras.probevue.manage** authorization can view all of the ProbeVue sessions in the system. Users without this authorization can view only the ProbeVue sessions they own.

-t

Specifies that the default values for the present boot session must be updated and displayed.

-T show | start | stop | reset

Specifies whether to show, start, stop, or reset the probe action profiling for the session that is specified by the **-s** flag. The **-s** flag must be used along with this flag.

Examples

1. To modify the next boot default buffer size and to turn on the dynamic tracing, enter:

```
probevctrl -c trace=on,default_buffer_size=8 -p
```

or

```
probevctrl -c "trace=on default_buffer_size=8" -p
```

2. To list the next boot ProbeVue configuration, enter:

```
probevctrl -l -p
```

3. To list the present ProbeVue configuration, enter:

```
probevctrl -l -t
```

4. To list all of the ProbeVue sessions, enter:

```
probevctrl
```

5. To list all of the ProbeVue sessions owned by the user guest, enter:

```
probevctrl -u guest
```

6. To increase the percentage of pinned memory that is allocated for the dynamic data structures (stack trace and associative array) for the next boot from a default 50 -75, enter:

```
probevctrl -n pin_mem_dvar_pc = 75
```

probevue Command

Purpose

Starts a dynamic trace session. The command can preprocess the header file and exit without starting the dynamic trace session.

Syntax

```
probevue [ -c "{ timestamp = { 0 | 1 | 2 } thread = { on | off } tid = { t1, ... } pid = { p1, ... }  
abs_mem_for_dvars=memory_in_KB num_threads_traced=number_of_thread_to_trace}" ] [ -d ] [ -i  
Async_Fetch_Interval ] [ -e Pinned_memory_dvar_percent ] [ -f to_print_time_profile_data_milli_or_micro ]  
[ -I Include_file1, ... ] [ -K ] [ -o Output_file ] [ -q info={none|normal|detail} ] [ -s Buffer_size ] [ -t  
Interval ] [ -T ] [ -u ] [ -X Program_name [ -A "Arguments_to_program" ] [ -L "Lib_path" ] [ -g ] [ Script_name  
[ Arguments_to_script ] ]
```

```
probevue [ -P C ++_header_file ]
```

```
probevue [ -l "{ syscall | syscallx | syscallx32 | syscallx64 | interval | systrace | sysproc | io | net}" ]
```

Description

The **probevue** command analyzes the operating system and user programs by dynamically enabling the user-specified probes, starting the actions that are associated with the probes when they are triggered, and presenting the captured trace data.

When you specify the **probevue** command with a vue script, the command enables the tracing that was specified in the script, and produces the tracing output.

When the **-P** option is specified with the C++ header file, the command produces the preprocessed encrypted C header file. The encrypted C header file can be further used to probe C++ application by using the **-I** option of the **probevue** command.

The arguments to the **probevue** command and the vue script can be specified in a script instead of the command line. The script can be run repeatedly with same arguments by using this script. A vue language construct `#VUE_CMD_ARGS` can be used to specify the arguments to the **probevue** command in the file and the `#VUE_SCRIPT_ARGS` language construct can be used to specify the arguments to the vue script.

Notes:

- When the arguments are specified in the script, all the arguments for a vue construct must be on the same line.
- If arguments to the **probevue** command are specified by command line and in the script, only command line arguments are considered and all command arguments in the script are ignored. This process applies to the vue script arguments also.
- The `#VUE_CMD_ARGS` and `#VUE_SCRIPT_ARGS` constructs do not support standard input.
- The dynamic memory requirement of the **probevue** command is proportional to the product of number of CPUs and per-CPU trace buffer size (the value of the **default_buffer_size** tunable parameter of the **probevctrl** command). Hence, in a system that has large number of CPUs and higher value of per-CPU buffer can cause the **probevue** command to exceed the memory limit set by the **ulimit** parameter. In such scenarios, run the **probevue** command with the **-u** flag to cross that limit.

Flags

Item	Description
-A "Arguments_to_program"	Specifies the arguments to the program that you specified to using the -X flag. If there are multiple arguments to the application, enclose each argument in quotation marks.

Item	Description
-g	Turn on grouping of output from every clause of the probevue session i.e. all the output from a clause will appear together without getting interleaved by output from any other clause that is running simultaneously in another CPU. The -g flag overrides the <code>group_output_start()</code> or the <code>group_output_end()</code> statements present in the VUE script. Output of every clause is grouped from clause start to clause end irrespective of the <code>group_output_start()</code> or the <code>group_output_end()</code> statements present in the clauses.
-c	<p>Specifies how the trace data needs to be formatted. You must enclose arguments to this option in quotation marks and separate each argument by spaces. The options are as follows:</p> <p>timestamp={0 1 2} Controls the reporting of the time stamp that is associated with an event in the trace report. Specify one of the following values:</p> <p>0 Displays the timestamp, in seconds and microseconds, for each message relative to the beginning of the trace. The first line of the trace output shows the base time from which the individual time stamps are measured.</p> <p>1 With each message, displays the actual time taken to create the message.</p> <p>2 With each message, displays the actual time taken to create the message as per the <i>printf's %A</i> format.</p> <p>Note: If both options are desired then 0,1 must be entered. That is, there must be no spaces between 0,1.</p> <p>thread={on off} Displays the thread ID which generated the message, with each message. The default value is off.</p> <p>pid={p1,..} Displays only the messages that were generated by the processes specified.</p> <p>Note: If the thread has died before the trace consumer tries to know the process to which the thread belongs, or if the process that you specified no longer exists, the consumer cannot display the messages that were generated by the threads in this process, when you filter the messages by the process ID.</p> <p>tid={t1,..} Displays only the messages that were generated by the threads that you specified.</p> <p>abs_mem_for_dvars=memory_in_KB Specifies the pinned memory, in kilo bytes (KB), that is allocated for dynamic type variables. This option is mutually exclusive to the -e option.</p> <p>num_threads_traced=number_of_thread_to_trace Specifies the maximum number of threads that the current ProbeVue session can support when the session has thread-local variables. This value overrides the corresponding global ProbeVue tunable value.</p>
-d	Displays the list of probes enabled for the session.

Item	Description
-e <i>Pinned_memory_dvar_percentage</i>	Specifies the percentage of the dynamic data structure memory allocated for dynamic type variables. A minimum of 10 and a maximum of 100 value can be specified as the percentage.
-f	Specifies the format in which the time taken by probe action must be displayed. The supported formats follow: milli Displays time in milliseconds. micro Display time in microseconds. The default format is milli .
-i <i>Async_Fetch_Interval</i>	Specifies how often the asynchronous statistics is fetched for the probevue command. This option overrides the global interval time for the probevue command. The minimum interval is 100 milliseconds.
-I <i>Include_file1</i>	Uses the file specified as a post-processed header file, that is one with no C-preprocessor operators. It can be passed through the command line to be included when compiling the vue script.
-K	Enables RAS events related functionality in a probeVue session.
-l	Lists all the probe points supported by the probe manager. When you specify the -l flag with the probevue command, no other flags must be used. You can specify more than one probe manager with the -l flag, such as <code>-l syscall -l syscallx -l interval</code> . The probe manager supports interval , syscall , syscallx , systrace , io , sysproc , and net probes for the -l flag. If you specify wrong arguments or an incorrect probe manager with the -l option, a usage error is displayed. <ul style="list-style-type: none"> • <code>probevue -l syscall</code>: Lists all the possible system call that can be traced on the system. • <code>probevue -l syscallx</code>: Displays all base system calls that can be traced on the system. This option lists the system call separately for the 32 and 64-bit systems. • <code>probevue -l syscallx32</code>: Displays the 32-bit base system calls that can be traced on the system. • <code>probevue -l syscallx64</code>: Displays the 64-bit base system calls that can be traced on the system. • <code>probevue -l interval</code>: Specifies the minimum and maximum interval duration supported for regular and root users with the interval probe. • <code>probevue -l systrace</code>: Displays a description about the systrace probe. <p>Note: For <code>syscallx</code> probe manager, when the -l syscallx probe is used, it displays both 32-bit and 64-bit base-system calls. To view only 32-bit calls, use syscallx32 probes, and to view only 64-bit calls, use syscallx64 probes.</p>

Item	Description
-L "Lib path"	<p>Prioritizes library search in the specified path. If the libraries are not found in the specified path, searches libraries in the default library path that is saved in the header of the <code>Loader</code> section in the eXtended COFF (XCOFF) file.</p> <p>The -L flag is applicable to the user function tracing (uft) and C++ probe managers that use the name of the executable file. The -L flag is ignored for other probe managers.</p>
-o <i>Output_file</i>	Writes the report to a file rather than to the standard output.
-P <i>C++ header file</i>	<p>Preprocesses the C++ header file and creates an output preprocessed file for each input C++ header file. The preprocessed output file has the same name as the input C++ header file, with a <code>.Vue</code> suffix.</p> <p>Note: You cannot use other flags with the -P option. The -P flag accepts any file name, except the file name with a <code>.Vue</code> suffix.</p>
-q <i>info=level</i>	<p>Specifies reporting level of informational messages while parsing the vue script. The possible values follow:</p> <p>none No informational messages are displayed.</p> <p>normal Only important informational messages are displayed.</p> <p>detail All informational messages that can result in incorrect execution are reported.</p>
-s <i>Buffer_size</i>	Specifies the size of the per-CPU trace buffers in KB. This is rounded to the next 4K page. You can use following levels:
-t <i>Interval</i>	<p>Specifies how often the trace buffers are read. The minimum interval that you can specify is 10 milliseconds. The time interval specified by the regular user (that is a user without the aix.ras.probevue.trace privilege) is rounded to the next highest multiple of 10 milliseconds. The read rate is retrieved from the probeVue configuration.</p> <p>Note: A regular user can specify the minimum read rate and the probevctrl command can change the default read rate.</p>
-T	Starts probe action profiling at the start of session. The -T flag ensures that probe actions are profiled when the session is started.
-u	Starts probeVue session and sets the data segment to unlimited value. The -u flag is used to print stack traces from multiple processes of large binaries.
-X <i>Program_name</i>	Starts a program and enables probes before the program starts. You can use the special environment variables \$_CPID and \$_CTID within a vue script to identify the process ID and the thread ID of the application that is launched.

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 start a **probeVue** session with script **syscall.e**, enter:

```
probevue syscall.e
```

2. To send the trace report to the **/tmp/trace_report** file, enter:

```
probevue -o /tmp/trace_report syscall.e
```

3. To display the trace report of the thread IDs 12345,4567 and the timestamp relative to the beginning of trace, enter:

```
probevue -c "timestamp=0 tid=12345,4567" syscall.e
```

4. To include the header file **stat.i** and allocate 4K of per-CPU buffer, enter:

```
probevue -I stat.i -s 4 syscall.e
```

5. To preprocess the C++ header file **myheader.h** , enter:

```
probevue -P myheader.h
```

The **probevue** command generates the **myheader.Vue** file, which is an encrypted **C++** header file and is included in the trace session by using the **-I** option.

6. To increase the percentage of pinned memory for the current session of the dynamic data structures (stack trace and associative array), from a default of 50 -75 for the **ASO.e** script, enter:

```
probevue -e 75 ASO.e
```

7. The following script is an example of providing the arguments in the script:

```
#!/usr/bin/probevue
#VUE_CMD_ARGS=-o /tmp/trace_out
#VUE_SCRIPT_ARGS=read

@@syscall:*:$1:entry
{
    printf("%t\n", get_stktrace(4));
}
```

The script runs as: `./script.e`

Files

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

proccred Command

Purpose

Prints the credentials (effective, real, saved user IDs and group IDs) of processes.

Syntax

proccred *ProcessID* ...

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/ProcessID** strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like **procrun** and **procstop** start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **proccred** command prints the credentials (effective, real, saved user IDs and group IDs) of processes.

Flags

Item	Description
<i>ProcessID</i>	Specifies the process id.

Examples

1. To display the credentials of process 5046, enter:

```
proccred 5046
```

Files

Item	Description
/proc	Contains the /proc filesystem.

procfiles Command

Purpose

Reports information about all file descriptors opened by processes.

Syntax

```
procfiles [ -F ] [ -n ] [ -c ] ProcessID ...
```

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The **proctools** commands provide ASCII reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/ProcessID** strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The **proctools** commands like **procrun** and **procstop** start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

Regular files have permission based on mode it was opened with. Any non-regular files have 0 access mode.

The **procfiles** command reports information on all file descriptors opened by processes. With the **-n** option it also displays the names of the corresponding files.

Flags

Item	Description
-c	Prints the output in column format.
-F	Forces procfiles to take control of the target process even if another process has control.
-n	Prints the names of the files referred to by file descriptors.
<i>ProcessID</i>	Specifies the process id.

Examples

1. To display status and control information on the file descriptors opened by process 11928, enter the following command:

```
procfiles 11928
```

The output of this command might look like this:

```
11928 : -sh
Current rlimit: 2000 file descriptors
0: S_IFCHR mode:0622 dev:10,4 ino:2584 uid:100 gid:100 rdev:28,1
O_RDONLY
1: S_IFCHR mode:0622 dev:10,4 ino:2584 uid:100 gid:100 rdev:28,1
O_RDONLY
2: S_IFCHR mode:0622 dev:10,4 ino:2584 uid:100 gid:100 rdev:28,1
O_RDONLY
63: S_IFREG mode:0600 dev:10,8 ino:311 uid:100 gid:100 rdev:40960,10317
O_RDONLY size:2574
```

2. To display name, status and control information on the file descriptors opened by process 15502, enter the following command:

```
procfiles -n 15502
```

The output of this command might look like this:

```
15502 : /home/guest/test
Current rlimit: 2000 file descriptors
0: S_IFCHR mode:0622 dev:10,4 ino:2584 uid:100 gid:100 rdev:28,1
O_RDONLY
1: S_IFCHR mode:0622 dev:10,4 ino:2584 uid:100 gid:100 rdev:28,1
O_RDONLY
2: S_IFCHR mode:0622 dev:10,4 ino:2584 uid:100 gid:100 rdev:28,1
O_RDONLY
3: S_IFREG mode:0644 dev:10,7 ino:26 uid:100 gid:100 rdev:0,0
O_RDONLY size:0 name:/tmp/foo
```

3. To display status and control information on the file descriptors opened by the 278684 process, enter the following command:

```
procfiles -c 278684
```

The output of this command might look like this:

```
278684 : -ksh
Current rlimit: 2000 file descriptors
```

FD	TYPE	MODE	DEV/RDEV	UID	GID	OPMOD	INODE
0	c	-----	10, 4(19, 0)	root	system	R-W	16385
1	c	-----	10, 4(19, 0)	root	system	R-W	16385
2	c	-----	10, 4(19, 0)	root	system	R-W	16385
61	-	rw-r--r--	10, 7	root	system	R-W	32
63	-	rw-----	10, 4	root	system	R-W A	1051

Files

Item	Description
<code>/proc</code>	Contains the <code>/proc</code> filesystem.

procflags Command

Purpose

Prints the `/proc` tracing flags, the **pending** and **held** signals, and other `/proc` status information for each thread in the specified processes.

Syntax

```
procflags [ -r ] ProcessID ...
```

Description

The `/proc` filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or `/proc/ProcessID` strings as input. The shell expansion `/proc/*` can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from `/proc` for the specified processes and displays it to the user. The proctools commands like `procrun` and `procstop` start and stop a process using the `/proc` interface.

The information gathered by the commands from `/proc` is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procflags** command prints the `/proc` tracing flags, the pending and held signals, and other `/proc` status information for each thread in the specified processes. The machine register contents are printed when option `-r` is used and the process is stopped on an event of interest. The events of interest are **PR_REQUESTED**, **PR_FAULTED**, **PR_SYSENTRY**, and **PR_SYSEXIT** as defined in `<sys/procfs.h>`.

Flags

Item	Description
<code>-r</code>	Displays the current machine registers state if a process is stopped in an event of interest.
<i>ProcessID</i>	Specifies the process id.

Examples

1. To display the tracing flags of process 5046, enter:

```
procflags 5046
```

The output of this command might look like this:

```
5046 : -sh
data model = _ILP32 flags = PR_FORK
/4289: flags = PR_ASLEEP | PR_NOREGS
```

2. To display the tracing flags and registers values of process 5040 which was stopped on an event of interest, enter:

```
procflags -r 5040
```

The output of this command might look like this:

```
5040 : ls
data model = _ILP32 flags = PR_FORK
/6999: flags = PR_STOPPED | PR_ISTOP
why = PR_FAULTED what = FLTBPT what = kfork
gpr0 = 0x0 gpr1 = 0x2ff227b0 gpr2 = 0xf0083bec
gpr3 = 0x2ff22cb3 gpr4 = 0x11 gpr5 = 0x65
gpr6 = 0x50 gpr7 = 0x0 gpr8 = 0x41707a7c
gpr9 = 0x4c4f47 gpr10 = 0x80000000 gpr11 = 0x34e0
gpr12 = 0x0 gpr13 = 0xdeadbeef gpr14 = 0x1
gpr15 = 0x2ff22c0c gpr16 = 0x2ff22c14 gpr17 = 0x0
gpr18 = 0xdeadbeef gpr19 = 0xdeadbeef gpr20 = 0xdeadbeef
gpr21 = 0xdeadbeef gpr22 = 0x10 gpr23 = 0xfd
gpr24 = 0x2f gpr25 = 0x2ff227f0 gpr26 = 0x0
gpr27 = 0x2ff22d87 gpr28 = 0x2ff22cb3 gpr29 = 0x0
gpr30 = 0x0 gpr31 = 0xf0048260 iar = 0xd01be900
msr = 0x2d032 cr = 0x28222442 lr = 0xd01d9de0
ctr = 0xec xer = 0x0 fpscr = 0x0
fpscr = 0x0
```

Files

Item	Description
<code>/proc</code>	Contains the <code>/proc</code> filesystem.

procldd Command

Purpose

Lists the objects loaded by processes, including shared objects explicitly attached using `dlopen()`.

Syntax

```
procldd [ -F ] ProcessID ...
```

Description

The `/proc` filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The `proctools` commands provide ASCII reports based on some of the available information.

Most of the commands take a list of process IDs or `/proc/ProcessID` strings as input. The shell expansion `/proc/*` can therefore be used to specify all processes in the system.

Each of the `proctools` commands gathers information from `/proc` for the specified processes and displays it to the user. The `proctools` commands like `procrun` and `procstop` start and stop a process using the `/proc` interface.

The information gathered by the commands from `/proc` is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procldd** command lists the objects loaded by processes, including shared objects explicitly attached using **dlopen()**. All the information needed is gathered from the **/proc/ProcessID/map** files.

Flags

Item	Description
-F	Forces procldd to take control of the target process even if another process has control.
<i>ProcessID</i>	Specifies the process id.

Examples

1. To display the list of objects loaded by process 12644, enter:

```
procldd 12644
```

The output of this command might look like this:

```
12644 : -ksh
ksh
/usr/lib/libiconv.a[shr4.o]
/usr/lib/libi18n.a[shr.o]
/usr/lib/nls/loc/en_US
/usr/lib/libcrypt.a[shr.o]
/usr/lib/libc.a[shr.o]
```

Files

Item	Description
/proc	Contains the /proc filesystem.

procmmap Command

Purpose

Prints the address space map of processes.

Syntax

```
procmmap [ -F ] [ -S ] { -X [ -f ] [ -n ] [ -u ] [ -q ] } ProcessID ...
```

Description

The **/proc** file system provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide *ascii* reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/ProcessID** strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like the **procrun** and **procstop** start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procmap** command prints the address space map of processes. It displays the starting address and size of each of the mapped segments in the process. It gets all the information necessary from the **/proc/ProcessID/map** files.

Flags

Item	Description
-F	Forces the procmap command to take control of the target process even if another process has control.
-S	Displays shared memory information of the target process. Note: The -S option is obsolete. You can use the -X option for similar functionality.
-X	Displays extended information about each address range and includes all mapped objects of the target process.
-n	Displays unsorted address ranges for the mapped segments of the target process.
-u	Displays the values in the most appropriate unit. The unit used in the report is specified for each metric.
-q	Suppresses the header information.
-f	Displays the filename instead of device number and inode number for the mapped file of the target process. The <i>MAPPED OBJECT</i> column for the mapped files displays one of the following values: <ul style="list-style-type: none"> • <code>dev:remote</code> for the remote files. • <code>dev:remote</code> from within a WPAR using global host storage. • <code><full pathname></code> from within a WPAR using direct storage. • <code>dev:<major#>, <minor#>, ino:<inode#></code> outside a WPAR using direct storage. • For the non-WPAR local files, the file names is resolved, or the string <code>dev<major#>, <minor#>, ino:<inode#></code> is displayed with the correct major, minor, and inode numbers for the file.
<i>ProcessID</i>	Specifies the process id.

The following are brief descriptions of the contents of the columns with **-X** flag:

Item	Description
Start-ADD	Start address of the memory region.
End-ADD	End address of the memory region.
SIZE	Size of the memory region.
MODE	Read, write and execute permission of the memory region.
PSIZ	Page size of the memory region. The following values are displayed:
s	4K page size
m	64K page size
L	16M page size
H	16GB page size
TYPE	Type of the memory region. The following values are displayed:

Item	Description
HEAP	Heap Region
KER/LDR	Other Kernel/Loader Segments
KERTXT	Kernel Text
MAINDATA	Main Data/BSS for the process
MAINTEXT	Main Executable for the process
MFILE	Map File
PLIBDATA	Private Library Data
PLIBTEXT	Private Library Text
PMMAP	Private Memory Map
PMMFILE	Private Memory Map file
PMMPXRT	Private POSIX Real Time Shared Memory Map
REALMEM	Real Memory Mapped
SHM	Shared Memory
SHMEXT	Extended Shared Memory
SHMFILE	Shared Memory File
SLIBDATA	Shared Library Data
SLIBTEXT	Shared Library Text
SMMAP	Shared Memory Map
SMMFILE	Shared Memory Map file
SMMPSXRT	Shared POSIX Real Time Shared Memory
STACK	Stack Region
VSID	Virtual Segment ID of the memory region.
MAPPED OBJECT	Mapped object name.

Examples

1. To display the address space of process 12644, enter:

```
procmap 12644
```

The output of this command might look like this:

```
12644 : -ksh
10000000      232K read/exec      ksh
20000ef8      54K read/write     ksh
d008b100      80K read/exec      /usr/lib/libiconv.a[shr4.0]
f03e4c70      41K read/write     /usr/lib/libiconv.a[shr4.0]
d0080100      40K read/exec      /usr/lib/libi18n.a[shr.o]
f03f0b78       4K read/write     /usr/lib/libi18n.a[shr.o]
d007a000      11K read/exec      /usr/lib/nls/loc/en_US
d007d130       8K read/write     /usr/lib/nls/loc/en_US
d00790f8       2K read/exec      /usr/lib/libcrypt.a[shr.o]
f03e3508       0K read/write     /usr/lib/libcrypt.a[shr.o]
d02156c0     2282K read/exec      /usr/lib/libc.a[shr.o]
f03474e0      621K read/write     /usr/lib/libc.a[shr.o]
  Total      3380K
```

2. To display the address ranges of process with PID 1573580, enter:

```
# procmap -X 1573580
```

```
1573580 : ./self_mod_32
```

Start-ADD	End-ADD	SIZE	MODE	PSIZ	TYPE	VSID	MAPPED OBJECT
0	10000000	262144K	r--	m	KERTXT	20002	
10000000	10001d33	7K	rwX	sm	MAINTEXT	1730DF3	self_mod_32
200001d0	200007fc	1K	rw-	sm	MAINDATA	4F0CCF	self_mod_32
200007fc	20011000	66K	rw-	sm	HEAP	4F0CCF	
20011000	20011685	1K	rwX	sm	PLIBTEXT	4F0CCF	./libself_priv.a[dl_priv32.o]
20011685	2001234c	3K	rw-	sm	HEAP	4F0CCF	
2001234c	200125c4	0K	rw-	sm	PLIBDATA	4F0CCF	./libself_priv.a[dl_priv32.o]
200125c4	20013000	2K	rw-	sm	HEAP	4F0CCF	
20013000	2ff23000	261184K	rw-	sm	STACK	4F0CCF	
30000000	30001000	4K	rw-	sm	SMMFILE	8C0C0C	dev:10,7 ino:35
30001000	30002000	4K	r--	sm	SMMFILE	8C0C0C	dev:10,7 ino:36
30002000	30003000	4K	rw-	sm	PMMFILE	8C0C0C	dev:10,7 ino:35
30003000	30004000	4K	r--	sm	PMMFILE	8C0C0C	dev:10,7 ino:36
30004000	30005000	4K	rw-	sm	SMMPSXRT	8C0C0C	POSIX RT SHM 1
30005000	30006000	4K	r--	sm	SMMPSXRT	8C0C0C	POSIX RT SHM 1
30006000	30007000	4K	rw-	sm	PMMPSXRT	8C0C0C	POSIX RT SHM 2
30007000	30008000	4K	r--	sm	PMMPSXRT	8C0C0C	POSIX RT SHM 2
30008000	30009000	4K	rw-	sm	SMMAP	8C0C0C	
30009000	3000a000	4K	r--	sm	SMMAP	8C0C0C	
3000a000	3000b000	4K	rw-	sm	PMMAP	8C0C0C	
3000b000	3000c000	4K	r--	sm	PMMAP	8C0C0C	
40000000	40000000	0K	rw-	s	SHMFILE	250CA5	dev:10,7 ino:35
50000000	50000000	0K	r--	s	SHMFILE	1960D16	dev:10,7 ino:36
60000000	60001000	4K	rw-	sm	SHM	D0C8D	shmid:16
70000000	70001000	4K	r--	sm	SHM	11F0D9F	shmid:17
d0100100	d052343c	4236K	rwX	m	SLIBTEXT	3010B81	/usr/lib/libc.a[shr.o]
d0564100	d0564abe	2K	rwX	m	SLIBTEXT	3010B81	/usr/lib/libcrypt.a[shr.o]
d0652100	d0653654	5K	rwX	m	SLIBTEXT	3010B81	./libself.a[support32.o]
d0654380	d0654a02	1K	rwX	m	SLIBTEXT	3010B81	./libself.a[shr32.o]
d0655a80	d0656105	1K	rwX	m	SLIBTEXT	3010B81	./libself.a[dl_shr32.o]
f05935cc	f0593844	0K	rw-	sm	PLIBDATA	17F0DFF	./libself.a[shr32.o]
f06a5b6f	f06a60c0	1K	rw-	sm	PLIBDATA	17F0DFF	./libself.a[support32.o]
f07b4ccc	f07b4f44	0K	rw-	sm	PLIBDATA	17F0DFF	./libself.a[dl_shr32.o]
f07dfbb0	f08b7388	861K	rw-	sm	PLIBDATA	17F0DFF	/usr/lib/libc.a[shr.o]
f08b86a8	f08b87c8	0K	rw-	sm	PLIBDATA	17F0DFF	/usr/lib/libcrypt.a[shr.o]
Total		528579K					

3. To display the address ranges with file name association for the mapped file of process with PID 2031848, enter:

```
# procmap -Xf 2031848
```

```
2031848 : ./self_mod_64
```

Start-ADD	End-ADD	SIZE	MODE	PSIZ	TYPE	VSID	MAPPED OBJECT
0	10000000	262144K	r--	m	KERTXT	20002	
10000000	100002058	8K	rwX	sm	MAINTEXT	3C0CBC	self_mod_64
1100002d0	1100009e0	1K	rw-	sm	MAINDATA	240CA4	self_mod_64
1100009e0	110010a00	64K	rw-	sm	HEAP	240CA4	
8000000000000e80	8000000000012eb	1K	rwX	sm	PLIBTEXT	5A0CDA	./libself_priv.a[dl_priv64.o]
8001000a0000180	8001000a00001c0	0K	rw-	sm	PLIBDATA	16A0DEA	./libself_priv.a[dl_priv64.o]
800200140000000	80020014003d000	244K	r--	sm	KER/LDR	2A20E22	
8ffffff00000000	900000000000000	262144K	r--	s	KER/LDR	530CD3	
900000000000e00	900000000440541	4349K	rwX	m	SLIBTEXT	28C0E0C	/usr/lib/libc.a[shr_64.o]

900000000466400	900000000466f43	2K	rxw	m	SLIBTEXT	28C0E0C	/usr/lib/libcrypt.a[shr_64.o]
900000000467980	900000000468e98	5K	rxw	m	SLIBTEXT	28C0E0C	./libself.a[support64.o]
900000000469100	900000000469568	1K	rxw	m	SLIBTEXT	28C0E0C	./libself.a[shr64.o]
90000000046a800	90000000046ac6b	1K	rxw	m	SLIBTEXT	28C0E0C	./libself.a[dl_shr64.o]
9001000a0000580	9001000a010cb88	1073K	rw-	sm	PLIBDATA	12A0DAA	/usr/lib/libc.a[shr_64.o]
9001000a010da28	9001000a010dbb8	0K	rw-	sm	PLIBDATA	12A0DAA	/usr/lib/libcrypt.a[shr_64.o]
9001000a0137380	9001000a01378b8	1K	rw-	sm	PLIBDATA	12A0DAA	./libself.a[support64.o]
9001000a0248400	9001000a0248440	0K	rw-	sm	PLIBDATA	12A0DAA	./libself.a[shr64.o]
9001000a045ab00	9001000a045ab40	0K	rw-	sm	PLIBDATA	12A0DAA	./libself.a[dl_shr64.o]
900200140000000	900200150000000	262144K	r--	s	KER/LDR	B70037	
9ffffffd0000000	9ffffffe0000000	262144K	r--	sm	KER/LDR	50005	
9ffffffe0000000	9ffffff00000000	262144K	r--	sm	KER/LDR	E000E	
9ffffff00000000	9ffffff000fa8e	62K	rxw	s	SLIBTEXT	1180198	/usr/ccs/bin/usla64
9ffffff000fa8e	9ffffff000fa8e	0K	rw-	s	PLIBDATA	1180198	/usr/ccs/bin/usla64
a00000000000000	a00000000001000	4K	rw-	sm	SMMFILE	420CC2	/tmp/mmfile1
a00000000001000	a00000000002000	4K	r--	sm	SMMFILE	420CC2	/tmp/mmfile2
a00000000002000	a00000000003000	4K	rw-	sm	PMMFILE	420CC2	/tmp/mmfile1
a00000000003000	a00000000004000	4K	r--	sm	PMMFILE	420CC2	/tmp/mmfile2
a00000000004000	a00000000005000	4K	rw-	sm	SMMPSXRT	420CC2	POSIX RT SHM 1
a00000000005000	a00000000006000	4K	r--	sm	SMMPSXRT	420CC2	POSIX RT SHM 1
a00000000006000	a00000000007000	4K	rw-	sm	PMMPSXRT	420CC2	POSIX RT SHM 2
a00000000007000	a00000000008000	4K	r--	sm	PMMPSXRT	420CC2	POSIX RT SHM 2
a00000000008000	a00000000009000	4K	rw-	sm	SMMAP	420CC2	
a00000000009000	a0000000000a000	4K	r--	sm	SMMAP	420CC2	
a0000000000a000	a0000000000b000	4K	rw-	sm	PMMAP	420CC2	
a0000000000b000	a0000000000c000	4K	r--	sm	PMMAP	420CC2	
a00010000000000	a00010000000000	0K	rw-	s	SHMFILE	250CA5	/tmp/mmfile1
a00010010000000	a00010010000000	0K	r--	s	SHMFILE	1960D16	/tmp/mmfile2
a00020000000000	a00020000001000	4K	rw-	sm	SHM	5B0CDB	shmid:18
a00030000000000	a00030000001000	4K	r--	sm	SHM	1980D18	shmid:19
ffffff000000000	100000000000000	4194304K	rw-	sm	STACK	10D0D8D	
Total		5510897K					

Files

Item	Description
/proc	Contains the /proc filesystem.

procrun Command

Purpose

Starts a process that has stopped on the **PR_REQUESTED** event.

Syntax

procrun *ProcessID* ...

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/ProcessID** strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like **procrun** and **procstop** start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procrun** command starts the process that has stopped on the **PR_REQUESTED** event.

Flags

Item	Description
<i>ProcessID</i>	Specifies the process id.

Examples

1. To restart process 30192 which was stopped on the **PR_REQUESTED** event, enter:

```
procrun 30192
```

Files

Item	Description
/proc	Contains the /proc filesystem.

procsig Command

Purpose

Lists the signal actions defined by processes.

Syntax

procsig *ProcessID* ...

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ASCII reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/ProcessID** strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like **procrun** and **procstop** start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procsig** command lists the signal actions defined by processes.

Flags

Item	Description
<i>ProcessID</i>	Specifies the process id.

Examples

1. To list all the signal actions defined for process 11928, enter:

```
procsig 11928
```

The output of this command might look like this:

```
HUP      caught
INT      caught
QUIT    caught
ILL      caught
TRAP    caught
ABRT    caught
EMT     caught
FPE     caught
KILL    default  RESTART
BUS     caught
SEGV    default
SYS     caught
PIPE    caught
ALRM    caught
TERM    ignored
URG     default
STOP    default
TSTP    ignored
CONT    default
CHLD    default
TTIN    ignored
TTOU    ignored
IO      default
XCPU    default
XFSZ    ignored
MSG     default
WINCH   default
PWR     default
USR1    caught
USR2    caught
PROF    default
DANGER  default
VTALRM  default
MIGRATE default
PRE     default
VIRT    default
ALRM1   default
WAITING default
CPUFAIL default
KAP     default
RETRACT default
SOUND   default
SAK     default
```

Files

Item	Description
/proc	Contains the /proc filesystem.

procstack Command

Purpose

Prints the hexadecimal addresses and symbolic names for all the threads in the process.

Syntax

```
procstack [ -F ] [ -g ] ProcessID ...
```

Description

The `/proc` filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The **proctools** commands provide ASCII reports based on some of the available information.

Most of the commands take a list of process IDs or `/proc/ProcessID` strings as input. The shell expansion `/proc/*` can therefore be used to specify all processes in the system.

Each of the **proctools** commands gathers information from `/proc` for the specified processes and displays it to the user. The **proctools** commands like **procrun** and **procstop** start and stop a process using the `/proc` interface.

The information gathered by the commands from `/proc` is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procstack** command prints the hexadecimal addresses and symbolic names for all the threads in the process.

Flags

Item	Description
-F	Forces the procstack command to take control of the target process even if another process has control.
-g	Prevents the conversion of symbol names to human-readable names.
<i>ProcessID</i>	Specifies the process ID.

Examples

1. To display the current stack of process 11928, enter:

```
procstack 11928
```

The output of this command might look like this:

```
11928 : -sh
d01d15c4 waitpid  (? , ? , ? ) + e0
10007a1c job_wait  (?) + 144
10020298 xec_switch (? , ? , ? , ? , ? ) + 9c0
10021db4 sh_exec   (? , ? , ? ) + 304
10001370 exfile   () + 628
10000300 main    (? , ? ) + a1c
10000100 __start  () + 8c
```

2. To display the current stack of all the threads of the multi-threaded process 28243 for application *appl*, enter:

```
procstack 28243
```

The output of this command would look like this:

```
28243 : appl
----- tid# 54321 -----
d0059eb4 _p_nsleep  (? , ? ) + 10
d01f1fc8 nsleep   (? , ? ) + b4
d026a6c0 sleep   (?) + 34
100003a8 main    () + 98
10000128 __start  () + 8c
```

```

----- tid# 43523 -----
d0059eb4 _p_nsleep  (? , ?) + 10
d01f1fc8 nsleep  (? , ?) + b4
d026a6c0 sleep  (?) + 34
10000480 PrintHello  (d) + 30
d004b314 _pthread_body  (?) + ec
----- tid# 36352 -----
d0059eb4 _p_nsleep  (? , ?) + 10
d01f1fc8 nsleep  (? , ?) + b4
d026a6c0 sleep  (?) + 34
10000480 PrintHello  (c) + 30
d004b314 _pthread_body  (?) + ec

```

Files

Item	Description
<code>/proc</code>	Contains the <code>/proc</code> filesystem.

procstop Command

Purpose

Stops processes on the **PR_REQUESTED** event.

Syntax

procstop *ProcessID* ...

Description

The `/proc` filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The **proctools** commands provide ASCII reports based on some of the available information.

Most of the commands take a list of process IDs or `/proc/ProcessID` strings as input. The shell expansion `/proc/*` can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from `/proc` for the specified processes and displays it to the user. The **proctools** commands like **procrun** and **procstop** start and stop a process using the `/proc` interface.

The information gathered by the commands from `/proc` is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procstop** command stops processes on the **PR_REQUESTED** event.

Flags

Item	Description
<i>ProcessID</i>	Specifies the process id.

Examples

- To stop process 7500 on the **PR_REQUESTED** event, enter:

```
procstop 7500
```

Files

Item	Description
<code>/proc</code>	Contains the <code>/proc</code> filesystem.

proctree Command

Purpose

Prints the process tree containing the specified process IDs or users.

Syntax

```
proctree [ -a ] [ { ProcessID | User } ]
```

```
proctree [ -a ] [ -T ] [ -t ] [ { -p ProcessID | -u User } ] [ -@ [ WparName ] ]
```

Description

The `/proc` filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The **proctools** commands provide ASCII reports based on some of the available information.

Most of the commands take a list of process IDs or `/proc/ProcessID` strings as input. The shell expansion `/proc/*` can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from `/proc` for the specified processes and displays it to the user. The **proctools** commands like **procrun** and **procstop** start and stop a process using the `/proc` interface.

The information gathered by the commands from `/proc` is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **proctree** command prints the process tree containing the specified process IDs or users. The child processes are indented from their respective parent processes. An argument of all digits is taken to be a process ID, otherwise it is assumed to be a user login name. The default action is to report on all processes, except children of process 0.

When you specify the `-@` flag with no parameters, all WPAR names are displayed. If you specify the `WparName` parameter, only those WPAR names are displayed.

For displaying thread IDs and associated pthread IDs, specify the `-t` option. For a kernel process, the **proctree** command displays only the thread ID.

Note: If the information about the process cannot be retrieved, the **proctree** command displays only the process ID. The other information about the process is shown as blank. For example, the **proctree** command shows only the process ID for the zombie process.

Flags

Item	Description
<code>-a</code>	Includes children of process 0 in the display. The default is to exclude them.
<code>ProcessID</code>	Specifies the process ID.
<code>-p ProcessID</code>	Specifies the process ID.
<code>-T</code>	Displays the formatted output of the process tree.
<code>-t</code>	Displays thread IDs and associated pthread IDs for the process.

Item	Description
<i>User</i>	Specifies the user name.
<i>-u User</i>	Specifies the user name.
<i>-@</i>	Displays all WPAR names. Note: The <i>-@</i> flag is not supported when executed within a workload partition.
<i>-@ WparName</i>	Displays only the processes of the WPAR you specify using the <i>WparName</i> parameter. Note: The <i>-@</i> flag is not supported when executed within a workload partition.

Examples

1. To display the ancestors and all the children of the 12312 process, enter the following command:

```
proctree 12312
```

The output of this command might look like this:

```
4954 /usr/sbin/srcmstr
  7224 /usr/sbin/inetd
    5958 telnetd -a
      13212 -sh
        14718 ./proctree 12312
```

2. To display the ancestors and children of the 12312 process, including children of process 0, enter the following command:

```
proctree -a 12312
```

The output of this command might look like this:

```
1 /etc/init
  4954 /usr/sbin/srcmstr
    7224 /usr/sbin/inetd
      5958 telnetd -a
        13212 -sh
          14724 ./proctree -a 12312
```

3. To display the process tree of WPAR corral2, enter the following command:

```
proctree -@ corral2
```

The output of this command might look like this:

```
corral2 401496 /etc/init
corral2 319680 /usr/sbin/srcmstr
corral2 102636 /usr/sbin/inetd
corral2 249954 /opt/rsct/bin/rmcd -a IBM.LPCCommands -r
corral2 254132 /opt/rsct/bin/IBM.AuditRMd
corral2 295098 /opt/rsct/bin/IBM.ServiceRMd
corral2 303218 /usr/dt/bin/dtlogin
corral2 307370 /usr/sbin/writesrv
corral2 323836 /usr/sbin/qdaemon
corral2 331970 /usr/sbin/muxatmd
corral2 348210 /usr/sbin/syslogd
corral2 352472 sendmail: accepting connections H nnections
corral2 364564 /opt/rsct/bin/IBM.ERrmd
corral2 405522 /usr/sbin/portmap
corral2 282800 /usr/bin/xmwlrm -L
corral2 311454 /usr/sbin/cron
corral2 376920 /usr/lib/errdemon
```

4. To display the WPAR name of the processes, enter the following command:

```
proctree -@
```

The output of this command might look like this:

```
Global      114788  /usr/dt/bin/dtlogin -daemon
Global      86108  dtlogin <:0>        -daemon
Global      123022  dtgreet 8 :0
Global      77944  /usr/lib/errdemon
Global      94314  /usr/sbin/syncd 60
Global      168084  /usr/sbin/srcmstr
Global      110688  /opt/rsct/bin/IBM.ServiceRMd
corral2     401496  /etc/init
corral2     319680  /usr/sbin/srcmstr
corral2     102636  /usr/sbin/inetd
corral2     249954  /opt/rsct/bin/rmcd -a IBM.LPCCommands -r
corral2     254132  /opt/rsct/bin/IBM.AuditRMd
corral2     331970  /usr/sbin/muxatmd
corral2     348210  /usr/sbin/syslogd
corral2     364564  /opt/rsct/bin/IBM.ERrmd
corral2     405522  /usr/sbin/portmap
corral2     282800  /usr/bin/xmwlms -L
corral2     311454  /usr/sbin/cron
corral2     376920  /usr/lib/errdemon
Global      151626  /usr/ccs/bin/shlap64
Global      274578  /usr/sbin/getty /dev/console
...
```

5. To display the ancestors, all of the children, and the WPAR name of the 102636 process, enter the following command:

```
proctree -p 102636 -@
```

The output of this command might look like this:

```
Global      168084  /usr/sbin/srcmstr
corral2     401496  /etc/init
corral2     319680  /usr/sbin/srcmstr
corral2     102636  /usr/sbin/inetd
```

6. To display the formatted process-tree output of the 213246 process, enter the following command:

```
proctree -T -p 213246
```

The output of this command might look like this:

```
192652      \--/usr/sbin/srcmstr
200830      \--/usr/sbin/inetd
213246      \--telnetd -a
229592      \---ksh
```

7. To display thread IDs and associated pthread IDs for the 344172 process, enter the following command:

```
proctree -t -p 344172
```

The output of this command might look like this:

```
192652      /usr/sbin/srcmstr
  TID : 225535 (pTID : 1)
 200830      /usr/sbin/inetd
  TID : 360677 (pTID : 1)
 323642      telnetd -a
  TID : 770057 (pTID : 1)
 307428      -ksh
  TID : 1056861 (pTID : 1)
 344172      apthd
  TID : 1065119 (pTID : 1)
  TID : 1028171 (pTID : 258)
  TID : 1011789 (pTID : 2057)
  TID : 1024105 (pTID : 1800)
```

8. To display the formatted process-tree output for the 344172 process along with thread IDs and associated pthread IDs, enter the following command:

```
proctree -tT -p 344172
```

The output of this command might look like this:

```
192652  \--/usr/sbin/srcmstr
        ~~TID : 225535 (pTID :    1)
200830  \--/usr/sbin/inetd
        ~~TID : 360677 (pTID :    1)
323642  \--telnetd -a
        ~~TID : 770057 (pTID :    1)
307428  \---ksh
        ~~TID : 1056861 (pTID :    1)
344172  \--apthd
        |~~TID : 1065119 (pTID :    1)
        |~~TID : 1028171 (pTID :   258)
        |~~TID : 1011789 (pTID :  2057)
        |~~TID : 1024105 (pTID :  1800)
```

Files

Item	Description
<code>/proc</code>	Contains the <code>/proc</code> filesystem.

procpwait Command

Purpose

Waits for all of the specified processes to terminate.

Syntax

```
procpwait [ -v ] ProcessID ...
```

Description

The `/proc` filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The **proctools** commands provide ASCII reports based on some of the available information.

Most of the commands take a list of process IDs or `/proc/ProcessID` strings as input. The shell expansion `/proc/*` can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from `/proc` for the specified processes and displays it to the user. The **proctools** commands like **procrun** and **procstop** start and stop a process using the `/proc` interface.

The information gathered by the commands from `/proc` is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procpwait** command waits for all of the specified processes to terminate.

Flags

Item	Description
-v	Specifies verbose output. Reports terminations to standard output.
<i>ProcessID</i>	Specifies the process id.

Examples

1. To wait for process 12942 to exit and display the status, enter:

```
procwait -v 12942
```

The output of this command might look like this:

```
12942 : terminated, exit status 0
```

Files

Item	Description
<code>/proc</code>	Contains the <code>/proc</code> filesystem.

procwdx Command

Purpose

Prints the current working directory of processes.

Syntax

```
procwdx [ -F ] ProcessID ...
```

Description

The `/proc` filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The **proctools** commands provide ASCII reports based on some of the available information.

Most of the commands take a list of process IDs or `/proc/ProcessID` strings as input. The shell expansion `/proc/*` can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from `/proc` for the specified processes and displays it to the user. The **proctools** commands like **procrun** and **procstop** start and stop a process using the `/proc` interface.

The information gathered by the commands from `/proc` is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procwdx** command prints the current working directory of processes.

Flags

Item	Description
-F	Forces procfiles to take control of the target process even if another process has control.
<i>ProcessID</i>	Specifies the process id.

Examples

1. To display the current working directory of process 11928, enter:

```
procwdx 11928
```

The output of this command might look like this:

Files

Item	Description
<code>/proc</code>	Contains the <code>/proc</code> filesystem.

prof Command

Purpose

Displays object file profile data.

Syntax

```
prof [ -t | -c | -a | -n ] [ -o | -x ] [ -g ] [ -z ] [ -h ] [ -s ] [ -S ] [ -v ] [ -L PathName ] [ Program ]
[ -m MonitorData ... ]
```

Description

The **prof** command interprets profile data collected by the **monitor** subroutine for the object file *Program* (**a.out** by default). It reads the symbol table in the object file *Program* and correlates it with the profile file (**mon.out** by default). The **prof** command displays, for each external text symbol, the percentage of execution time spent between the address of that symbol and the address of the next, the number of times that function was called, and the average number of milliseconds per call.

Note: Symbols from C++ object files have their names demangled before they are used.

To tally the number of calls to a function, you must have compiled the file using the **cc** command with the **-p** flag. The **-p** flag causes the compiler to insert a call to the **mcoun**t subroutine into the object code generated for each recompiled function of your program. While the program runs, each time a parent calls a child function the child calls the **mcoun**t subroutine to increment a distinct counter for that parent-child pair. Programs not recompiled with the **-p** flag do not have the **mcoun**t subroutine inserted and therefore keep no record of which function called them.

The **-p** flag also arranges for the object file to include a special profiling startup function that calls the **monitor** subroutine when the program begins and ends. The call to the **monitor** subroutine when the program ends actually writes the **mon.out** file. Therefore, only programs that explicitly exit or return from the main program cause the **mon.out** file to be produced.

Note: To change the name of the generated output file, use the **PROF** environment variable and set it as follows:

```
PROF=filename:<filename>
```

For example, if you set **PROF=myprof**, then the generated file will be named as **myprof.out**.

The location and names of the objects loaded are stored in the **mon.out** file. If you do not select any flags, **prof** will use these names. You must specify a program or use the **-L** option to access other objects.

Note: Imported external routine calls, such as a call to a shared library routine, have an intermediate call to local **glink** code that sets up the call to the actual routine. If the timer clock goes off while running this code, time is charged to a routine called *routine.gl*, where *routine* is the routine being called. For example, if the timer goes off while in the **glink** code to call the **printf** subroutine, time is charged to the **printf.gl** routine.

Flags

The mutually exclusive flags **a**, **c**, **n**, and **t** determine how the **prof** command sorts the output lines:

Item Description

- a** Sorts by increasing symbol address.
- c** Sorts by decreasing number of calls.
- n** Sorts lexically by symbol name.
- t** Sorts by decreasing percentage of total time (default).

Note: The **prof** command can still run successfully if you use more than one of flags **a**, **c**, **n**, and **t** in the same command. The **prof** command accepts the first of these flags it encounters on the command line and ignores the others.

The mutually exclusive flags **o** and **x** specify how to display the address of each symbol monitored.

Item Description

- o** Displays each address in octal, along with the symbol name.
- x** Displays each address in hexadecimal, along with the symbol name.

Note: The **prof** command can still run successfully if you use both the **-o** and **-x** flags in the same command. The **prof** command accepts the first of these two flags it encounters on the command line and ignores the other flag.

Use the following flags in any combination:

Item	Description
-g	Includes non-global symbols (static functions).
-h	Suppresses the heading normally displayed on the report. This is useful if the report is to be processed further.
-L <i>PathName</i>	Uses alternate path name for locating shared objects.
-m <i>MonitorData</i>	Takes profiling data from <i>MonitorData</i> instead of mon.out .
-s	Produces a summary file in mon.sum . This is useful when more than one profile file is specified.
-S	Displays a summary of monitoring parameters and statistics on standard error.
-v	Suppresses all printing and sends a graphic version of the profile to standard output for display by the plot filters. When plotting, low and high numbers, by default 0 and 100, can be given to cause a selected percentage of the profile to be plotted with accordingly higher resolution.
-z	Includes all symbols in the profile range, even if associated with 0 (zero) calls and 0 (zero) time.

Examples

1. To display, without a header, the amount of time spent at each symbol address, sorted by time, enter:

```
prof -t -h
```

2. The following example obtains a local version of any shared libraries used to create the **runfile** file in the **/home/score/lib** directory. The data file used will be **runfile.mon** rather than **mon.out**.

```
prof -x -L/home/score/lib runfile -m runfile.mon
```

Files

Item	Description
mon.out	Default profile.
a.out	Default object file.
mon.sum	Summary profile.

proff Command

Purpose

Formats text for printers with personal printer data streams.

Syntax

```
proff [ -LList ] [ -PPrinter ] [ -t ] [ nroffFlags ] [ File ... ]
```

Description

The **proff** command formats text by using the **nroff** command on the specified files for printers that support ppds (personal printer data streams), such as the Quietwriter III printer, the Quickwriter printer, and the Proprinter printer.

If no file is specified, standard input is read. A parameter value of - (minus) specifies standard input.

Parameters

Item	Description
<i>nroffFlags</i>	Specifies the nroff command flags used by the proff command to format the text file for a ppds-supported printer output.
<i>File</i>	Specifies the text file that the proff command formats for printers that support ppds.

Flags

Item	Description
-L List	Passes the specified list as flags for the qprt command. To pass a single flag to the qprt command, use the -L flag followed immediately by the nroff command flag being passed. For example: <pre>-L-h.</pre> To pass multiple flags or a string to the lpr command, use the -L flag followed immediately by the flags or string enclosed by " " (double quotes): <pre>-L"-h -x -m".</pre>
-P Printer	Sends output to a specified printer corresponding to an entry in the /etc/qconfig file. The default is taken from the PRINTER environment variable, if it exists; otherwise the system default queue name is used.
-t	Sends output to standard output.

Item	Description
-	Specifies that standard input is used as the source for the formatting process. All other flags are passed to the nroff command.

Example

The following is a typical command sequence to process output for the IBM Proprinter printer:

```
proff -t testfile
```

Environment Variable

Item	Description
PRINTER	Specifies the desired printer queue.

Files

Item	Description
/usr/share/lib/nterm/tab.ppds	Contains driving tables for printers with personal printer data streams.
/etc/qconfig	Describes the queues and devices.

projctl Command

Purpose

Supports project-based advanced accounting activities.

Syntax

projctl add *projname projnumber* [*comment*] [{ **-d** *projpath* | **-p** [*DN*] }]

projctl merge *sourceprojpath* [**-d** *targetprojfile*]

projctl rm *projname* [{ **-d** *projpath* | **-p** [*DN*] }]

projctl chg *projname* [**-p** *pid* [, *pid*]] [**-f**]

projctl exec *projname* *<cmd line>* [**-f**]

projctl chattr agg *projname* { **-s** | **-u** } [{ **-d** *projpath* | **-p** [*DN*] }]

projctl qpolicy [**-g** [*DN*]]

projctl qprojs [**-n**]

projctl qproj [*projectname*]

projctl qapp *appname*

projctl { **chkusr** | **chkgrp** | **chkprojs** | { **chkadm** | **chkall** } [**-d** *admpath*] }

projctl ldusr [**-r**] [**-a**]

projctl unldusr [**-a**]

projctl ldgrp [**-r**] [**-a**]

projctl unldgrp [**-a**]


```

projectl ldprojs -g [ -r ] [ -a ]
projectl ldprojs -g [DN] -d projpath
projectl ldprojs -p [DN] -d projpath
projectl unldprojs -g [DN] [ -f ] [ -a ]
projectl unldprojs -p [DN]
projectl ldadm -g [name] [ -r ] [ -a ]
projectl ldadm -g [name:]DN | name ] -d admpath
projectl ldadm -p [ [name:]DN | name ] -d admpath
projectl unldadm -g [ -a ]
projectl unldadm -p [ [name:]DN | name ]
projectl ld [ -r ]
projectl ldall [ -d admpath ] [ -r ] [ -a ]
projectl unldall [ -f ] [ -a ]

```

Description

The various subcommands of **projectl** command perform project-based advanced accounting activities such as adding a new project, removing a new project, and loading a specific accounting policy. These various options of **projectl** command are as explained below.

Flags

Item	Description
-a	Automatically loads the policies during system reboot.
-d	Generally specifies the path from where the project definition file or the admin policy file should be referred. When used in the merge subcommand, it specifies the target project definition file where the merged project definitions are to be stored.
-f	Overrides the policy rules when specified with chg and exec subcommands. Clears the project assigned to the processes when called with unldall subcommand. Force unload all the project definitions when called with unldprojs subcommand.
-g	Specifies that the projects and policies are to be downloaded from the LDAP repository.
-n	Sorts the list of project definitions based on the name.
-p	When used in the chg subcommand, passes the list of process IDs that require a change in project assignment. When used in the add , rm , and chattr subcommands, specifies the LDAP DN where the project definition is to be updated. When used in the ld and unld subcommands, specifies that the projects and policies are to be uploaded to the LDAP repository. Its argument indicates the DN where the projects and policies are to be uploaded.
-r	Reloads the policies.
-s	Used in projectl chattr agg subcommand to enable the project aggregation property.
-u	Used in projectl chattr agg subcommand to disable the project aggregation property.

Parameters

Item	Description
<i>admpath</i>	Path from where to select the admin policy file.

Item	Description
<i>appname</i>	Absolute path of the application whose project assignment list is requested.
<i>cmd line</i>	Absolute path of the command to be executed through projectl exec command.
<i>comment</i>	Project comments.
<i>DN</i>	Distinguished Name that indicates the absolute path to the project and policy objects on the LDAP server.
<i>name</i>	Name of the alternate admin policy definitions on the LDAP server.
<i>pid</i>	Process IDs.
<i>projname</i>	Name of the project.
<i>projnumber</i>	Numeric value for the project.
<i>projpath</i>	Path from where to select the project definition file.
<i>sourceprojpath</i>	Path from where the project definition file to be merged is to be picked up.
<i>targetprojfile</i>	Target project definition file where the project definitions should be merged.

Subcommands

add Subcommand

The **add** subcommand adds the definition of the project to the project definition file. If the **-d** flag is specified then the project definition is added into the project definition file, under the named path. The default is to add to the **/etc/project/projdef** system project definition file. The project definition file under any other path should be named as **.projdef**. If the new project is to be added to the system project definition file and the projects are already loaded in kernel, then the specified new project will be added into kernel project registry. Otherwise, the entry will be made only in the file. The **add** subcommand takes the project name, project number, and an option argument for project comments as parameters. By default, the aggregation property of the project will be set to no for all the projects created using this command.

If **-p** is specified, the new project definition is added to default project *DN* or the specified *DN* on the LDAP server. If **-p** is not specified, **.config** will provide source information. Running the **-p** option requires root authority.

Each entry created by **projectl add** in the Project Definition File has the following format:

```
ProjectName:ProjectNumber:AggregationStatus::Comment
```

Examples for Project Definitions that illustrate the file format are as follows:

```
:: Project Definition File
:: Dated: 23-JUN-2003
AIX:3542:yes::To Classify AIX Legacy Applications
Test_Project:0x10000:yes::To Classify Testing work
```

chattr agg Subcommand

The **chattr agg** subcommand enables and disables aggregation property for the given project. If **-s** flag is used the aggregation is enabled. If **-u** flag is used aggregation is disabled. If **-d** flag is specified then the project definition is updated in the project definition file under the specified path. The default is to update the system project definition file (**/etc/project/projdef**). If the update is to the system project definition file and it is already loaded in kernel, then the specified new project is updated in kernel project registry as well. Otherwise, the changes will be made only to the project definition file.

If **-p** is specified, the project definition is modified on default project *DN* or the specified *DN* on the LDAP server. If **-p** is not specified, **.config** will provide source information. Executing the **-p** option requires root authority.

chg Subcommand

The **chg** subcommand enables the user to change the list of projects that the user is permitted to use for his processes. The intended project name is given as input to this subcommand. If the process IDs are provided as input, those processes will be classified under the specified project. If there are no process IDs provided as input, the project change will happen to the process which started the **projectl** command.

By default, the **chg** subcommand changes the project assignment within the scope of available rules. To override the rules and assign a project directly to a process, the **-f** force option must be specified.

chk Subcommand

The **chk** subcommands check the validity of various project policies. The subcommands validate the projects and policies so that they can be loaded safely into the kernel. There are several **chk** subcommands to support various project policies. The subcommands include:

Item	Description
chkadm	Validates the admin policies. Each rule in the admin policy file usually has four attributes: user-id, group-id, application path name, and the project names. The chkadm subcommand checks whether these attributes are valid and reports any errors found in the policies. When the -d option is used, the chkadm subcommand uses the admin policy file from the specified path for checking the rules. It also uses the alias and the temporary project definition file (.projdef), if required. The projects used in the rule will be first searched in the system project definition file. If it is not found there, then the .projdef file under the specified path will be used.
chkall	Performs all the above validation activities, that is, it validates projects, user, group and admin policies. When the -d option is used, the chkadll subroutine uses the admin, alias, and project definition files from the specified path to validate the admin policies.
chgrp	Validates the group policies. The validation involves checking whether the project list of the group contains valid projects.
chkprojs	Validates the system project definition file. Project Definitions are validated for uniqueness, project name and number validity, and attributes validity. The project name should be a POSIX alphanumeric string and the project number should be within the numeric range 0x00000001 - 0x00ffffff. The project numbers can be either decimal or hexadecimal numbers. All hexadecimal numbers should be shown with a prefix of 0x. The aggregation property can be either a y or a n to indicate the status of aggregation. The chkprojs subcommand performs all these validity checks on the project definitions and reports any errors found with the project definitions.
chkusr	Validates the user policies. The validation involves checking whether the project list of the user contains valid projects.

Note: If wildcard characters are used in the admin policy rules then **chkadm** and **chkall** subcommands expand the wildcard characters and validate the output obtained.

exec Subcommand

The **exec** subcommand allows a user to launch arbitrary commands with any of the project names from the list of projects on which the command can work. Similar to **chg** option, used to override the rules and use any project to run the command line, the **-f** force option should be used. To get the list of projects that the command can be assigned to, use the **projectl qapp** subcommand.

ld Subcommand

The **ld** subcommands are used to load and reload projects and policies. There are specific load commands to perform the load operation on a specific policy. These various subcommands are as follows:

Item	Description
ld	Loads the policies, which should be loaded during the system startup. It refers the /etc/project/.config file to determine which policies to load. If the kernel is loaded already with any one policy or project definition, then this command simply returns.
ldadm	<p>Loads the admin policies. Similar to ldusr and ldgrp subcommands, ldadm also checks and loads the projects first, if they are yet to be loaded. Then it loads the admin policy rules, after validating them. When the -d option is used, the admin policy file will be picked from the specified path. The alias and the temporary project definition file under the specified path will be used to check the existence of alias and project entries. After the policies are loaded, this subcommand also copies the admin policy file to /etc/project/.admin. Loading the admin policies related to LDAP is handled by the following -p and -g arguments:</p> <p>projectl ldadm -g [name] Specifies that an admin policy will be loaded into the kernel using the LDAP repository. If -g is not specified, the local admin policy (/etc/project/admin) will be downloaded into the kernel.</p> <p>projectl ldadm -g [[name:]DN name] -d admpath Specifies that an LDAP admin policy will be downloaded to a local file without downloading the policy into the kernel. The source admin policy is located at the specified DN or is found using the accounting DNs in the ldap.cfg file. The -d parameter is used to specify where the policy files (projects, admin, and alias) are written. If the target location is below /etc/project/, the files are written according to the conventions used by the system. Files are written to:</p> <ul style="list-style-type: none"> • /etc/project/admin, /etc/project/alias, /etc/project/projdef • /etc/project/ldap/admin, /etc/project/ldap/alias, /etc/project/ldap/projdef • /etc/project/projdef, /etc/project/alter/policyname/admin, .../alias • /etc/project/ldap/projdef, /etc/project/ldap/alter/policyname/admin, .../alias <p>Otherwise, the three files are written to the specified directory. When an explicit DN is specified with the -g option, the projects are not downloaded because they could also be located on a different DN. In this case, the user has to download them separately.</p> <p>projectl ldadm -p [[name:]DN name] -d admpath Specifies that an admin policy located at the directory <i>localpath</i> will be uploaded to the LDAP server. This command also uploads projects that it finds in the <i>localpath/.projdef</i> temporary project definition file. When an explicit DN is specified with the -p option, only the admin policy is uploaded to the LDAP server because the projects could be located on a different DN. In this case, the user must explicitly upload the respective <i>.projdef</i> file to the appropriate DN. The system does not know the identity of this DN. The -d argument must be specified when the -g or -p arguments are used. The -r and -a arguments cannot be specified with the -p argument. If the -a argument is specified and the -g argument is not specified, the admin policies in the <i>.config</i> file are loaded. If the -r option is used, the <i>.active</i> file is used to determine the identity of the policies to load. The -r and -a options cannot be used together.</p>
ldall	Downloads user, group, and admin policies into the kernel. Similar to the ldusr and ldgrp commands, this option attempts to download LDAP projects if an accounting DN has been specified for projects, because the User and Group Policies are not associated with Local or LDAP Users individually. This command attempts to download the default Admin policy using the configured admin DN in addition to downloading the Local Admin Policy.

Item	Description
ldgrp	Loads the group project policies. If they are not yet loaded, the ldgrp subcommand checks and loads the projects first. It then verifies the validity of the project list for all the groups and loads the rules.
ldprojs	<p>Loads the project definitions from the system project definition /etc/project/projdef file. Before loading the projects, it checks the validity of the rules. If the rules are found to valid, then it loads them.</p> <p>projectl ldprojs -g Specifies that the project definitions will be loaded into the kernel using the LDAP repository.</p> <p>projectl ldprojs -p Specifies that project definitions are to be uploaded to the LDAP server. If -g and the -p are not specified, the locally defined projects (/etc/project/projdef) are loaded into the kernel.</p> <p>projectl ldprojs -g [DN] -d localpdfpath Specifies that the project definition file from the LDAP repository will be downloaded to a local file without downloading the projects into the kernel. If the -d argument is not specified, the projects are downloaded to /etc/project/ldap/projdef and they are downloaded into the kernel. The -d argument directs you to create the file at the designated location, but not to download it into the kernel. In this case, the projdef file is created at the designated location rather than in the .projdef file. The source project definitions are located at the specified DN. Alternately, you can find them using the configured accounting DN in the ldap.cfg file.</p> <p>projectl ldprojs -d localpdfpath Loads the local project definition file into the kernel.</p> <p>projectl ldprojs -p [DN] -d localpdfpath Specifies that the project definitions located at the specified path will be uploaded to the LDAP server. The project definitions should be available in the projdef file at the specified directory. The -d argument must be specified when the -g or -p directs you to create the file at the designated location, but not to download it into the kernel. In this case, the projdef arguments are used. This way, the upload and download operations can be symmetric with respect to the specification of parameters. The -r and -a arguments cannot be specified with the -p argument. If the -a argument is specified and the -g argument is not specified, the project repositories in the .config file are loaded. If the -r option is used, the .active file is used to determine the project repositories to load. The -r and -a options cannot be used together.</p>
ldusr	Loads the user project policies. If they are not yet loaded, the lduser subcommand checks and loads the projects first. It then verifies the validity of the project list for all the users and loads the rules.

Note:

- When **-r** option is used, all the above subcommands reload the respective policies. The **ld -r** subcommand queries the kernel to get the details of loaded policies and reloads them. The policy files to be reloaded will be referred from the **/etc/project/.active** file.
- When **ldadm** and **ldall** subcommands are issued with both the options **-d** and **-r**, **-r** will be ignored.
- All the **ld** subcommands update the **/etc/project/.active** file with the details of the policy that is loaded. When the **-a** option is passed, these subcommands also update the **/etc/project/.config** file in addition to updating the **.active** file. The **/etc/project/.config** file provides the details on the policies to be loaded automatically on system reboot.

merge Subcommand

The **merge** subcommand merges the projects defined in the project definition file under the specified path with the system project definition **/etc/project/projdef** file, by default. If a target project file name

is passed using the **-d** option, the project definitions under the specified path are merged with the target project definition file. The merge operation will fail if there are conflicting entries between the target project definition file and the project definition file under the specified path. The **merge** command skips any duplicate entries to maintain unique entries in the target project definition file.

qapp Subcommand

The **qapp** subcommand displays the list of projects that an application can switch to in the current environment. It displays the list of all projects with which the specified application can be started.

qpolicy Subcommand

The **qpolicy** subcommand displays the currently loaded policies. This command queries the kernel to get the information about the types of loaded policies and displays them. If **-g** is specified, this command lists the policies from the LDAP default admin DN or from the specified DN.

qproj Subcommand

The **qproj** subcommand displays the details of the project name passed as its argument. If no argument is passed, then this subcommand lists all the project definitions in the system to which the calling process can be assigned. The display format will be the same as that of **qprojs** subcommand.

qprojs Subcommand

The **qprojs** subcommand displays the list of all the project definitions that is currently loaded in the kernel registry. The **-n** option provides the list sorted based on the project name. The display contains the project name, project number, and its aggregation status.

rm Subcommand

The **rm** subcommand removes the definition of locally defined projects from the project definition file. If the **-d** flag is specified, then the project definition is removed from the project definition file under the specified path. The default is to remove it from the system project definition file (**/etc/project/projdef**). If the update is to the system project definition file and it is already loaded in kernel, then the specified project is removed from kernel project registry. Otherwise, the entry will be removed only from the file.

If **-p** is specified, the source will be the LDAP from where the project definitions are to be removed. If an explicit DN is specified, the project definition will be removed from that specific DN. If no DN is passed, the default DN configured in the `ldap.cfg` file will be used. If the LDAP projects are currently loaded, the project definition is removed from the kernel project registry and the local LDAP project file also. Otherwise, only the LDAP repository is updated.

Note: The **-p** and **-d** options cannot be used together. If neither of these options are specified, the `.config` file will be used to provide the source information. This command requires root authority to execute.

unld Subcommand

The **unld** subcommands are used to unload project policies. Similar to the **ld** subcommands, the **unld** subcommands are used to unload specific policies. These various subcommands are as follows:

Item	Description
unldadm	Unloads the admin policies.
unldall	Unloads all the loaded policies.
unldgrp	Unloads the group policies.
unldprojs	Unloads only the project definitions.
unldusr	Unloads the user policies.

Note:

- All these subcommands update the **.active** file after the respective policy is unloaded.

- When the **-a** option is used, the **/etc/project/.config** file is also updated with the unloaded status of the respective policy.
- The **-g** parameter specifies that the respective LDAP repository should be unloaded from the kernel. If **-g** is not specified, then the loaded repositories that are named in the **.active** file are unloaded.
- The **-p** option must be specified to remove the specified LDAP repository from the LDAP server.
- In the **unldadm** and **unsubcommand**, the **name** parameter indicates the admin policy name on the admin DN.

Exit Status

Item	Description
0	The command completed successfully.
>0	An error occurred.
1	Default error return code for read, write, and malloc failures.
2	EINVAL and ENOENT
3	EPERM and EACCES
4	EEXIST

Examples

1. To add a project **newproj** to the system project definition file, type:

```
projctl add newproj 34 "Test Project"
```

2. To remove the project **test1** from the project definition file under the path **/tmp/myproj**, type:

```
projctl rm test1 -d /tmp/myproj
```

3. To enable the aggregation status of the project **newproj**, type:

```
projctl chattr agg newproj -s
```

4. To execute the **ps** command under the project **newproj**, overriding the existing rules, type:

```
projctl exec newproj "/usr/bin/ps" -f
```

5. To retrieve the currently loaded policies, type:

```
projctl qpolicy
```

Output:

```
Project definitions are loaded.
Project definition file name: /etc/project/projdef
User policies are loaded.
```

6. To load the admin policies from the path **/tmp/myproj**, type:

```
projctl ldadm -d /tmp/myproj
```

7. To unload all the project policies now and during system reboot, type:

```
projctl unldall -a
```

8. To add a new project to the LDAP repository on a different DN, where DN is `ou=projects,ou=acct,ou=cluster1,cn=aixdata`, type:

```
projectl add newproj 34 -p ou=projects,ou=acct,ou=cluster1,cn=aixdata
```

9. To download the LDAP projects from the default DN to a local file under the `/etc/project/ldap` path, type:

```
projectl ldprojs -g -d /etc/project/ldap
```

10. To load the LDAP admin policies stored under the label `newdef` in the default DN to the kernel, type:

```
projectl ldadm -g newdef
```

Location

`/usr/bin/projectl`

Files

Item	Description
<code>/usr/bin/projectl</code>	Contains the projectl command.
<code>/etc/project/projdef</code>	Contains the system project definition file.
<code>/etc/project/ldap/projdef</code>	Contains the default LDAP project definition file.
<code>/etc/project/.active</code>	Contains the status of currently loaded policies.
<code>/etc/project/.config</code>	Contains the status of the policies to be loaded during system reboot.
<code>/etc/security/ldap/ldap.cfg</code>	Contains the LDAP client configuration details for handling advanced accounting data.

prompter Command

Purpose

Starts a prompting editor.

Syntax

```
prompter [ -erase Character ] [ -kill Character ] [ -prepend | -nopprepend ] [ -rapid | -norapid ] File
```

Description

Part of the Message Handler (MH) package, the **prompter** command starts the prompting editor for message entry. The **prompter** command is not started by the user. The **prompter** command is called by other programs only.

The **prompter** command opens the file specified by the *File* parameter, scans it for empty components such as the `To :` component, and prompts you to fill in the blank fields. If you press the Enter key without filling in a required field, the **prompter** command deletes the component.

The **prompter** command accepts text for the body of the message after the first blank line or line of dashes in the file. If the body already contains text and the **-nopprepend** flag is specified, the **prompter** command displays the text followed by the message:

```
-----Enter additional text
```


The **prompter** command appends any new text entered after the existing message. If you specify the **-prepend** flag, the **prompter** command displays the following message:

```
-----Enter initial text
```

Any new text precedes the body of the original message. When you press the Ctrl-D key sequence for End of File, the **prompter** command ends text entry and returns control to the calling program.

Flags

Item	Description
-erase <i>Character</i>	Sets the character to be used as the erase character. The value of the <i>Character</i> variable can be the octal representation of the character in the form <code>\NNN</code> where <code>\NNN</code> is a number or the character itself. For example, the character <code>\e</code> is <code>\145</code> in octal representation.
-help	Lists the command syntax, available switches (toggles), and version information. Note: For MH, the name of this flag must be fully spelled out.
-kill <i>Character</i>	Sets the character to be used as the kill, or stop, character. The value of the <i>Character</i> variable can be the octal representation of the character in the form <code>\NNN</code> where <code>\NNN</code> is a number or the character itself. For example, the character <code>\e</code> is <code>\145</code> in octal representation.
-noprepend	Appends additional text after text already in the message body.
-norapid	Displays text already in the message body. This is the default.
-prepend	Appends additional text before text already in the message body. This is the default.
-rapid	Does not display text already in the message body.

Profile Entries

Item	Description
<code>Msg-Protect:</code>	Sets the protection level for your new message files.
<code>prompter-next:</code>	Specifies the editor used after exiting the prompter 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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Files

Item	Description
<code>\$HOME/html</code>	

proto Command

Purpose

Constructs a prototype file for a file system.

Syntax

proto *Directory* [*Prefix*]

Description

The **proto** command creates a prototype file for a file system or part of a file system. The **mkfs** command.

Specify the root directory from which the prototype file is made with the *Directory* parameter. The prototype file includes the complete subtree below the *Directory* parameter, and is contained on the same file system as the base directory specified by the *Directory* parameter.

The *Prefix* parameter is added to the names of all the initialization files, forcing the initialization files to be taken from a place other than the prototype. Before the output from the **proto** command can be used with the **LC_COLLATE** environment variables.

Example

To make a prototype file for an existing file system `/works`, enter:

```
proto /works
```

If the `/works` file system contains two directories called `dir1` and `dir2`, and the `dir1` directory contains the `file1` file, then the **proto** command displays:

```
#Prototype file for /works
d--- 755 0 0
  dir1 d--- 755 0 0
    file1 ---- 644 0 0 /works/dir1/file1
      $
  dir2 d--- 755 0 0
    $
  $
$
```

Files

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

proxymngr Command

Purpose

Proxy manager service.

Syntax

proxymngr [**-config** filename] [**-timeout** seconds] [**-retries** #] [**-verbose**]

Description

The **proxymngr** (proxy manager), is responsible for resolving requests from **xfindproxy** (and other similar clients), starting new proxies when appropriate, and keeping track of all of the available proxy services. The proxy manager strives to reuse existing proxies whenever possible.

There are two types of proxies that the proxy manager deals with, managed and unmanaged proxies.

A managed proxy is a proxy that is started on demand by the proxy manager.

An unmanaged proxy is started either at system boot time, or manually by a system administrator. The proxy manager is made aware of its existence, but no attempt is made by the proxy manager to start unmanaged proxies.

Flags

Item	Description
-config	Overrides the default proxymngr config file. See below for more details about the proxymngr config file.
-timeout	Sets the number of seconds between attempts made by the proxy manager to find an unmanaged proxy. The default is 10.
-retries	Sets the maximum number of retries made by the proxy manager to find an unmanaged proxy. The default is 3.
-verbose	Causes various debugging and tracing records to be displayed as requests are received and proxies are started.

Proxy Manager Config File

The proxy manager maintains a local configuration file describing the proxy services available. This configuration file is installed in **/usr/X11R6.3/lib/X11/proxymngr/pmconfig** during the installation of **proxymngr**. The location of the configuration file can be overwritten using the **-config** command line flag.

Aside from lines starting with an exclamation point for comments, each line of the configuration file describes either an unmanaged or managed proxy service.

For unmanaged proxies, the format is:

```
<service-name> unmanaged <proxy-address>
```

service-name is the name of the unmanaged proxy service, and must not contain any spaces, for example XFWP. *service-name* is case insensitive.

proxy-address is the network address of the unmanaged proxy. The format of the address is specific to the *service-name*. For example, for the XFWP service, the *proxy-address* might be `firewall.x.org:100`.

If there is more than one entry in the config file with the same unmanaged *service-name*, the proxy manager will try to use the proxies in the order presented in the config file.

For managed proxies, the format is:

```
<service-name> managed <command-to-start-proxy>
```

service-name is the name of the managed proxy service, and must not contain any spaces, for example LBX. *service-name* is case insensitive.

command-to-start-proxy is the command executed by the proxy manager to start a new instance of the proxy. If *command-to-start-proxy* contains spaces, the complete command should be surrounded by single quotes. If desired, *command-to-start-proxy* can be used to start a proxy on a remote machine. The specifics of the remote execution method used to do this is not specified here.

Example: sample configuration file

```

! proxy manager config file
!
! Each line has the format:
!   <serviceName> managed <startCommand>
!   or
!   <serviceName> unmanaged <proxyAddress>
!
lbx managed /usr/X11R6.3/bin/lbxproxy
!
! substitute site-specific info
xftp unmanaged firewall:4444

```

Proxy Manager Details

When the proxy manager gets a request from **xfindproxy** (or another similar client), its course of action will depend on the *service-name* in question.

For a managed proxy service, the proxy manager will find out if any of the already running proxies for this service can handle a new request. If not, the proxy manager will attempt to start up a new instance of the proxy (using the *command-to-start-proxy* found in the config file). If that fails, an error will be returned to the caller.

For an unmanaged proxy service, the proxy manager will look in the config file to find all unmanaged proxies for this service. If there is more than one entry in the config file with the same unmanaged *service-name*, the proxy manager will try to use the proxies in the order presented in the config file. If none of the unmanaged proxies can satisfy the request, the proxy manager will timeout for a configurable amount of time (specified by **-timeout** or default of 10) and reattempt to find an unmanaged proxy willing to satisfy the request. The number of retries can be specified by the **-retries** argument, or a default of 3 will be used. If the retries fail, the proxy manager has no choice but to return an error to the caller (since the proxy manager can not start unmanaged proxy services).

prs Command (SCCS)

Purpose

Displays a Source Code Control System (SCCS) file.

Syntax

```
prs [-a] [-d String] [-r [SID] | [-c Cutoff]] [-e | -l] File ...
```

Description

The **prs** command first reads the specified files and then writes to standard output a part or all of a Source Code Control System (SCCS) file. If you specify a directory for the *File* parameter, the **prs** command performs the requested actions on all SCCS files (those with the **s.** prefix). If you specify a - (minus) for the *File* parameter, the **prs** command reads standard input and interprets each line as the name of an SCCS file. The **prs** command continues to read input until it reaches an end-of-file character.

Data Keywords

Data keywords specify the parts of an SCCS file to be retrieved and written to standard output. All parts of an SCCS file have an associated data keyword. There is no limit to the number of times a data keyword can be in a specified file.

The information that the **prs** command displays consists of user-supplied text and appropriate values (extracted from the SCCS file) substituted for the recognized data keywords in the order they are displayed in the specified file. The format of a data keyword value is either simple, in which the keyword substitution is direct, or multiline, in which the substitution is followed by a carriage return. Text consists of any characters other than recognized data keywords. Specify a tab character with **\t** (backslash, letter t) and a carriage return or new-line character with a **\n** (backslash, letter n). Remember to use the **\t** and

\n with an extra \ (backslash) to prevent the shell from interpreting the \ and passing only the letter **t** or **n** to the **prs** command as text.

The following table lists the keywords associated with information in the delta table of the SCCS file. All the keywords have a simple format unless otherwise indicated.

Delta Table Keywords

Keyword	Data Represented	Value
:R:	Release number	num
:L:	Level number	num
:B:	Branch number	num
:S:	Sequence number	num
:I:	SCCS ID string (SID)	:R::L::B::S:
:Dy:	Year delta created	YY
:Dm:	Month delta created	MM
:Dd:	Day delta created	DD
:D:	Date delta created	YY/MM/DD
:Th:	Hour delta created	HH
:Tm:	Minute delta created	MM
:Ts:	Second delta created	SS
:T:	Time delta created	HH/MM/SS
:DT:	Delta type	D or R
Item	Description	Value
:P:	User who created the delta	login name
:DS:	Delta sequence number	num
:DP:	Previous delta sequence number	num
:Dt:	Delta information	:DT::I::D::T::P::DS::DP:
:Dn:	Sequence numbers of deltas included	:DS: . . .
:Dx:	Sequence numbers of deltas excluded	:DS: . . .
:Dg:	Sequence numbers of deltas ignored	:DS: . . .
:DI:	Sequence numbers of deltas included,excluded, and ignored	:Dn:/:Dx:/:Dg:
:Li:	Lines inserted by delta	num

Item	Description	Value
:Ld:	Lines deleted by delta	num
:Lu:	Lines unchanged by delta	num
:DL:	Delta line statistics	:Li:/:Ld:/:Lu:
:MR: (multiline format)	MR numbers for delta	text
:C: (multiline format)	Comments for delta	text

The following table lists the keywords associated with header flags in the SCCS file. All the keywords have a simple format unless otherwise indicated.

Header Flag Keywords

Keyword	Data Represented	Value
:Y:	Module type	text
:MF:	MR validation flag set	yes or no
:MP:	MR validation program name	text
:KF:	Keyword/error warning flag set	yes or no
:BF:	Branch flag set	yes or no
:J:	Joint edit flag set	yes or no
:LK:	Locked releases	:R: . . .
:Q:	User-defined keyword	text
:M:	Module name	text
:FB:	Floor boundary	:R:
:CB:	Ceiling boundary	:R:
:Ds:	Default SID	:I:
:ND:	Null Delta flag set	yes or no
:FL: (multiline format)	Header flag list	text

The following table lists the keywords associated with other parts of the SCCS file. All the keywords have a simple format unless otherwise indicated.

Other Keywords

Keyword	Data Represented	Value
:UN: (multiline format)	User names	text

Other Keywords (*continued*)

Keyword	Data Represented	Value
:FD: (multiline format)	Descriptive text	text
:BD: (multiline format)	Body of text	text
:GB: (multiline format)	Text in a g-file	text
:W:	A what string	:Z::M: \tab :I:
:A:	A what string	:Z::Y::M::I::Z:
:Z:	A what string delimiter	@(#)
:F:	SCCS file name	text
:PN:	SCCS file path name	text

Flags

Each flag or group of flags applies independently to each named file.

Item	Description
-a	Writes information for the specified deltas, whether or not they have been removed (see the rmDEL command). If you do not specify the -a flag, the prs command supplies information only for the specified deltas that have not been removed.
-c Cutoff	Specifies a cutoff date and time for the -e and -l flags. Specify the <i>Cutoff</i> value in the following form: <pre>YY[MM[DD[HH[MM[SS]]]]]</pre> All omitted items default to their maximum values, so specifying -c8402 is the same as specifying -c840229235959 . You can separate the fields with any non-numeric character. For example, you can specify -c84/2/20,9:22:25 or -c"84/2/20 9:22:25" or -c84/2/20 9:22:25" . The -c flag cannot be specified with the -r flag.
-d String	Specifies the data items to be displayed. The string consists of optional text and SCCS file-data keywords. The string may include MBCS (multibyte character set) characters. If the string contains spaces, you must enclose the string in quotation marks.
-e	Requests information for all deltas created earlier than and including the delta specified by the -r flag.
-l	Requests information for all deltas created later than and including the delta specified by the -r flag.
-r [SID]	Specifies the SCCS ID string (SID) of the delta for which the prs command will retrieve information. Do not enter a space between the -r flag and the optional SID parameter. If no SID is specified, the command retrieves the information for the SID of the highest numbered delta. The -r flag cannot be specified with the -c flag.

Exit Status

This command returns the following exit values:

Item Description

- 0 Successful completion.
- >0 An error occurred.

Examples

1. To display information on all deltas generated for SCCS file name **s.test.c** (including all deltas removed using the **rmdel** command), type:

```
prs -a s.test.c
```

2. To display user login name, the number of lines inserted by delta, and the number of lines deleted by delta for SID 1.2 of **s.test.c**, type:

```
prs -r1.2 -d":P:\n:Li:\n:Ld:" s.test.c
```

Files

Item	Description
/usr/bin/prs	Contains the prs command.

prtacct Command

Purpose

Formats and displays files in **taacct** format.

Syntax

```
/usr/sbin/acct/prtacct [ -X ] [ -W ] [ -f Fields ] [ -v ] File [ "Heading" ]
```

Description

The **prtacct** command formats and displays any total-accounting file; these files are in **taacct** format. You can enter this command to view any **taacct** file, such as the daily reports on connect time, process time, disk usage, and printer usage. To specify a title for the report with the *Heading* parameter, enclose the heading text in " " (quotation marks).

Flags

Item	Description
-f Fields	Selects fields to be displayed, using the field-selection mechanism of the acctmerg command.
-v	Produces verbose output in which more precise notation is used for floating-point numbers.
-W	Processes all available characters for each user name instead of truncating to the first 8 characters. This flag causes the prtacct command to expect to read in taacctx structures. It will then print out in the same column order, but it will allow long user names to misalign the columns. If the -W flag and the -X flag are used together, the -X takes precedence.

Item	Description
-X	Processes all available characters for each user name instead of truncating to the first 8 characters. This flag causes the prtacct command to expect to read in tacctx structures and print out the user name in the last column. If the -W flag and the -X flag are used together, the -X will take precedence.

Security

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

Examples

To format and display selected records from the total accounting file for connect-time accounting, you first must create a file upon which to execute the **prtacct** command. In this example, you create the **tacct** file using the **acctcon1** and **acctcon2** commands. Enter:

```
tail /var/adm/wtmp > wtmp.sav
acctcon1 -t < wtmp.sav | sort +1n +2 | acctcon2 > tacct
```

If you created this file previously to process connect-time accounting data, you do not need to create it again.

The next step uses the **prtacct** command with the **-f** flag to display the fields of data in the total-accounting file that you want to see. The text for a heading can be included in quotation marks. To view the login name, prime connect-time, and nonprime connect-time records, and include the heading, Connect-time Accounting, enter:

```
prtacct -f 2,11,12 tacct "Connect-time Accounting"
```

You can also use this command to format and display other total-accounting files, such as the daily reports on process time, disk usage, and printer usage.

Files

Item	Description
/usr/sbin/acct	The path to the accounting commands.
/var/adm/pacct	Current file for process accounting.
/var/adm/pacct*	Used if the pacct file gets too large.

prtconf Command

Purpose

Displays system configuration information.

Syntax

```
prtconf [ -c ] [ -k ] [ -L ] [ -m ] [ -s ] [ -v ]
```

Description

If you run the **prtconf** command without any flags, it displays the system model, machine serial, processor type, number of processors, processor clock speed, cpu type, total memory size, network information, filesystem information, paging space information, and devices information.

Flags

Item	Description
-c	Displays cpu type, for example, 32-bit or 64-bit.
-k	Display the kernel in use, for example, 32-bit or 64-bit.
-L	Displays LPAR partition number and partition name if this is an LPAR partition, otherwise returns "-1 NULL".
-m	Displays system memory.
-s	Displays processor clock speed in MegaHertz.
-v	Displays the VPD found in the Customized VPD object class for devices.

Exit Status

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

Examples

1. To display the system configuration information, enter:

```
prtconf
```

The system displays a message similar to the following:

```
System Model: IBM,7025-F50
Machine Serial Number: 1025778
Processor Type: PowerPC_604
Number Of Processors: 2
Processor Clock Speed: 332 MHz
CPU Type: 32-bit
Kernel Type: 32-bit
LPAR Info: -1 NULL
Memory Size: 512 MB
Good Memory Size: 512 MB
Firmware Version: IBM,L02113
Console Login: enable
Auto Restart: false
Full Core: false

Network Information
Host Name: vd01.austin.ibm.com
IP Address: 9.3.207.112
Sub Netmask: 255.255.255.128
Gateway: 9.3.207.1
Name Server: 9.3.199.2
Domain Name: austin.ibm.com
Paging Space Information

Total Paging Space: 512MB
Percent Used: 1%

Volume Groups Information
=====
rootvg:
PV_NAME          PV STATE          TOTAL PPs   FREE PPs   FREE DISTRIBUTION
hdisk0           active            537         394        107..43..29..107..108
=====

INSTALLED RESOURCE LIST
```

The following resources are installed on the machine.
+/- = Added or deleted from Resource List.
* = Diagnostic support not available.

Model Architecture: chrp
Model Implementation: Multiple Processor, PCI bus

+ sys0 00-00 System Object
+ sysplanar0 00-00 System Planar
+ mem0 00-00 Memory
etc.

2. To display the processor clock speed, enter:

```
prtconf -s
```

The system displays a message similar to the following:

```
Processor Clock Speed: 332 MHz
```

3. To display the VPD for all physical devices in the Customized database, enter:

```
prtconf -v
```

The system displays a message similar to the following:

```
INSTALLED RESOURCE LIST WITH VPD
```

The following resources are installed on your machine.

Model Architecture: chrp
Model Implementation: Uni-Processor, PCI bus

```
sys0          P1-C1      System Object
sysplanar0    P1-C1      System Planar
mem0          P1-C1      Memory
L2cache0     P1-C1      L2 Cache
proc0        P1-C1      Processor
              Device Specific.(YL).....P1-C1
pci0          P1        PCI Bus
              Device Specific.(YL).....P1
isa0          P1        ISA Bus
              Device Specific.(YL).....P1
fda0          P1/D1     Standard I/O Diskette Adapter
              Device Specific.(YL).....P1/D1
fd0           P1-D1     Diskette Drive
siokma0       P1/K1     Keyboard/Mouse Adapter
              Device Specific.(YL).....P1/K1
sioka0        P1-K1     Keyboard Adapter
kbd0          P1-K1-Lkbd PS/2 keyboard
sioma0        P1-01     Mouse Adapter
mouse0        P1-01-Lmouse3 button mouse
siota0        P1/Q1     Tablet Adapter
              Device Specific.(YL).....P1/Q1
paud0         P1/Q2     Ultimedia Integrated Audio
              Device Specific.(YL).....P1/Q2
ppa0          P1/R1     CHRP IEEE1284 (ECP) Parallel Port Adapter
              Device Specific.(YL).....P1/R1
sa0           P1/S1     Standard I/O Serial Port
              Device Specific.(YL).....P1/S1
```

```

tty0          P1/S1-L0   Asynchronous Terminal
sa1          P1/S2     Standard I/O Serial Port

Device Specific.(YL).....P1/S2

ent0          P1/E1     IBM 10/100 Mbps Ethernet PCI Adapter (23100020)

Network Address.....0004AC2A0419
Displayable Message.....PCI Ethernet Adapter (23100020)
Device Specific.(YL).....P1/E1

scsi0         P1/Z1     Wide/Fast-20 SCSI I/O Controller

Device Specific.(YL).....P1/Z1

cd0           P1/Z1-A3   SCSI Multimedia CD-ROM Drive (650 MB)

Manufacturer.....IBM
Machine Type and Model.....CDRM00203
ROS Level and ID.....1_00
Device Specific.(Z0).....058002028F000018
Part Number.....97H7608
EC Level.....F15213
FRU Number.....97H7610

hdisk0        P1/Z1-A5   16 Bit SCSI Disk Drive (4500 MB)

Manufacturer.....IBM
Machine Type and Model.....DDRS-34560W
FRU Number.....83H7105
ROS Level and ID.....53393847
Serial Number.....RDHW5008
EC Level.....F21433
Part Number.....03L5256
Device Specific.(Z0).....000002029F00003A
Device Specific.(Z1).....00K0159S98G
Device Specific.(Z2).....0933
Device Specific.(Z3).....0299
Device Specific.(Z4).....0001
Device Specific.(Z5).....22
Device Specific.(Z6).....F21390

bl0           P1.1-I2/G1  GXT255P Graphics Adapter

GXT255P 2D Graphics Adapter:
EC Level.....E76756
FRU Number.....93H6267
Manufacture ID.....IBM053
Part Number.....93H6266
Serial Number.....88074164
Version.....RS6K
Displayable Message.....GXT255P
ROM Level.(alterable).....02
Product Specific.(DD).....00
Product Specific.(DG).....00
Device Specific.(YL).....P1.1-I2/G1

pci1          P1.1     PCI Bus

Device Specific.(YL).....P1.1

```

4. To display the kernel type in use, type:

```
prtconf -k
```

The system displays information for the kernel type as follows:

```
Kernel Type: 32-bit
```

5. To display memory, type:

```
prtconf -m
```

The system displays memory, as follows:

```
Memory Size: 512 MB
```

Files

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

prtgbldconfig Command

Purpose

The **prtgbldconfig** command configures the global settings for the AIX printing subsystem.

Syntax

```
prtgbldconfig [ -s name = value ] [ -r name ]
```

Description

The **prtgbldconfig** command either sets a printing subsystem setting or resets it to a default value. Currently, this command is used to set the ERRMSGCONTROL setting. This setting affects the global printer message. This setting can be used to select one of the following options:

- ALLON (All messages turned on).
- LOGALL (All messages turned on, but logged to a log file).
- CRITON (Only the most critical error messages turned on).
- ALLOFF (All messages turned off).

Currently, the LOGALL option and the CRITON option is same as the ALLON option.

Flags

Item	Description
-s <i>name</i> = <i>value</i>	Specifies that the printing subsystem setting specified in the <i>name</i> parameter is set with the value specified in the <i>value</i> parameter.
-r <i>name</i>	Resets the printing subsystem setting specified in the <i>name</i> parameter to a default value.

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 set the printing subsystem to ignore all of the messages generated by the printing subsystem, enter:

```
prtgbldconfig -s ERRMSGCONTROL=ALLOFF
```

2. To reset the error message control of the printing subsystem to a default value, enter:

```
prtgbldconfig -r ERRMSGCONTROL
```

Note: Some messages generated by the printing subsystem cannot be ignored and are logged in the console log file. For more information about starting a print job, see the **qprt** command.

Files

Item	Description
<code>/etc/prtglobalconfig</code>	Contains the global configuration file.
<code>/usr/sbin/qdaemon</code>	Contains the qdaemon daemon.
<code>/etc/qconfig</code>	Contains the configuration file.
<code>/etc/qconfig.bin</code>	Contains the digested, binary version of the <code>/etc/qconfig</code> file.

ps Command

Purpose

Shows status of processes. This document describes the standard AIX **ps** command as well as the [System V](#) version of the **ps** command.

Syntax

X/Open Standards

```
ps [ -A ] [ -M ] [ -N ] [ -Z ] [ -a ] [ -d ] [ -e ] [ -f ] [ -k ] [ -l ] [ -F format ] [ -o Format ] [ -c Clist ] [ -G Glist ] [ -g Glist ] [ -m ] [ -n NameList ] [ -p Plist ] [ -P ] [ -t Tlist ] [ -U Ulist ] [ -u Ulist ] [ -T pid ] [ -L pidlist ] [ -X ] [ -@ [ WparName ] ]
```

Berkeley Standards

```
ps [ a ] [ c ] [ e ] [ ew ] [ eww ] [ ewww ] [ g ] [ n ] [ w ] [ x ] [ l | s | u | v ] [ t tty ] [ X ] [ ProcessNumber ]
```

Description

The **ps** command writes the status of active processes and if the **-m** flag is given, displays the associated kernel threads to standard output. While the **-m** flag displays threads associated with processes using extra lines, you must use the **-o** flag with the **THREAD** field specifier to display extra thread-related columns.

Without flags, the **ps** command displays information about the current terminal. The **-f**, **-o**, **l**, **-l**, **s**, **u**, and **v** flags only determine how much information is provided about a process; they do not determine which processes are listed. The **l**, **s**, **u**, and **v** flags are mutually exclusive.

With the **-o** flag, the **ps** command examines memory or the paging area and determines what the command name and parameters were when the process was created. If the **ps** command cannot find this information, the command name stored in the kernel is displayed in square brackets.

The **COLUMNS** environment variable overrides the system-selected, horizontal screen size.

The command-line flags that accept a list of parameters (the **-o**, **-G**, **-g**, **-p**, **-t**, **-U**, and **-u** flags) are limited to 128 items. For example, the **-u Ulist** flag can specify no more than 128 users.

For cases in which the output of the **ps** command does not include workload partition (WPAR) names but does include Project IDs (**PROJECT**), User IDs (**UID** or **USER**), or Group IDs (**GID**) associated with a process running within a workload partition under the current operating environment, the IDs are preceded by a plus sign (+) to indicate the association with a workload partition. Each workload partition contains its own definition of users, groups, and project IDs that may be different from the IDs defined for the global environment. The **-@** option may be specified to include workload partition names in the output.

Note: The **ps** command does not show the decrease in the memory usage count when the application releases the memory. When the memory is released from the application, the memory is assigned to the per process memory freelist. The **ps** command accounts the memory that is released as the allocated memory for the application.

Depending on the flags used with the **ps** command, column headings are displayed above the information displayed to standard output. The headings are defined in the following list and the flags that cause these headings to be displayed are shown in parentheses :

ADDR

(**-l** and **l** flags) Contains the segment number of the process stack, if normal; if a kernel process, the address of the pre-process data area.

BND

(**-o THREAD** flag) The logical processor number of the processor to which the kernel thread is bound if any. For a process, this field is displayed if all its threads are bound to the same processor.

C

(**-f**, **l**, and **-l** flags) CPU utilization of process or thread, incremented each time the system clock ticks and the process or thread is found to be running. The value is decayed by the scheduler by dividing it by 2 once per second. For the `sched_other` policy, CPU utilization is used in determining process scheduling priority. Large values indicate a CPU intensive process and result in lower process priority, whereas small values indicate an I/O intensive process and result in a more favorable priority.

CMD

(**-f**, **-l**, and **l** flags) Contains the command name. Under the **-f** flag, the **ps** command tries to determine the current command name and arguments, both of which may be changed asynchronously by the process. These are then displayed. If this fails, the command name is written as it would appear without the **-f** option in square brackets.

COMMAND

(**s**, **u**, and **v**) Contains the command name. The full command name and its parameters are displayed with the **-f** flag.

F Field Table		
Flags	Hexadecimal Value	Definition
SLOAD	0x00000001	Indicates that the process is operating in core memory.
SNOSWAP	0x00000002	Indicates that the process cannot be swapped out.
STRC	0x00000008	Indicates that the process is being traced.
SWTED	0x00000010	Indicates that the process stopped while being traced.
SFWTED	0x00000020	Indicates that the process stopped after a call to the fork subroutine, while being traced.
SEWTED	0x00000040	Indicates that the process stopped after a call to the exec subroutine, while being traced.
SLWTED	0x00000080	Indicates that the process stopped after a call to the load or unload subroutine, while being traced.
SFIXPRI	0x00000100	Indicates that the process has a fixed priority, ignoring the pcpu field descriptor.
SKPROC	0x00000200	Indicates a Kernel process.
SOMASK	0x00000400	Indicates restoration of the old mask after a signal is received.

F Field Table (continued)		
Flags	Hexadecimal Value	Definition
SWAKEONSIG	0x00000800	Indicates that the signal will abort the sleep subroutine. The contents must <i>not</i> be equal to those of the PCATCH flag. The contents of both PCATCH and SWAKEONSIG must be greater than those of PMASK .
SUSER	0x00001000	Indicates that the process is in user mode.
SLKDONE	0x00002000	Indicates that the process has done locks.
STRACING	0x00004000	Indicates that the process is a debugging process.
SMPTRACE	0x00008000	Indicates multi-process debugging.
SEXIT	0x00010000	Indicates that the process is exiting.
SSEL	0x00020000	Indicates that the processor is selecting: wakeup/waiting danger.
SORPHANPGRP	0x00040000	Indicates an orphaned process group.
SNOCNTLPROC	0x00080000	Indicates that the session leader relinquished the controlling terminal.
SPPNOCLDSTOP	0x00100000	Indicates that the SIGHLD signal is <i>not</i> sent to the parent process when a child stops.
SEXECED	0x00200000	Indicates that process has been run.
SJOBSESS	0x00400000	Indicates that job control was used in the current session.
SJOBOFF	0x00800000	Indicates that the process is free from job control.
PSIGDELIVERY	0x01000000	Indicates that the process is used by the program-check handler.
SRMSHM	0x02000000	Indicates that the process removed shared memory during a call to the exit subroutine.
SSLOTFREE	0x04000000	Indicates that the process slot is free.
SNOMSG	0x08000000	Indicates that there are no more uprintf subroutine messages.

WPAR

(-@ flag) Contains the workload partition name. Under the -@ flag, the **ps** command displays the name of the workload partition in which the process is running. Specify the -@ flag with the *wparname* parameter to display the process information.

DPGSZ

(Z flag) The data page size of the process.

F

(-l and l flags) Some of the more important F field flags (hexadecimal and additive) associated with processes and threads are listed in the following table:

F Field Table		
Flags	Hexadecimal Value	Definition
SLOAD	0x00000001	Indicates that the process is operating in core memory.
SNOSWAP	0x00000002	Indicates that the process cannot be swapped out.
STRC	0x00000008	Indicates that the process is being traced.
SKPROC	0x00000200	Indicates a kernel process.
SEXIT	0x00010000	Indicates that the process is exiting.
SLPDATA	0x00020000	Indicates that the process uses large pages.
SEXECED	0x00200000	Indicates that the process has been run.
SEXECING	0x01000000	Indicates that the process is execing (performing an exec).
SPSEARLYALLOC	0x04000000	Indicates that paging space for this process is allocated early.
TKTHREAD	0x00001000	Indicates that the thread is a kernel-only thread.

Note: You can see the definitions of all process and thread flags by consulting the **p_flags** and **t_flags** fields in the **/usr/include/sys/proc.h** and **/usr/include/sys/thread.h** files respectively.

LIM

(**v** flag) The soft limit on memory used, specified through a call to the **setrlimit** subroutine. If the limit has not been specified, then **xx** is displayed. If the limit is set to the system limit (unlimited), a value of **UNLIM** is displayed.

NI

(**-l** and **l** flags) The nice value; used in calculating priority for the **sched_other** policy.

PID

(all flags) The process ID of the process.

PGIN

(**v** flag) The number of disk I/Os resulting from references by the process to pages not loaded in core.

PPID

(**-f**, **l**, and **-l** flags) The process ID of the parent process.

PRI

(**-l** and **l** flags) The priority of the process or kernel thread; higher numbers mean lower priority.

PROJECT

(**-P** flag) Project name assigned to the process. Under the current operating environment, the **PROJECT** and **USER** fields are not translated to names for processes running within a workload partition. The **-U** and **-u** flags only apply to the current operating environment, unless the **-@** flag is included with a specific workload partition name. If the **-@** flag is used to specify a workload partition other than the current operating environment, and the **-U** and **-u** flags are specified, the list of user IDs must be numeric.

RSS

(**v** flag) The real-memory (resident set) size of the process (in 1 KB units).

S

(**-l** and **l** flags) The state of the process or kernel thread :

For processes:

O Nonexistent
A Active
W Swapped
I Idle (waiting for startup)
Z Canceled
T Stopped

For kernel threads:

O Nonexistent
R Running
S Sleeping
W Swapped
Z Canceled
T Stopped

SC

(**-o THREAD** flag) The suspend count of the process or kernel thread. For a process, the suspend count is defined as the sum of the kernel threads suspend counts.

SCH

(**-o THREAD, sched** flag) The scheduling policy for a kernel thread. The policies `sched_other`, `sched_fifo`, and `sched_rr` are respectively displayed using: 0, 1, 2. The scheduling policies is displayed only when a **sched** flag is specified.

SIZE

(**v** flag) The virtual size of the data section of the process (in 1 KB units).

SHMPGSZ

(**Z** flag) The shared memory page size of the process.

SPGSZ

(**Z** flag) The stack page size of the process.

SSIZ

(**s** flag) The size of the kernel stack. This value is always 0 (zero) for a multi-threaded process.

STAT

(**s, u, and v** flags) Contains the state of the process:

O Nonexistent
A Active
I Intermediate

Z
Canceled

T
Stopped

K
Available kernel process

STIME

(**-f** and **u** flags) The starting time of the process. The **LANG** environment variables control the appearance of this field.

SUBPROJ

(**-P** flag) Subproject Identifier assigned to the process.

SZ

(**-l** and **l** flags) The size in 1 KB units of the core image of the process.

THCNT

(**-o thcount** flag) The number of kernel threads owned by the process.

TID

(**-o THREAD** flag) The thread ID of the kernel thread.

TIME

(all flags) The total runtime for the process. The time is displayed in the format of *mm:ss* or *mmm:ss* if the runtime reaches 100 minutes, which is different from the displayed format if you use the **-o time** flag.

TPGSZ

(**Z** flag) The text page size of the process.

TRS

(**v** flag) The size of resident-set (real memory) of text.

TSIZ

(**v** flag) The size of text (shared-program) image.

TTY

(all flags) The controlling terminal for the process:

-
The process is not associated with a terminal.

?
Unknown.

Number

The TTY number. For example, the entry 2 indicates TTY2.

UID

(**-f**, **-l**, and **l** flags) The user ID of the process owner. The login name is printed under the **-f** flag.

USER

(**u** flag) The login name of the process owner. Under the current operating environment, the **PROJECT** and **USER** fields are not translated to names for processes running within a workload partition.

WCHAN

(**-l** flag) The event for which the process or kernel thread is waiting or sleeping. For a kernel thread, this field is blank if the kernel thread is running. For a process, the wait channel is defined as the wait channel of the sleeping kernel thread if only one kernel thread is sleeping; otherwise a star is displayed.

WCHAN

(**l** flag) The event on which the process is waiting (an address in the system). A symbol that classifies the address is selected, unless a numerical output is requested.

%CPU

(**u** and **v** flags) The percentage of time the process has used the CPU since the process started. This value is computed by dividing the time the process uses the CPU by the elapsed time of the

process. In a multi-processor environment, the value is further divided by the number of available CPUs because several threads in the same process can run on different CPUs at the same time. (Because the time base over which this data is computed varies, the sum of all **%CPU** fields can exceed 100%.)

%MEM

(**u** and **v** flags) The percentage of real memory used by this process. The **%MEM** value tends to exaggerate the cost of a process that is sharing program text with other processes. It does not account for times when multiple copies of a program are run and a copy of the program text is shared by all instances. The size of the text section is accounted for in every instance of the program. This means that if several copies of a program are run, the total **%MEM** value of all processes could exceed 100%.

A process that has exited and has a parent that has not yet waited for the process is marked `<defunct>`. A process that is blocked trying to exit is marked `<exiting>`. The **ps** command attempts to determine the file name and arguments given when the process was created by memory or by the swap area.

Notes:

1. The process can change while the **ps** command is running. Some data displayed for defunct processes is irrelevant.
2. The **ps** program examines the memory to retrieve the file name and arguments used when the process was created. However, a process can destroy information, making this method of retrieving file name and arguments unreliable.
3. The **ps** program searches the local resources for users and group information.

Flags

The following flags are preceded by a - (minus sign):

Item	Description
-A	Writes to standard output information about all processes.
-a	Writes to standard output information about all processes, except the session leaders and processes not associated with a terminal.
-c <i>Clist</i>	Displays only information about processes assigned to the workload management classes listed in the <i>Clist</i> variable. The <i>Clist</i> variable is either a comma-separated list of class names or a list of class names enclosed in double quotation marks (" "), which is separated from one another by a comma or by one or more spaces, or both.
-d	Writes information to standard output about all processes, except the session leaders.
-e	Writes information to standard output about all processes, except kernel processes.
-F <i>Format</i>	Same as the -o <i>Format</i>
-f	Generates a full listing.
-G <i>Glist</i>	Writes information to standard output only about processes that are in the effective groups listed for the <i>Glist</i> variable. The <i>Glist</i> variable is either a comma-separated list of effective group identifiers, or a list of effective group identifiers enclosed in double quotation marks (" ") and separated from one another by a comma or by one or more spaces.
-g <i>Glist</i>	Writes information to standard output only about processes that are in the process groups listed for the <i>Glist</i> variable. The <i>Glist</i> variable is either a comma-separated list of process group identifiers or a list of process group identifiers enclosed in double quotation marks (" ") and separated from one another by a comma or by one or more spaces.
-k	Lists kernel processes.
-l	Generates a long listing. Also see the l flag.
-L <i>pidlist</i>	Generates a list of descendants of each and every pid that has been passed to it in the <i>pidlist</i> variable. The <i>pidlist</i> variable is a list of comma-separated process IDs. The list of descendants from all the given pid is printed in the order in which they appear in the process table.
-M	Lists all 64 bit processes.
-m	Lists kernel threads as well as processes. Output lines for processes are followed by an additional output line for each kernel thread. This flag does not display thread-specific fields (bnd , scount , sched , thcount , and tid), unless the appropriate -o <i>Format</i> flag is specified.
-N	Gathers no thread statistics. With this flag, ps reports those statistics that can be obtained by not traversing through the threads chain for the process.

Item	Description
-n <i>NameList</i>	Specifies an alternative system name-list file in place of the default. The operating system does not use the -n flag because information is supplied directly to the kernel.
-o <i>Format</i>	<p data-bbox="544 233 1471 321">Displays information in the format specified by the <i>Format</i> variable. Multiple field specifiers can be specified for the <i>Format</i> variable. The <i>Format</i> variable is either a comma-separated list of field specifiers or a list of field specifiers enclosed within a set of " " (double-quotation marks) and separated from one another by a comma or by one or more spaces, or both.</p> <p data-bbox="544 331 1471 443">Each field specifier has a default header. The default header can be overridden by appending an = (equal sign) followed by the user-defined text for the header. The fields are written in the order specified on the command-line in column format. The field widths are specified by the system to be at least as wide as the default or user-defined header text. If the header text is null (such as if -o user= is specified), the field width is at least as wide as the default header text. If all header fields are null, no header line is written.</p> <p data-bbox="544 453 1036 472">The following field specifiers are recognized by the system:</p> <p data-bbox="544 489 1430 554">args Indicates the full command name being executed. All command-line arguments are included, though truncation may occur. The default header for this field is COMMAND.</p> <p data-bbox="544 569 1463 634">bnd Indicates to which (if any) processor a process or kernel thread is bound. The default header for this field is BND.</p> <p data-bbox="544 648 1458 711">class Indicates the workload management class assigned to the process or thread. The default header for this field is CLASS.</p> <p data-bbox="544 726 1443 789">comm Indicates the short name of the command being executed. Command-line arguments are not included. The default header for this field is COMMAND.</p> <p data-bbox="544 804 1471 932">cpu Determines process scheduling priority. CPU utilization of a process or thread, incremented each time the system clock ticks and the process or thread is found to be running. The value is decayed by the scheduler by dividing it by 2 once per second. For the sched_other policy, Large values indicate a CPU intensive process and result in lower process priority whereas small values indicate an I/O intensive process and result in a more favorable priority.</p> <p data-bbox="544 947 932 989">dpgsz Indicates the data page size of a process.</p> <p data-bbox="544 1003 1417 1066">etime Indicates the elapsed time since the process started. The elapsed time is displayed in the following format:</p> <p data-bbox="589 1081 732 1100">[[<i>dd</i>-]<i>hh</i>:]<i>mm</i>:<i>ss</i></p> <p data-bbox="589 1115 1438 1157">where <i>dd</i> specifies the number of days, <i>hh</i> specifies the number of hours, <i>mm</i> specifies the number of minutes, and <i>ss</i> specifies the number of seconds. The default header for this field is ELAPSED.</p> <p data-bbox="544 1171 1455 1234">group Indicates the effective group ID of the process. The textual group ID is displayed. If the textual group ID cannot be obtained, a decimal representation is used. The default header for this field is GROUP.</p> <p data-bbox="544 1249 1349 1287">nice Indicates the decimal value of the process nice value. The default header for this field is NI.</p>

Item

-o *Continued*

Description**pcpu**

Indicates the ratio of CPU time used to CPU time available, expressed as a percentage. The default header for this field is **%CPU**.

pgid

Indicates the decimal value of the process group ID. The default header for this field is **PGID**.

pid

Indicates the decimal value of the process ID. The default header for this field is **PID**.

ppid

Indicates the decimal value of the parent process ID. The default header for this field is **PPID**.

rgroup

Indicates the real group ID of the process. The textual group ID is displayed. If the textual group ID cannot be obtained, a decimal representation is used. The default header for this field is **RGROUP**.

ruser

Indicates the real user ID of the process. The textual user ID is displayed. If the textual user ID cannot be obtained, a decimal representation is used. The default header for this field is **RUSER**.

scount

Indicates the suspend count for a kernel thread. The default header for this field is **SC**.

sched

Indicates the scheduling policy for a kernel thread. The default header for this field is **SCH**.

shmpgsz

Indicates the shared memory page size of a process.

spgsz

Indicates the stack page size of a process.

tag

Indicates the Workload Manager application tag. The default header for this field is **TAG**. The tag is a character string up to 30 characters long and may be truncated when displayed by **ps**. For processes that do not set their tag, this field displays as a - (hyphen).

tcpu

Total CPU time. Indicates the total accumulated CPU time for a single process. The command displays the information when WLM is running either in active or passive mode else, this field displays as a - (hyphen). The default header for this field is **TCPU**.

tctime

Total connect time. Indicates the total amount of time that a login session can be active. This is meaningful only in the case of session leader processes. The default header for this field is **TCTIME**.

tdiskio

Total disk I/O. Indicates the total accumulated blocks of disk I/O for a single process. The default header for this field is **TDISKIO**.

tpgsz

Indicates the text page size of a process.

vmsize

Indicates the WLM virtual memory limits. When this is used, a new header, **VMSIZ** is displayed. **VMSIZ** displays the virtual memory used by the process. This value is displayed in 1 MB units.

thcount

Indicates the number of kernel threads owned by the process. The default header for this field is **THCNT**.

Item`-o Continued`**Description****THREAD**

Indicates the following fields:

- User name (the **uname** field)
- Process and parent process IDs for processes (the **pid** and **ppid** fields)
- Kernel thread ID for threads (the **tid** field)
- The state of the process or kernel thread (the **S** field)
- The CPU utilization of the process or kernel thread (the **C** field)
- The priority of the process or kernel thread (the **PRI** field)
- The suspend count of the process or kernel thread (the **scount** field)
- The wait channel of the process or kernel thread (the **WCHAN** field)
- The flags of the process or kernel thread (the **F** field)
- The controlling terminal of the process (the **tty** field)
- The CPU to which the process or kernel thread is bound (the **bnd** field)
- The command being executed by the process (the **comm** field).

Threads are not displayed with the **-o THREAD** flag, unless the **-m** flag is also specified.**Note:** The `ps -o THREAD` flag does not print the scheduler policies. The scheduling policies are displayed only when a **sched** flag is specified.**tid**Indicates the thread ID of a kernel thread. The default header for this field is **TID**.**time**

Indicates the cumulative CPU time since the process started. The time is displayed in the following format:

`[dd-] hh:mm:ss`where *dd* specifies the number of days, *hh* specifies the number of hours, *mm* specifies the number of minutes, and *ss* specifies the number of seconds. The default header for this field is **TIME**.**tty**Indicates the controlling terminal name of the process. The default header for this field is **TT**.**user**Indicates the effective user ID of the process. The textual user ID is displayed. If the textual user ID cannot be obtained, a decimal representation is used. The default header for this field is **USER**.**vsz**Indicates, as a decimal integer, the size in kilobytes of the process in virtual memory. The default header for this field is **VSZ**.Otherwise, multiple fields in a specified format can be displayed by the *Format* variable, including field descriptors. If field descriptors are used in the *Format* variable, it must be enclosed in double quotation marks (" "). The following table shows how field descriptors correspond to field specifiers:

Field Descriptors	Field Specifiers	Default Headers
%a	args	COMMAND
%c	comm	COMMAND
%t	etime	ELAPSED
%D	dpgsz	DPGSZ
%G	group	GROUP
%n	nice	NI
%C	pcpu	%CPU
%r	pgid	PGID
%p	pid	PID
%P	ppid	PPID
%g	rgroup	RGROUP
%u	ruser	RUSER
%S	spgsz	SPGSZ
%x	time	TIME
%T	tpgsz	TPGSZ gd
%y	tty	TTY
%U	user	USER
%z	vsz	VSZ

Each field specifier has a default header. The default header can be overridden by appending an equal sign (=) followed by the user-defined text for the header. The fields are written in the order specified on the command-line in column format. The field widths are specified by the system to be at least as wide as the default or user-defined header text. If the header text is null (for example, `-o user=` is specified), the field width is at least as wide as the default header text. If all header fields are null, no header line is written.

Item**Description**

Following is the mapping between the default headers and various field specifiers. Every entry in the Default Header column can be overridden by appending an equal sign (=) to the corresponding entry in the Field specifier followed by the user-defined text for the header.

Default Header	Field specifier
ARGS	"args"
COMM	"comm"
COMM	"command"
COMM	"ucomm"
F_ETIME	"etime"
GROUP	"group"
GROUP	"gname"
GID	"gid"
NICE	"nice"
PRI	"pri"
NICE	"ni"
PCPU	"pcpu"
PMEM	"pmem"
PGID	"pgid"
PID	"pid"
PPID	"ppid"
RGROUP	"rgroup"
RGROUP	"rgname"
RGID	"rgid"
RUSER	"ruser"
RUSER	"runame"
RUID	"ruid"
TIME	"time"
TIME	"cputime"
TTY	"tty"
TTY	"tt"
TTY	"tname"
TTY	"longtname"
USER	"user"
USER	"uname"
UID	"uid"
LOGNAME	"logname"
STIME	"start"
VSZ	"vsz"
VSZ	"vsize"
RSS	"rssize"
FLAG	"flag"
STATUS	"status"
CP	"cp"
PAGEIN	"pagein"
WCHAN	"wchan"
NWCHAN	"nwchan"
ST	"st"
TID	"tid"
SCOUNT	"scount"
BIND	"bnd"
SCHED	"sched"
THCOUNT	"thcount"
TAG	"tag"
CLASS	"class"
TCPU	"tcpu"
TDISKIO	"tdiskio"
TCTIME	"tctime"
MACLAB	"mac"

-p Plist

Displays only information about processes with the process numbers specified for the *Plist* variable. The *Plist* variable is either a comma-separated list of process ID numbers or a list of process ID numbers enclosed in double quotation marks (" ") and separated from one another by a comma or by one or more spaces, or both.

-P

Displays the Project name, Project origin, and subproject identifier for the project. If the stick bit is set for the process, the project name is preceded by an asterisk (*) character. The `Project origin` field designates the currently loaded project repository (LOCAL or LDAP).

-t Tlist

Displays only information about processes associated with the controlling ttys listed in the *Tlist* variable. The *Tlist* variable is either a comma-separated list of tty identifiers or a list of tty identifiers enclosed in double quotation marks (" ") and separated from one another by a comma or by one or more spaces, or both.

-T pid

Displays the process hierarchy rooted at a given pid in a tree format using ASCII art. This flag can be used in combination with the **-f**, **-F**, **-o**, and **-l** flags.

-u Ulist

This flag is equivalent to the **-U Ulist** flag. The **-u** flag only applies to the current operating environment unless the **-@** flag is also specified. If the **-@** flag is used to specify a workload partition other than the current operating environment and the **-u** flag is specified, the list of user IDs must be numeric.

-U Ulist

Displays only information about processes with the user ID numbers or login names specified for the *Ulist* variable. The *Ulist* variable is either a comma-separated list of user IDs or a list of user IDs enclosed in double quotation marks (" ") and separated from one another by a comma and one or more spaces. The **-U** flag only applies to the current operating environment unless the **-@** flag is also specified. If the **-@** flag is used to specify a workload partition other than the current operating environment and the **-U** flag is specified, the list of user IDs must be numeric. In the listing, the **ps** command displays the numerical user ID unless the **-f** flag is used; then the command displays the login name. This flag is equivalent to the **-u Ulist** flag. See also the **u** flag.

Item	Description
-X	Prints all available characters of each user/group name instead of truncating to the first eight characters.
-Z	Displays the page size settings of processes.
	DPGSZ Indicates the data page size of a process.
	SHMPGSZ Indicates the shared memory page size the process allocates.
	SPGSZ Indicates the stack page size of a process.
	TPGSZ Indicates the text page size of a process.
-@ [<i>WparName</i>]	Displays the process information that is associated with the workload partition <i>WparName</i> . If you do not specify the <i>WparName</i> parameter, the process information for all workload partitions is displayed. Workload partition information is displayed for all processes. You must specify other flags to the ps command to determine which process information to be displayed.

Options

The following options are not preceded by a minus sign (-):

Item	Description
a	Displays information about all processes with terminals (ordinarily only the own processes of the user are displayed).
c	Displays the command name, as stored internally in the system for purposes of accounting, rather than the command parameters, which are kept in the process address space.
e	Displays the environment as well as the parameters to the command, up to a limit of 80 characters.
ew	Wraps the display from the e flag one extra line.
eww	Wraps the display from the e flag and displays the ENV list until the flag reaches the LINE_MAX value.
ewww	Wraps the display from the e flag and displays the ENV list until the flag reaches the INT_MAX value.
g	Displays all processes.
l	Displays a long listing having the F, S, UID, PID, PPID, C, PRI, NI, ADDR, SZ, PSS, WCHAN, TTY, TIME, and CMD fields.
n	Displays numerical output. In a long listing, the WCHAN field is printed numerically rather than symbolically. In a user listing, the USER field is replaced by a UID field.
s	Displays the size (SSIZ) of the kernel stack of each process (for use by system maintainers) in the basic output format. This value is always 0 (zero) for a multi-threaded process.
t tty	Displays processes whose controlling tty is the value of the <i>tty</i> variable, which should be specified as printed by the ps command; that is, 0 for terminal /dev/tty/0 , 1 for /dev/lft0 , and pts/2 for /dev/pts/2 .
u	Displays user-oriented output. This includes the USER, PID, %CPU, %MEM, SZ, RSS, TTY, STAT, STIME, TIME, and COMMAND fields.
v	Displays the PGIN, SIZE, RSS, LIM, TSIZ, TRS, %CPU, %MEM fields.
w	Specifies a wide-column format for output (132 columns rather than 80). If repeated, (for example, ww), uses arbitrarily wide output. This information is used to decide how much of long commands to print.
x	Displays processes without a controlling terminal in addition to processes with a controlling terminal.
X	Prints the full user name or group name. The name is not truncated.

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 display all processes, type:

```
ps -e -f
```

To display all processes with expanded user name, type:

```
ps -X -e -f
```

2. To list processes owned by specific users, type:

```
ps -f -l -ujim,jane,su
```

3. To list processes that are associated with the **/dev/console** and **/dev/tty1** ttys, type:

```
ps -t console, tty/1
```

4. To list processes not associated with a terminal, type:

```
ps -t -
```

5. To display a specified format with field specifiers, type:

```
ps -o ruser,pid,ppid=parent,args
```

The output is:

```
RUSER  PID      parent  COMMAND
helene  34        12      ps -o ruser,pid,ppid=parent,args
```

6. To display a specified format with field descriptors, type:

```
ps -o "< %u > %p %y : %a"
```

The output is:

```
< RUSER >      PID      TT :      COMMAND
< helene >     34      pts/3 : ps -o < %u > %p %y : %a
```

7. To display information about processes and kernel threads controlled by the current terminal, type:

```
ps -lm
```

The output is like:

```
      F S UID  PID PPID  C PRI NI ADDR  SZ WCHAN  TTY  TIME  CMD
240003 A  26 8984 7190  1  60 20 2974 312          pts/1  0:00 -ksh
      400 S  -   -   -   1  60 -   -   -          -     -   -
200005 A  26 9256 8984 15  67 20 18ed 164          pts/1  0:00 ps
      0 R  -   -   -  15  67 -   -   -          -     -   -
```

8. To display information about all processes and kernel threads, type:

```
ps -emo THREAD
```

The output is like:

```
USER  PID  PPID  TID S  C PRI SC   WCHAN  FLAG  TTY  BND  CMD
jane  1716 19292  - A 10 60  1      * 260801 pts/7  -  biod
      -   -   - 4863 S  0 60  0 599e9d8 8400  -   -  -
      -   -   - 5537 R 10 60  1 5999e18 2420  -   3  -
luke  19292 18524  - A  0 60  0 586ad84 200001 pts/7  -  -ksh
      -   -   - 7617 S  0 60  0 586ad84 400    -   -  -
luke  25864 31168  - A 11 65  0      - 200001 pts/7  -  -
      -   -   - 8993 R 11 65  0      - 0      -   -  -
```

9. To list all the 64-bit processes, type:

```
ps -M
```

10. To display the project assignment details for the processes, type:

```
ps -P
```

11. To display the page size settings of the processes, type:

```
ps -Z
```

The output is like:

PID	TTY	TIME	DPGSZ	SPGSZ	TPGSZ	SHMPGSZ	CMD
41856	pts/15	0:00	4K	4K	4K	64K	ps
84516	pts/15	0:00	4K	4K	4K	64K	ksh

Files

Item	Description
/usr/bin/ps	Contains the ps command.

Using the **ps** command in *Performance management*.

System V ps command

Syntax (System V)

```
/usr/sysv/bin/ps [ -a ] [ -A ] [ -c ] [ -d ] [ -e ] [ -f ] [ -j ] [ -l ] [ -L ] [ -P ] [ -y ] [ -g pgrplist ] [ -o format ] [ -p proclist ] [ -s sidlist ] [ -t termlist ] [ { -u | -U } uidlist ] [ -G grplist ] [ -X ]
```

Description (System V)

The **ps** command prints information about active processes. Without flags, **ps** prints information about processes associated with the controlling terminal. The output contains the process ID, terminal identifier, cumulative runtime, and the command name. The information displayed with flags varies accordingly.

Output

Depending on the flags used with the **ps** command, column headings vary for the information displayed. The headings are defined in the following list (flags that cause these headings to appear are shown in parentheses):

F (-l)

Flags (hexadecimal and additive) associated with the process, or the thread if the **-L** option is specified. Some of the more important F field flags (hexadecimal and additive) associated with processes and threads are shown below:

F Field Table		
Flags	Hexadecimal Value	Definition
SLOAD	0x00000001	Indicates that the process is operating in core memory.
SNOWAP	0x00000002	Indicates that the process cannot be swapped out.
STRC	0x00000008	Indicates that the process is being traced.
SKPROC	0x00000200	Indicates a Kernel process.
SEXIT	0x00010000	Indicates that the process is exiting.
SEXECED	0x00200000	Indicates that process has been run.
SEXECING	0x01000000	Indicates that the process is execing (performing an exec).
TKTHREAD	0x00001000	Indicates that the thread is a kernel only thread.

Note: You can see the definitions of all process and thread flags by referring to the **p_flags** and **t_flags** fields in the **/usr/include/sys/proc.h** and **/usr/include/sys/thread.h** files respectively.

S (-l)

The state of the process or kernel thread :

For processes:

O

Nonexistent

A

Active

W

Swapped

I

Idle

Z

Canceled

T

Stopped

For kernel threads:

O

Nonexistent

R

Running

S

Sleeping

W

Swapped

Z

Canceled

T

Stopped

UID (-f, -l)

The user ID number of the process (the login name is printed under the -f option).

PID (all)

The process ID of the process.

PPID (-f, -l)

The process ID of the parent process.

CLS (-c)

Scheduling class for the process. Printed only when the -c flag is used.

NI (-l)

The nice value of the process used in calculating priority for the **sched_other** policy.

PRI (-c, -l)

The priority of the process or kernel thread. Higher numbers mean lower priority.

ADDR (-l)

Contains the segment number of the process stack, if normal; if a kernel process, the address of the preprocess data area.

SZ (-l)

The size in pages of the core image of the process.

WCHAN(-l)

The event for which the process or kernel thread is waiting or sleeping. For a kernel thread, this field is blank if the kernel thread is running. For a process, the wait channel is defined as the wait channel of the sleeping kernel thread if only one kernel thread is sleeping; otherwise a star is displayed.

STIME (-f,-u)

The starting time of the process. The **LANG** environment variables control the appearance of this field.

TTY (all)

The controlling terminal for the process:

- The process is not associated with a terminal.
- ? Unknown

TIME (all)

The total runtime for the process. The time is displayed in the format of *mm:ss* or *mmm:ss* if the runtime reaches 100 minutes, which is different from the displayed format if you use the **-o time** flag.

LTIME (-L)

The runtime for an individual LWP.

CMD (all)

Contains the command name. The full command name and its parameters are displayed with the **-f** flag.

LWP (-L)

The tid of the kernel thread.

NLWP(-Lf)

The number of kernel threads in the process.

PSR (-P)

The logical processor number of the processor to which the kernel thread is bound (if any). For a process, this field is shown if all its threads are bound to the same processor.

RSS (-ly)

The real memory (resident set) size of the process (in 1 KB units).

Format

The following list describes the field specifiers recognized by the system. These field specifiers can be used with the **-o** flag to specify the format for the output of the **ps** command.

The field specifiers recognized by the system are:

addr

Indicates the segment number of the process stack, if normal; if a kernel process, the address of the preprocess data area. The default header for this field is **ADDR**.

args

Indicates the full command name being executed. All command-line arguments are included, though truncation may occur. The default header for this field is **COMMAND**.

c

CPU utilization of process or thread, incremented each time the system clock ticks and the process or thread is found to be running. The value is decayed by the scheduler by dividing it by 2 once per second. For the **sched_other** policy, CPU utilization is used in determining process scheduling priority. Large values indicate a CPU intensive process and result in lower process priority whereas small values indicate an I/O intensive process and result in a more favorable priority. The default header for this field is **C**.

class

Indicates the scheduling policy for a kernel thread. The policies are `sched_other`, `sched_fifo` and `sched_rr`. The default header for this field is **CLS**.

comm

Indicates the short name of the command being executed. Command-line arguments are not included. The default header for this field is **COMMAND**.

etime

Indicates the elapsed time since the process started. The elapsed time is displayed in the format

```
[[ dd -] hh: ]mm :ss
```

where *dd* specifies the number of days, *hh* specifies the number of hours, *mm* specifies the number of minutes, and *ss* specifies the number of seconds.

The default header for this field is **ELAPSED**.

f

Indicates flags (hexadecimal and additive) associated with the process. The default header for this field is **COMMAND**.

fname

Indicates the first 8 bytes of the base name of the process's executable file. The default header for this field is **COMMAND**.

gid

Indicates the effective group ID number of the process as a decimal integer. The default header for this field is **GID**. The login name is printed under the **-f** option.

group

Indicates the effective group ID of the process. The textual group ID is displayed. If the textual group ID cannot be obtained, a decimal representation is used. The default header for this field is **GROUP**.

lwp

Indicates the thread ID of the kernel thread. The default header for this field is **TID**.

nice

Indicates the decimal value of the process nice value. The default header for this field is **NI**.

nlwp

Indicates the number of kernel threads owned by the process. The default header for this field is **THCNT**.

pcpu

Indicates the ratio of CPU time used to CPU time available, expressed as a percentage. The default header for this field is **%CPU**.

pgid

Indicates the decimal value of the process group ID. The default header for this field is **PGID**.

pid

Indicates the decimal value of the process ID. The default header for this field is **PID**.

pmem

Indicates the percentage of real memory used by this process. The default header for this field is **%MEM**.

ppid

Indicates the decimal value of the parent process ID. The default header for this field is **PPID**.

pri

Indicates the priority of the process or kernel thread ; higher numbers mean lower priority. The default header for this field is **PRI**.

psr

Indicates the logical processor number of the processor to which the kernel thread is bound (if any). The default header for this field is **PSR**.

rgid

Indicates the real group ID number of the process as a decimal integer. The default header for this field is **RGID**.

rgroup

Indicates the real group ID of the process. The textual group ID is displayed. If the textual group ID cannot be obtained, a decimal representation is used. The default header for this field is **RGROUP**.

rss

Indicates the real memory (resident set) size of the process (in 1 KB units). The default header for this field is **RSS**.

ruid

Indicates the real user ID number of the process as a decimal integer. The default header for this field is **RUID**.

ruser

Indicates the real user ID of the process. The textual user ID is displayed. If the textual user ID cannot be obtained, a decimal representation is used. The default header for this field is **RUSER**.

s

Indicates the state of the process. The default header for this field is **S**.

sid

Indicates the process ID of the session leader. The default header for this field is **SID**.

stime

Indicates the starting time of the process. The LANG environment variables control the appearance of this field. The default header for this field is **STIME**.

time

Indicates the cumulative CPU time since the process started. The time is displayed in the same format as in **etime**. The default header for this field is **TIME**.

tty

Indicates the controlling terminal name of the process. The default header for this field is **TT**.

uid

Indicates the effective user ID number of the process as a decimal integer. The default header for this field is **UID**.

user

Indicates the effective user ID of the process. The textual user ID is displayed. If the textual user ID cannot be obtained, a decimal representation is used. The default header for this field is **USER**.

vsz

Indicates, as a decimal integer, the size in kilobytes of the core image of the process. The default header for this field is **VSZ**.

wchan

Indicates the event for which the process or kernel thread is waiting or sleeping. For a kernel thread, this field is blank if the kernel thread is running. For a process, the wait channel is defined as the wait channel of the sleeping kernel thread if only one kernel thread is sleeping; otherwise a star is displayed.

The default header for this field is **WCHAN**.

Flags (System V)

Some flags accept lists as arguments. Items in a list can be either separated by commas or else enclosed in double quotes and separated by commas or spaces. Values for *proclist* and *pgprlist* must be numeric.

Item	Description
-a	Writes to standard output information about all processes, except the session leaders and processes not associated with a terminal.
-A	Writes to standard output information about all processes.
-c	Prints information in a format that reflects scheduler properties. The -c flag affects the output of the -f and -l flags, as described below.
-d	Writes to standard output information about all processes, except the session leaders.
-e	Writes to standard output information about all processes, except kernel processes.

Item	Description
-f	Generates a full listing.
-g pgrplist	Writes to standard output information only about processes that are in the process groups specified by <i>pgrplist</i> . Values for <i>pgrplist</i> must be numeric.
-G grplist	Writes to standard output information only about processes that are in the process groups specified by <i>grplist</i> . The -G flag accepts group names.
-j	Displays session ID and process group ID.
-l	Generates a long listing.
-L	Prints status of active threads within a process.
-o format	Displays information in the format specified by <i>format</i> . Multiple field specifiers can be specified for the format variable. The field specifiers that can be used with the -o flag are described above in the Format section.
-p proclst	Displays information only about processes with the process numbers specified by <i>proclst</i> . Values for <i>proclst</i> must be numeric.
-P	Displays the logical processor number of the processor to which the primary kernel thread of the process is bound (if any).
-s sidlist	Displays all processes whose session leader's IDs are specified by <i>sidlist</i> .
-t termlist	Displays information only about processes associated with the terminals specified by <i>termlist</i> .
-u uidlist	Displays information only about processes with the user ID numbers or login names specified by <i>uidlist</i> .
-U uidlist	Displays information only about processes with the user ID numbers or login names specified by <i>uidlist</i> .
-X	Prints all available characters of each user and group name instead of truncating to the first 8 characters.
-y	When combined with the -l option, changes the long listing so that it prints the "RSS" and "SZ" fields in kilobytes and does not print the "F" and "ADDR" fields.

Exit Status (System V)

This command returns the following exit values:

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

Security (System V)

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 (System V)

1. To display all processes, enter:

```
ps -e -f
```

2. To list processes owned by the user 'guest', enter:

```
ps -f -l -u guest
```

3. To list processes that are associated with the **/dev/pts/0** and **/dev/pts/1** terminals, enter:

```
ps -t pts/0,pts/1
```

4. To list processes not associated with a terminal, enter:

```
ps -t -
```

5. To display a specified format with field specifiers, enter:


```
ps -o ruser,pid,ppid,args
```

6. To display information about all kernel threads in a process, enter:

```
ps -L
```

7. To display session ID and process group IDs of all the processes, enter:

```
ps -jA
```

8. To display the scheduling class and priority of processes, enter:

```
ps -c -l
```

9. To display information about kernel threads and the number of kernel threads in a process, enter:

```
ps -L -f
```

10. To display the processor to which the process or kernel thread is bound to, enter:

```
ps -P
```

11. To print an ASCII art for a given process (inetd in the example below), enter :

```
ps -T 14220
```

Output will look like the following:

```
  PID   TTY   TIME CMD
 14220   -   0:00 inetd
 16948   -   0:00 | \--telnetd
 32542 pts/4 0:00 | | \--ksh
 26504   -   0:00 | \--telnetd
 41272 pts/5 0:00 | | \--ksh
 26908 pts/5 0:00 | | \--vi
 28602   -   0:00 | \--telnetd
 24830 pts/0 0:00 | | \--ksh
 676416 pts/0 0:00 | | \--ksh
 29984   -   0:00 | \--telnetd
 38546 pts/6 0:00 | | \--ksh
 32126   -   0:00 | \--telnetd
 11162 pts/7 0:00 | | \--ksh
 34466   -   0:00 | \--rpc.ttdbserver
 35750   -   0:00 | \--telnetd
 23612 pts/3 0:00 | | \--ksh
 36294   -   0:00 | \--telnetd
 38096 pts/8 0:00 | | \--ksh
 39740   -   0:00 | \--telnetd
 42226 pts/9 0:01 | | \--ksh
 40632   -   0:00 | \--telnetd
 40232 pts/2 0:00 | | \--ksh
 32910 pts/2 0:00 | | \--dbx
 987990 pts/2 0:00 | | \--a.out
 40722   -   0:00 | \--telnetd
 16792 pts/10 0:00 | | \--ksh
 37886 pts/10 0:00 | | \--ps
 105716   -   0:00 | \--telnetd
 29508 pts/1 0:00 | | \--ksh
 39478 pts/1 0:00 | | \--ksh
 38392 pts/1 0:00 | | \--vi
```

12. To print information about all processes rooted at a given pid, enter:

```
ps -fL 14220
```

Output will look like the following:

```
  UID   PID  PPID   C   STIME   TTY   TIME CMD
  root 14220  8676   0   Apr 07   -   0:00 /usr/sbin/inetd
  root 16948 14220   0   Apr 06   -   0:00 telnetd -a
  root 23612 35750   0   Apr 10 pts/3 0:00 -ksh
  root 24830 28602   1 18:30:56 pts/0 0:00 -ksh
  root 28602 14220   0 18:30:55   -   0:00 telnetd -a
```

```

root 32542 16948 0 Apr 06 pts/4 0:00 -ksh
root 34466 14220 0 Apr 10 - 0:00 rpc.ttdbserver 100083 1
root 35750 14220 0 Apr 10 - 0:00 telnetd -a
root 40228 24830 8 18:36:01 pts/0 0:00 ps -fl 14220

```

13. To display all processes with expanded user name, type:

```
ps -X -e -f
```

14. To display the scheduling policies of a thread, enter the following command:

```

#ps -m -o THREAD,sched
USER      PID      PPID      TID ST  CP  PRI SC  WCHAN  F      TT BND COMMAND      SCH
suresana 1609830 4227284   -  A   16  68  1  - 200001 pts/144 - ps -m
           -o THREAD sched  0
-         -         - 6381739 R   16  68  1  -400000 - - - 0
suresana 4227284 4239476   -  A   1  60  1  -200801 pts/144 - bash  0
-         -         - 4177981 S   1  60  1  -410400 - - - 0
suresana 4239476 921694   -  A   0  60  1  -240001 pts/144 - -ksh  0
-         -         - 5554385 S   0  60  1  -10400  - - - 0

```

Files (System V)

Item	Description
/usr/sysv/bin/ps	Contains the System V R4 ps command.
/etc/passwd	Contains the user ID information.
/dev/pty*	Indicates terminal (PTY) names.
/dev/tty*	Indicates terminal (TTY) names.

ps4014 Command

Purpose

Converts a Tektronix 4014 file to PostScript format.

Syntax

```
ps4014 [ -m ] [ -C ] [ -N ] [ -R ] [ -sWidth,Height ] [ -lLeft,Bottom ] [ -SWidth ] [ -pOutFile ] [ File ]
```

Description

The **ps4014** command reads in a Tektronix 4014 format file and converts it to PostScript format for printing on a PostScript printer. If no file is specified, the standard input is used. The resulting PostScript file can be directed to standard output or to a named file.

Note: By default, the 4014 image is scaled to occupy nearly the entire page in a landscape orientation.

Flags

Note: The **-m**, **-C**, and **-N** flags specify values for 4014 hardware options that affect the interpretation of 4014 commands.

Item	Description
-C	Causes a carriage return to move the pen position to the left margin but not down to the next line. By default, a carriage return command moves the pen down to the next line and over to the left margin.

Item	Description
-l <i>Left,Bottom</i>	Specifies the location on the printed page of the bottom left corner of the converted raster image. The values specified by the <i>Left</i> and <i>Bottom</i> parameters are the distances (in inches) from the bottom left corner of the printed page to the bottom left corner of the image.
-m	Enables the "Margin 2" mode for the 4014.
-N	Causes line feed to move the pen position down to the next line but not to the left margin. By default, a line feed command moves the pen down to the next line and over to the left margin.
-p <i>OutFile</i>	Causes the PostScript file to be written to the file specified by the <i>OutFile</i> parameter rather than the standard output.
-R	Rotates the image 90 degrees on the page for portrait orientation. The default is landscape orientation.
-s <i>Width,Height</i>	Specifies the size of the converted raster image on the printed page. The <i>Width</i> and <i>Height</i> parameters specify the dimensions (in inches) of the resulting image on the printed page.
-S <i>Width</i>	Allows you to scale the image without distorting its shape. The <i>Width</i> parameter specifies the width, in inches, of the resulting image on the printed page. The height of the image is computed to maintain the same ratio of height to width on the output image as on the input raster-format file.

International Character Support

See the [html](#)

ps630 Command

Purpose

Converts Diablo 630 print files to PostScript format.

Syntax

```
ps630 [ -fBodyfont ] [ -pFile ] [ -sPitch ] [ -FBoldfont ] [ File ... ]
```

Description

The **ps630** command converts Diablo 630 format print files to PostScript format for printing on a PostScript printer. If no *File* variable is specified, the **ps630** command reads from standard input. By default, the PostScript file is sent to the standard output.

The **ps630** command can convert **nroff** files generated with the **-Txerox** flag. Typewheel emulation information can be specified as options. Font specifications (for bold and regular) are PostScript font names (such as Times-Roman, Times-Bold, Courier-Bold, Courier-BoldOblique). You can select 10, 12, or 15 characters per inch.

Some applications produce bold type by double-striking a character. This type of bolding is not translated into PostScript format. Only the bold effect produced by issuing the proper Diablo command sequence (Esc-O) results in bold characters.

The output of the **ps630** command cannot be page-reversed. Times-Roman and Helvetica are narrow fonts that may look squeezed if no adjustment to the page width is made by the application.

The following Diablo 630 commands are not supported:

- Print suppression

- HY-Plot
- Extended character set
- Downloading print wheel information or program mode
- Page lengths other than 11 inches
- Paper feeder control
- Hammer energy control
- Remote diagnostic
- Backward printing control.

Note: The Diablo 630 command for reverse printing is supported.

Flags

Item	Description
-fBodyfont	Sets the font to be used for normal printing. The default is Courier.
-pFile	Causes the PostScript file to be written to the file specified by the <i>File</i> parameter rather than to the standard output.
-sPitch	Selects type size for printing (both the regular and bold fonts are scaled to this size). Pitch is in characters per inch and must be one of 10, 12, or 15. The default is 12.
-FBoldfont	Sets the font to be used for bold type. The default is Courier-Bold.

International Character Support

See the [html](#)

psc or psdit Command

Purpose

Converts **troff** intermediate format to PostScript format.

Syntax

```
{ psc | psdit } [ -f1 CodeSet:Font ] [ -F FontDirectory ] [ -M MediaName ] [ -p Prologue ] [ -o List ] [ File ]
```

Description

The **psc** and **psdit** commands translate a file created by device-independent **troff** to PostScript format for printing with a PostScript printer. If no file is specified, the standard input is used. The PostScript file is sent to the standard output.

Note: The input for the **psc** and **psdit** commands should be prepared with the corresponding **-Tpsc** option, such as the **troff** or **pic** command.

The **psc** and **psdit** commands can handle extended characters created by modifying the printer code field in the font file (`/usr/lib/font/devpsc/R`). The modified field contains a string surrounded by double quotation marks. The string contains a `\b` (backslash b) followed by a sequence of characters from the standard font that is composed into a new character by overstriking.

The **psc** and **psdit** commands allow users to cause the **troff** command to include arbitrary PostScript code in the generated PostScript file. The **psc** and **psdit** commands recognize the undefined **%** (percent) command in the **troff** intermediate file format to signal the start of raw PostScript code to be placed as is in the output file. Everything between (but not including) the **%** (percent sign) and a line containing a **.** (period) will be placed in the generated PostScript output.

This PostScript output is not insulated from the **troff** command coordinate system or the state of the generated PostScript output. However, two functions are defined in the prologue so that users can insulate themselves if so desired. The **PB** (picture begin) function performs a PostScript save operation, translates the PostScript coordinate system to **troff**'s idea of the current position on the page, and changes the scale and orientation of the coordinate system axes to the standard PostScript 72 units per inch. The **PE** (picture end) macro ends this protected environment.

Several methods can be used to incorporate such included PostScript code into the **troff** intermediate file. For example, the **.sy**, **\!**, and **.cf** subcommands of the **troff** command use the following example to include the PostScript language description of a completely separate, printable document. In this example, the **showpage** operator is redefined to include `mypic.ps` as an illustration:

```
standard troff input
\&
.fl
\!%PB
\!/showpage{}def
.fl
.sy cat mypic.ps
\!PE
\!.
more standard troff input
```

Information containing various media sizes for the **psdit** command and the **enscript** command are contained in the file `/usr/lib/ps/MediaSizes`.

The information required for each entry in the **MediaSizes** file can be obtained from the **PostScript Printer Description**, or **PPD**, file that matches the PostScript printer used with TranScript. The **PPD** files are available from Adobe Systems Incorporated. The measurements extracted from the **PPD** files are in points. A printer's point is 1/72 of an inch.

Any line in the **MediaSizes** file beginning with an ASCII ***** (asterisk) is ignored when matching media size names provided on the command line to the **enscript** command and the **psdit** command.

Each entry in the **MediaSizes** file contains either eight or nine fields. The first eight fields are required for all entries. The ninth field is optional. Fields are separated by white space. The fields for each entry are as follows:

Field Name	Description
EntryName	Character string to match against a media name provided with the -M option with the enscript command or the psdit command.
MediaWidth	Media width in points.
MediaDepth	Media depth in points.
ImageableLLX	Imageable lower left-hand corner x coordinate in points.
ImageableLLY	Imageable lower left-hand corner y coordinate in points.
ImageableURX	Imageable upper right-hand corner x coordinate in points.
ImageableURY	Imageable upper right-hand corner y coordinate in points.
PageRegionName	PostScript sequence for the particular printer to identify the size of the imageable area.
PaperTrayName	PostScript sequence for the particular printer to select a particular paper/media tray. This field is optional.

Note: The sequence can be multiple PostScript operators or words for both the **PageRegionName** field and the **PaperTrayName** field. To specify such a sequence, use the ASCII `"` (double quotation mark character) to delimit the entire sequence.

The following are examples of field entries in the **MediaSizes** file:

Name	Entries
Letter	Width 612 Depth 792 llx 18 lly 17 urx 597 ury 776 Page- Region- Name Letter Page- Tray- Name
Legal	Width 612 Depth 1008 llx 18 lly 17 urx 597 ury 992 Page- Region- Name Legal Page- Tray- Name

Flags

-f1 *CodeSet:Font*

Item	Description
-F <i>FontDirectory</i>	Takes font information from <i>FontDirectory</i> instead of the default.
-M <i>MediaName</i>	Specifies a media name to use to determine the amount of imageable area on the paper. The name provided is matched against entries in the MediaSizes file. For instance, -M legal would request a legal size of paper as the imageable area. If this option is not used, the default size is letter size, which is 8.5 inches wide by 11.0 inches deep.
-p <i>Prologue</i>	Uses the contents of <i>Prologue</i> instead of the default PostScript prologue.

Item	Description
-oList	Prints pages whose numbers are given in the list separated by commas. The list contains single numbers and ranges in the format <i>N1-N2</i> , where <i>N1</i> and <i>N2</i> represent page numbers. A missing <i>N1</i> means the range begins with the lowest-numbered page; a missing <i>N2</i> means the range ends with the highest-numbered page.

Examples

The following statements are equivalent:

```
pic -Tpsc File | troff -Tpsc | psc
pic -Tpsc File | troff -Tpsc | psdit
```

Environment Variables

Item	Description
PSLIBDIR	Path name of a directory to use instead of the /usr/lib/ps file for the psc and psdit command prologue.
TRANSCRIPT	Absolute path name of a file to use instead of /usr/lib/ps/transcript.conf for the MBCS handling.

Files

Item	Description
/usr/lib/font/devpsc/*	Contains the troff default description files for a PostScript virtual device.
/usr/lib/ps/psdit.pro	Contains the default PostScript prologue.
/usr/lib/ps/MediaSizes	Contains the default file used for media sizes.
/usr/lib/ps/transcript.conf	Contains the default value used for PostScript codeset and font name.

pshare Command

Purpose

Enables or reports the availability of shared login ports.

Syntax

```
pshare [ -a ] [ Device ]
```

Description

The **pshare** command enables shared ports. Shared ports are bidirectional. If you do not specify a *Device* parameter, the **pshare** command reports the names of all currently enabled shared ports. To enable a shared port, the **getty** command attempts to create a **lock** file in the **/etc/locks** directory that contains the ASCII process ID of the process. If another process is already using the port, the **getty** command waits until the port is available and tries again. The system enables a port by updating an entry in the **/etc/inittab** file and then sending a signal to the **init** process. After receiving the signal and reading the updated status entry, the process takes the appropriate action.

Use the *Device* parameter to specify the ports to be enabled. Permitted values include:

- Full device name, such as the **/dev/tty1** device
- Simple device name, such as the **tty1** device
- A number (for example, 1 to indicate the **/dev/tty1** device)

Note: You must have root user authority to run this command.

Flags

Item Description

-a Enables all ports as shared.

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 enable the workstation attached to the **/dev/tty2** port as a shared port, enter:

```
pshare /dev/tty2
```

Item	Description
/etc/inittab	Controls system initialization.

Files

Item	Description
/etc/locks	Contains lock files for the pshare and pdelay commands.
/usr/sbin/pshare	Contains the pshare command.

psplot Command

Purpose

Converts files in plot format to PostScript format.

Syntax

```
psplot [ -g Prologue ] [ File... ]
```

Description

The **psplot** command reads files in plot format and converts them to PostScript format on the standard output. If no files are specified, the standard input is used. The conversion is almost one-to-one, with one PostScript function call for each plot primitive. You can modify the behavior of the file by changing the definitions of the PostScript functions in the prologue.

Flags

Item	Description
<code>-gPrologue</code>	Uses the contents of the <i>Prologue</i> file instead of the default PostScript prologue. If this flag is not specified, the default prologue file is used.

International Character Support

The html

psrasc Command

Purpose

Collects centralized RAS data.

Syntax

psrasc type [-d] [-n *number*] -o *outputFile* logSpace/logStream

Description

The **psrasc** command extracts the Reliability/Availability/Serviceability (RAS) data log records centralized on a PowerHA pureScale® log stream and builds a file in the RAS data AIX format. The PowerHA pureScale service name is **CentralizedLogService**. Binding information for that service name must be setup before using the **psrasc** command.

RAS data types

When the specified type is **syslog**, the log records contain system log messages, including message initiator hostname. The format of the generated file is similar to the system log destination files. When the specified type is **errlog**, the log records contain error log entries. The generated file is an error log file that can be later exploited by the **errpt** command.

Flags

Item	Description
type	Specifies the type of RAS data contained in the log records. This must be the first parameter. Supported RAS data types are: syslog and errlog . From this type, depends the format of the output file.
-d	Specifies that the collected log record are deleted.
-n <i>number</i>	Specifies the number of log records to collect. Oldest log records are collected. When this parameter is not specified, all the log records are collected.
-o <i>outputFile</i>	Specifies the relative or absolute pathname of the output file. If the file already exists, it is overwritten.
log_space/ log_stream	Specifies the fullname of the log stream from which system log messages are collected. Fullname is made of the parent log space name and the log stream name separated by a / (slash).

Exit Status

This command returns the following exit values:

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

Examples

1. To collect log records of the log stream named **CentralizedRAS/Syslog** into the **syslog.out** file on the

PowerHA pureScale server identified by the **CentralizedLogService** service name, enter:

```
psrasc syslog -o syslog.out CentralizedRAS/Syslog
```

2. To collect the 100 oldest log records of the log stream named **CentralizedRAS/Syslog** into the **/var/adm/ras/cluster_syslog** file on the PowerHA pureScale server

identified by the **CentralizedLogService** service name and delete them, enter:

```
psrasc syslog -d -n 100 -o /var/adm/ras/cluster_syslog CentralizedRAS/Syslog
```

3. To collect log records of the log stream named **CentralizedRAS/Errlog** into the file **centralizedRAS_errlog** on the PowerHA pureScale server identified by the **CentralizedLogService** service name, enter:

```
psrasc errlog -o centralizedRAS_errlog CentralizedRAS/Errlog
```

4. To collect and delete the 100 oldest log records of the log stream named **CentralizedRAS/Errlog** into the **/var/adm/ras/cluster_errlog** error log file on

the PowerHA pureScale server identified by the **centralizedRAS_error** service, enter:

```
psrasc errlog -d -n 100 -o /var/adm/ras/cluster_errlog CentralizedRAS/Errlog
```

psrev Command

Purpose

Reverses the page order of a PostScript file and selects a page range for printing.

Syntax

```
psrev [ -R ] [ -sPagespec,... ] [ File ]
```

Description

The **psrev** command reverses the page order of the file specified by the *File* variable and prints the pages specified by the *Pagespec* parameter. The file must conform to PostScript file structuring conventions. If no value for the *File* is specified, the **psrev** command reads from standard input. The **psrev** command writes the resulting file to the standard output.

Flags

Item	Description
-R	Does not reverse the page order (but subsets the pages if specified).
-sPagespec	Specifies a range (or several ranges) of pages to be printed. The <i>Pagespec</i> parameter is a string with no spaces. The <i>Pagespec</i> parameter can be a single page number or a range of the form <i>N-M</i> , which prints pages <i>N</i> through <i>M</i> . <i>-N</i> prints from the beginning of the document to page <i>N</i> . <i>M-</i> prints from page <i>M</i> to the end of the document.

Examples

The following are examples of using the **psrev** command showing page ranges and an individual page in nonreversed order:

```
psrev -R -s2-4,6
```

```
psrev -R -s2-4,6-8
```

Files

Item	Description
<code>/var/tmp/RV*</code>	Contains the temporary file if the input is a pipe.

psroff Command

Purpose

Converts files from **troff** format to PostScript format.

Syntax

```
psroff [ -t ] [ -dQueue ] [ -nNumber ] [ -tTitle ] [ -DFontDirectory ] [ -FFontFamily ] [ -PFlag ] [ troffFlags ]  
[ File ... ]
```

Description

The **psroff** command is a shell script that runs the **troff** command in an environment to produce output on a PostScript printer. It uses the **psdit** command to convert **troff** intermediate output to PostScript format, and spools this output for printing. If no files are specified, the standard input is used.

To include arbitrary PostScript language commands or files in a **troff** document, see the **psdit** command.

PostScript Font Information

The PostScript Fonts for Transcript table shows the fonts available for the TranScript commands. The fonts are available by long name when using the **enscript** command, and by short name when using the **psroff** or **troff** commands. The following table shows the **psroff** commands (short names) used to declare a default set of fonts. The alphabetic characters are case-sensitive:

PostScript Fonts for Transcript	
Long Name (Short Name)	Font Family
AvantGarde-Book (ag)	AvantGarde
AvantGarde-Demi (Ag)	AvantGarde
AvantGarde-DemiOblique (AG)	AvantGarde
AvantGarde-BookOblique (aG)	AvantGarde
Bookman-Demi (Bo)	Bookman
Bookman-DemiItalic (BO)	Bookman
Bookman-Light (bo)	Bookman
Bookman-LightItalic (bO)	Bookman
Courier (C)	Courier

PostScript Fonts for Transcript (<i>continued</i>)	
Long Name (Short Name)	Font Family
Courier-Bold (CB)	Courier
Courier-BoldOblique (CO)	Courier
Courier-Oblique (CO)	Courier
Garamond-Bold (Ga)	Garamond
Garamond-BoldItalic (GA)	Garamond
Garamond-Light (ga)	Garamond
Garamond-LightItalic (gA)	Garamond
Helvetica (H)	Helvetica
Helvetica-Bold (HB)	Helvetica
Helvetica-Oblique (HO)	Helvetica
Helvetica-BoldOblique (HD)	Helvetica
Helvetica-Narrow (hn)	Helvetica
Helvetica-Narrow-Bold (Hn)	Helvetica
Helvetica-Narrow-BoldOblique (HN)	Helvetica
Helvetica-Narrow-Oblique (hN)	Helvetica
LubalinGraph-Book (lu)	Lubalin
LubalinGraph-BookOblique (LU)	Lubalin
LubalinGraph-Demi (Lu)	Lubalin
LubalinGraph-DemiOblique (LU)	Lubalin

Item	Description
NewCenturySchlbk (NC)	NewCentury
NewCenturySchlbk-Bold (Nc)	NewCentury
NewCenturySchlbk-Italic (nC)	NewCentury
NewCenturySchlbk-Roman (nc)	NewCentury
Optima (op)	Optima
Optima-Bold (Op)	Optima
Optima-BoldOblique (OP)	Optima
Optima-Oblique (oP)	Optima
Palatino-Bold (PB)	Palatino
Palatino-BoldItalic (PX)	Palatino
Palatino-Italic (PI)	Palatino
Palatino-Roman (PA)	Palatino
Souvenir-Demi (Sv)	Souvenir
Souvenir-DemiItalic (SV)	Souvenir

Item	Description
Souvenir-Light (sv)	Souvenir
Souvenir-LightItalic (sV)	Souvenir
Times-Bold (TB)	Times
Times-BoldItalic (TD)	Times
Times-Italic (TI)	Times
Times-Roman (TR)	Times
Symbol (S)	(none)
ZapfChancery-MediumItalic (ZC)	Zapf
ZapfDingbats	(none)

Flags

Item	Description
-D <i>FontDirectory</i>	Finds font family directories in the specified font directory, rather than the standard font directory, which was configured in the installation procedure. It may be necessary to use both this flag and the -F flag to imitate the -F flag in the troff command.
-d <i>Queue</i>	Causes the output to be queued to the queue specified by the <i>Queue</i> parameter. If the -d flag is not used, the psroff command queues output on the default queue, the first queue known to the qdaemon . This flag is recognized by the spooler print.
-F <i>FontFamily</i>	Uses the specified font family for the R/I/B/BI fonts, rather than the Times default family. The Times, Courier, and Helvetica font families are defined at your site, and others are available as well. Ensure that the printer you use contains the font family you pick. This flag overrides the troff command -F flag. If you want to use the troff command -F flag, you should run the troff command directly or use the -D flag instead.
-n <i>Number</i>	Causes the number of output copies specified by the <i>Number</i> parameter to be produced. The default is one. This flag is recognized by the spooler print.
-P <i>Flag</i>	Passes the <i>Flag</i> parameter to the spooler. This flag is useful when a conflict exists between a spooler flag and a flag with the psroff command.
-t	Sends the PostScript output to the standard output, rather than spooling it to a printer. This flag overrides the troff command -t flag. If you want the troff command -t flag, you should run the troff command directly.
-t <i>Title</i>	Sets the job name for use on the first banner page. The default is to use the name of the first input file. This flag is recognized by the spooler print.

Parameters

Item	Description
<i>troffFlags</i>	Specifies standard flags available with the troff command.
<i>File</i>	Specifies the troff intermediate output file. The default is the standard input.

Files

Item	Description
<code>/usr/share/lib/tmac/tmac.*</code>	Contains the standard macro files.
<code>/usr/lib/font/devpsc/*</code>	Contains the troff description files for PostScript virtual device.
<code>/usr/lib/ps/*.afm</code>	Contains Adobe Font Metrics (AFM) files for use with the enscript command.
<code>/usr/lib/ps/font.map</code>	Contains the list of font names with their abbreviations.
<code>/usr/lib/ps/ditroff.font</code>	Contains font family files for the troff command.

pstart Command

Purpose

Enables or reports the availability of login ports (normal, shared, and delayed).

Syntax

```
pstart [ -a ] [ Device ]
```

Description

The **pstart** command enables all ports (normal, shared, and delayed) listed in the `/etc/inittab` file. The system enables a port by updating an entry in the `/etc/inittab` file and then sending a signal to the **init** process. When the **init** process receives the signal and reads the updated status entry, it takes the appropriate action.

Use the *Device* parameter to specify the ports to be enabled. Permitted values include:

- A full device name, such as the `/dev/tty1` device
- A simple device name, such as the `tty1` device
- A number (for example, 1 to indicate the `/dev/tty1` device)

If you do not specify a *Device* parameter, the **pstart** command reports the names of all enabled ports and whether they are currently enabled as normal, shared, or delayed.

Note: You must have root user authority to run this command.

Flag

Item	Description
------	-------------

<code>-a</code>	Enables all ports (normal, shared, and delayed ports).
-----------------	--

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 display the names of all ports (normal, shared, and delayed) currently enabled and how they are enabled, enter:

```
pstart
```

2. To enable all normal, shared, and delayed ports listed in the **/etc/inittab** file, enter:

```
pstart -a
```

Files

Item	Description
/etc/locks	Contains lock files for the pshare and pdelay commands.
/usr/sbin/pstart	Contains the pstart command file.

pstat Command

Purpose

Interprets the contents of the various system tables and writes it to standard output.

Syntax

```
pstat [ -a ] [ -A ] [ -f ] [ -i ] [ -p ] [ -P ] [ -s ] [ -S ] [ -t ] [ -u ProcSlot ] [ -T ] [ -U ThreadSlot ]  
[ [ KernelFile ] CoreFile ]
```

Description

The **pstat** interprets the contents of the various system tables and writes it to standard output. You must have root user or **system** group authority to run the **pstat** command.

Flags

Item	Description
-a	Displays entries in the process table.
-A	Displays all entries in the kernel thread table.
-f	Displays the file table.
-i	Displays the i-node table and the i-node data block addresses.
-p	Displays the process table.
-P	Displays runnable kernel thread table entries only.
-s	Displays information about the swap or paging space usage.
-S	Displays the status of the processors.
-t	Displays the tty structures.
-u ProcSlot	Displays the user structure of the process in the designated slot of the process table. An error message is generated if you attempt to display a swapped out process.
-T	Displays the system variables. These variables are briefly described in var.h.
-U ThreadSlot	Displays the user structure of the kernel thread in the designated slot of the kernel thread table. An error message is generated if you attempt to display a swapped out kernel thread.

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 display the i-nodes of the system dump saved in the **dumpfile** core file, enter:

```
pstat -i dumpfile
```

Symbols are retrieved from the **/usr/lib/boot/unix** file.

2. To display the file table and the user structure for the process in process table slot 0 (zero) of the system currently running, enter:

```
pstat -f -u 0
```

3. To display the tty structures for a system dump, whose core file is **dumpfile** and whose kernel is the **/usr/lib/boot/unix.back** file, enter:

```
pstat -t /usr/lib/boot/unix.back dumpfile
```

4. To display all threads in the kernel thread table and the user structure of the thread in thread table slot 2, enter:

```
pstat -A -U 2
```

Files

Item	Description
/usr/sbin/pstat	Contains the pstat command.
/dev/mem	Default system-image file.
/usr/lib/boot/unix	Default kernel-image file.
/usr/include/sys/*.h	Contains header files for table and structure information.

ptpd Daemon

Purpose

Starts the Precision Time Protocol (1588-2008) daemon (**ptpd**).

Syntax

```
/usr/sbin/ptpd [-?] [-h] [-H] [-e setting] [-k] [-v] [-O] [-L] [-A] [-s] [-m] [-M] [-y] [-E] [-P] [-a] [-n] [-C] [-V] [-c file] [-R dir] [-f file] [-S file] [-d domain_number] [-u IP_address] [-r number] [-l file] [-i dev]
```

Description

The **ptpd** daemon implements the Precision Time Protocol (PTP) version 2 as defined by the IEEE 1588-2008 standard. PTP provides precise time coordination of LAN-connected computers. You must run this daemon with root authority to manipulate the system clock and use lower port numbers. The **ptpd** daemon supports IPv4 multicast, unicast, hybrid mode (mixed), and Ethernet mode operations. The

ptpd daemon can achieve and maintain submicrosecond level timing precision, even without hardware assistance.

Configure the **ptpd** daemon by using the `/etc/ptpd2.conf` configuration file (default file). The short (**-x**) and long (**--xxxxx**) flags provide basic control over the daemon operation, and provide only the basic PTP protocol settings. Other settings can be displayed by the **-h**, **-H**, and **-e** flags.

The **ptpd** daemon can be started either from System Resource Controller (SRC) or from the command line.

Use the following SRC commands to manipulate the **ptpd** 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 read the appropriate configuration file again.

lssrc

Gets the status of a subsystem, group of subsystems, or a subserver.

Note: The **ptpd** daemon does not support Network Time Protocol (NTP) and Simple Network Management Protocol (SNMP) related functions.

The **ptpd** daemon uses the following terms:

slave device

A system running the **ptpd** daemon that accepts commands from a *master* device and synchronizes its system time to match the associated boundary clock time.

master device

A device bound to a boundary clock that synchronizes time with a set of one or more PTP *slave* devices on the same network.

grandmaster device

A *master* device that has the best clock provided by the Best Master Clock algorithm. It synchronizes all of the other *master* devices (also known as the boundary clocks), which in turn update and synchronize all of the associated *slave* devices. The grandmaster clock is also known as best master clock or best clock.

Flags

Short flag	Long flag	Description
-a	--delay-override	Overrides delay request interval announced by master in worker state. This flag sets the <code>ptpengine</code> attribute to the following setting: <pre>ptpengine:log_delayreq_override</pre>
-A	--auto-lock	Uses preset or port mode-specific lock-file names. This flag is useful when you run multiple instances of the ptpd daemon.
-c file	--config-file file	Specifies the path of the configuration file.
-C	--foreground	Specifies the command to run in foreground. This flag sets the <code>global</code> attribute as follows: <pre>global:foreground=Y</pre>

Note: This option is ignored if the **ptpd** daemon is started from SRC.

Short flag	Long flag	Description
-d <i>domain_number</i>	--domain <i>domain_number</i>	Specifies the PTP domain number to become part of. This flag sets the <code>ptpengine</code> attribute as follows: <pre>ptpengine:domain</pre>
-D [DD]	--debug	Specifies the debug level. This flag sets the <code>global</code> attribute as follows: <pre>global:debug_level</pre> You can specify multiple instances to increase the debug level information. For example, the -D option specifies minimal debug information. The -DD option specifies detailed debug information. The -DDD option specifies maximum debug information.
-e <i>setting</i>	--explain <i>setting</i>	Shows help information for a single setting. This flag sets the section attribute as follows: <pre>section:key</pre>
-E	--e2e	Specifies end-to-end delay detection. This flag sets the <code>ptpengine</code> attribute as follows: <pre>ptpengine:delay_mechanism=E2E</pre>
-f <i>file</i>	--log-file <i>file</i>	Specifies the path of the log file. This flag sets the <code>global</code> attribute as follows: <pre>global:logfile</pre>
-h	--help	Shows help screen.
-H	--long-help	Shows detailed help for all settings and behaviors.
-i <i>dev</i>	--interface <i>dev</i>	Specifies the interface to be used for ptpd implementation. For example, <code>en0</code> . This flag sets the <code>ptpengine</code> attribute as follows: <pre>ptpengine:interface</pre>
-k	--check-config	Checks the PTP configuration and exits. Returns 0 if the configuration is correct.
-l <i>file</i>	--lockfile <i>file</i>	Specifies the path of lock-file. This flag sets the <code>global</code> attribute as follows: <pre>global:lock_file</pre>
-L	--ignore-lock	Skips checking and locking the lock-file. This flag sets the <code>global</code> attribute as follows: <pre>global:ignore_lock</pre>
-m	--masterslave	Specifies the full IEEE 1588 implementation: master, slave when not gandmaster (best master). This flag sets the <code>ptpengine</code> attribute as follows: <pre>ptpengine:preset=masterslave</pre>

Short flag	Long flag	Description
-M	--masteronly	Specifies master only mode: passive when not best GM. This flag sets the ptpengine attribute as follows: <pre>ptpengine:preset=masteronly</pre>
-n	--clock: no-adjust	Specifies not to adjust the clock. This flag sets the clock attribute as follows: <pre>clock:no_adjust</pre>
-O	--default-config	Shows default configuration and exits. The output can be used as a configuration file.
-p	--print-lockfile	Prints path of the lock-file and exits. This flag is useful for init scripts in combination with auto lock-files.
-P	--p2p	Specifies peer-to-peer delay detection. This flag sets the ptpengine attribute as follows <pre>ptpengine:delay_mechanism=P2P</pre>
-r number	--delay-interval number	Specifies the interval of delay request message (log 2). This flag sets the ptpengine attribute as follows: <pre>ptpengine:log_delayreq_interval</pre>
-R dir	--lock-directory dir	Specifies the directory to store lock-files. This flag sets the global attribute as follows: <pre>global:lock_directory</pre>
-S file	--statistics-file file	Specifies the path of statistics file. This flag sets the global attribute as follows: <pre>global:statistics_file</pre>
-s	--slaveonly	Turns on the slave-only mode. This flag sets the ptpengine attribute as follows: <pre>ptpengine:preset=slaveonly</pre>
-u	--unicast	Specifies unicast mode (no unicast negotiation) and sends all messages to IP. This flag sets the ptpengine attribute as follows: <pre>ptpengine:ip_mode=unicast + ptpengine:unicast_address</pre>
-v	--version	Prints the version string and exits.
-V	--verbose	Specifies the command to run in foreground and to log all the messages to the standard output. This flag sets the global attribute as follows: <pre>global:verbose_foreground=Y</pre>

Note: This option is ignored if the **ptpd** daemon is started from SRC.

Short flag	Long flag	Description
-y	--hybrid	Specifies hybrid mode: mixed multicast and unicast operation. Multicast for sync and announce, unicast for delay request and response. This flag sets the <code>ptpengine</code> attribute as follows:

```
ptpengine:ip_mode=hybrid
```

PTP daemon port states

The `ptpd` port can have the following states:

State	Description
init	Initializing
flt	Faulty
lstn_init	Listening (first time)
lstn_reset	Listening (subsequent reset)
pass	Passive (not best master, not announcing)
uncl	Uncalibrated
slv	Worker
pmst	Pre-master
mst	Master (active)
dsbl	Disabled
? (unk)	Unknown state

Statistics log file format

The following options are available when you enable the `ptpd` statistics log:

`ptpengine:log_statistics`

Updates the login information for each received PTP packet.

`ptpengine:statistics_file`

Specifies the location path of the statistics log file.

Note: This option enables statistics gathering.

When the statistics logging is enabled, a `ptpd` worker logs clock sync information when sync and delay response message are received. When the `ptpd` daemon starts up or flushes the log, a comment line (starting with #) is logged, containing the names of all columns. The log file is in the comma-separated values (CSV) format and can be easily imported into statistics tools and spreadsheet software packages for analysis and creating graphs. The size of the log files increases when you run the `ptpd` daemon for longer duration and with high message rates. Therefore, to reduce the number of messages logged, the `global:statistics_log_interval` setting can be used to limit the log output to one message per configured interval. The size and maximum number of the statistics log files can also be controlled.

The description of the columns in the statistics log file follows:

Timestamp

Time when the message was received. The date and time information are represented as text, UNIX time stamp (with fractional seconds), or both forms (in this case, an extra field is added), depending on the `global:statistics_timestamp_format` setting. When you import the log file to plotting software, if the software can understand UNIX time, set the time stamp format to `unix` or `both`, because some software do not interpret the fractional part of the second when it converts the date and time from text.

State

The state of the port. For more information about various port states, see [“PTP daemon port states” on page 1422](#).

Clock ID

Port identity of the current best master, as defined by IEEE 1588 standard. This ID is the local clock's ID if the local clock is the best master. This parameter is displayed as `clock_id` or `port (host)`. Port is the PTP clock port number, not the User Datagram Protocol (UDP) port numbers. The clock ID is an Extended Unique Identifier (EUI)-64 64-bit ID, converted from the 48-bit MAC address, by inserting `0xfffe` at the middle of the MAC address.

One-way delay

Current value of one-way delay (or mean-path delay) in seconds, calculated by the `ptpd` daemon that is in the worker state from the delay request and delay response message exchange.

Note: If this value remains at zero, it means that no delay response messages are being received, which might be because of a network issue.

Offset from master

Current offset value from master device in seconds. It is the main output of the PTP engine that is in the worker state. This value is the input for clock corrections in the clock servo algorithms. This value is typically measured when estimating the performance of the worker device.

Slave to master

Intermediate offset value (seconds) extracted from the delay request and delay response message exchange. This value is used for computing one-way delay. If the last value was rejected by a filter, the previous value is shown in the log file. This value is zero (0) if the delay response messages are not received.

Master to slave

Intermediate offset value (seconds) extracted from the sync messages. This value is used for computing the offset value from the master devices. If the last value was rejected by a filter, the previous value is shown in the log file.

Observed drift

The frequency difference between the worker clock and the master clock as measured by the integral accumulator of the clock control proportional integral (PI) servo model. This value stabilizes when the clock offset value is stabilized, and this value is used to detect clock stability.

Last packet received

This field shows which message was received last. It displays S for sync messages and D for delay response messages. If a worker device logs no D entries, it means that the worker device is not receiving delay response messages because of network issue.

One-way delay mean

One-way delay mean computed over the last sampling window.

One-way delay std dev

One-way delay standard deviation computed over the last sampling window.

Offset from master mean

Offset from master mean computed over the last sampling window.

Offset from master std dev

Offset from master standard deviation computed over the last sampling window.

Observed drift mean

Observed drift or local clock frequency adjustment mean computed over the last sampling window.

Observed drift std dev

Observed drift or local clock frequency adjustment standard deviation computed over the last sampling window. A lower value indicates that the clock is controlled less aggressively. Therefore, the value is more stable.

Note: All the statistical measures (mean and standard deviation) are computed and displayed only if the `ptpd` daemon was created by using the `--enable-statistics` flag. The duration of the sampling period is controlled with the `global:statistics_update_interval` setting.

Handling signals

The **ptpd** daemon handles the following signals:

Item	Description
SIGHUP	Reloads the configuration file (if used by the daemon) and reopens log files. The refresh subcommand of the SRC performs the same task.
SIGUSR1	When the subsystem is in the worker state, the ptpd daemon forces the clock to step to a current offset value from a master value.
SIGUSR2	Dumps all PTP protocol counters to current log target (and clears the counters if the <code>ptpengine:sigusr2_clears_counters</code> attribute is set).
SIGINT SIGTERM	Closes log files and other open files. It also cleans up lock file and exits.
SIGKILL	Forces an unclean exit.

Exit status

Upon exit, the **ptpd** daemon returns 0 on success, either successfully started in daemon mode, or exited cleanly. The value of 0 is also returned when the **-k (--check-config)** option is used and the configuration was correct. A nonzero exit code is returned on errors. The value of 127 is returned if the **ptpd** daemon is started by a non-root user. The value of 3 is returned on lock-file errors and when the **ptpd** daemon cannot be started as daemon. The value of 2 is returned on memory allocation errors when the daemon is started. For all other error conditions such as configuration errors, running the **ptpd** daemon in help mode or without any parameters, self shutdown of subsystems, and network startup errors, the value of 1 is returned.

Examples

1. To start the **ptpd** daemon with the SRC, enter the following command:

```
startsrc -s ptpd
```

2. To stop the **ptpd** daemon with the SRC, enter the following command:

```
stopsrc -s ptpd
```

3. To refresh the **ptpd** daemon with the SRC, enter the following command:

```
refresh -s ptpd
```

The **ptpd** daemon reloads the configuration file (if used by the daemon) and reopens log files.

4. To check whether the configuration file in the `/etc/ptpd2.conf` path is configured correctly, enter the following command:

```
ptpd -k
```

5. To view the meaning of a single setting, enter the following command:

```
ptpd -e ptpengine:interface
```

The output explains the meaning of the `ptpengine:interface` setting.

Files

Item	Description
<code>/etc/ptpd2.conf</code>	Default path of the ptpd daemon configuration file.

Item	Description
/usr/samples/tcpip/ptpd2/ ptpd2.conf	Sample file of the ptpd2.conf configuration file.

ptsc Command

Purpose

Collects information from a trusted platform module (TPM) in preparation for an attestation request from an openpts verifier.

Syntax

ptsc [options] [commands]

Description

The **ptsc** command is the openpts collector. The command is used to gather measurements and events from the TPM (through the **tscd** interface), construct reference manifests (RMs) and convey them when requested to the openpts verifier. When a system is first configured for trusted boot, the collector must be initialized by using the **-i** option. This option generates a UUID and an associated RM stored in the /var/ptsc/<UUID>/rm0.xml file. If the system is changed and a new RM is required, the **-u** option is used and the verifier must be reinitialized.

Flags

Item	Description
Commands	
-i	Initializes the openpts collector.
-s	Specifies the startup (both self-test and the timestamp).
-t	Indicates the self-test.
-u	Updates the RM.
-U	Updates the RM automatically.
-D	Displays the configuration settings of the target or ALL the options. This is the default setting.
-m	If -M mode
Options	
-c <i>configfile</i>	Changes the location of the configuration file. The default is /etc/ptsc.conf.
-P <i>name=value</i>	Sets the properties.
-R	Removes the RM.
-Z	Uses an SRK secret of all zeros.
Miscellaneous	
-h	Displays the command usage information.
-V	Displays the information in verbose mode. Multiple -V options increase the verbosity and is used for debugging.

Files

Item	Description
<code>/etc/ptsc.conf</code>	The configuration file. This is the default location of the configuration file.
<code>/var/ptsc/rm-uuid</code>	The UUID of the current RM.
<code>/var/ptsc/uuid</code>	The UUID of the collector.
<code>/var/ptsc/<UUID>/rm0.xml</code>	The reference manifest.

ptsevt Command

Purpose

Manages the notifications of updates to the AIX system boot image.

Syntax

```
ptsevt [ -a ] [ -r ] [ host port ]
```

```
ptsevt -c
```

```
ptsevt [ -u uuid ] -e
```

Description

The **ptsevt** utility delivers events, by using the **-e** option about the boot image updates to which the attestation software known as listeners can subscribe. The optional **-u** argument can be used to specify the universally unique identifier (UUID) of the collector of the AIX system that is being updated. If the **-u** argument is not specified, the **ptsevt** command uses the default value found in the `/var/ptsc/uuid` file.

Subscribers can be added or removed by using the **-a** and **-r** options, respectively. The host can be a symbolic address or an IP or IPv6 number, and the TCP port must be a decimal number.

The **-c** option is used to clear the subscription list.

Flags

Item	Description
-a	Adds the listener specified by the host and port arguments to the destinations mentioned in the subscriber list.
-c	Clears the list of subscribers.
-e	Sends an event notification to all subscribers in the list.
-r	Removes the listener specified by the host and port arguments from the list of subscribers.
-u	Specifies the UUID that is sent as part of the notification. By default, the ptsevt command uses the value found in the <code>/var/ptsc/uuid</code> file.

Files

Item	Description
<code>/var/ptsc/subscribers</code>	The subscribers list.
<code>/var/ptsc/subscribers.lock</code>	The subscribers list lock file.
<code>/var/ptsc/uuid</code>	The default UUID sent as part of the notification.

ptsevtd Command

Purpose

Manages the notifications of updates to the AIX system boot image.

Syntax

```
ptsevtd [ -c command ] [ -d ] [ -f ] [ -p port name ]
```

Description

The **ptsevtd** daemon listens to the events delivered by the **ptsevt** command when an attested system is being updated. By default, whenever an event is received, the **ptsevtd** command calls the **openpts** command with the universally unique identifier (UUID) of the system that is sending the event as the first argument. This process updates the corresponding reference manifest with the latest or the expected measurements. The **-c** option can be used to specify an alternative command that is called when a notification is received.

Use the **-f** option to run the daemon in the foreground. The **-d** option is specified multiple times to make the output more verbose. The **-p** argument specifies the port to be used to listen for event notifications.

Flags

Item	Description
-c	Specifies the command to call when a notification is received. If the option is not specified, the openpts command is used by default.
-d	Specifies the level to increase the verbosity of the output.
-f	Runs the listener in the foreground. The output is sent to the stderr console.
-p	Specifies the TCP port to use for event notifications. The default is 34185.

ptx Command

Purpose

Generates a permuted index.

Syntax

```
ptx [ -f ] [ -r ] [ -t ] [ -b Breakfile ] [ -g Number ] [ -w Number ] [ -i Ignore | -o Only ] [ - ] [ Infile [ Outfile ] ]
```

Description

The **ptx** command reads the specified English-language text (the *Infile* parameter), creates a rearranged index from it, and writes to the specified file (*Outfile*). Standard input and standard output are the defaults.

The **ptx** command searches the specified file (*Infile*) for keywords, sorts the lines, and generates the file *Outfile*. The *Outfile* file can then be processed with the **nroff** or **troff** command to produce a rearranged index.

The **ptx** command follows three steps:

1. Performs the permutation, generates one line for each keyword in an input line, and rotates the keyword to the front of the line.
2. Sorts the permuted file.

3. Rotates the sorted lines so that the keyword comes at the middle of each line.

The resulting lines in the *Outfile* file are in the following form:

```
.xx "" "before keyword" "keyword" "after keyword"
```

where *.xx* is an **nroff** or **troff** macro provided by the user or by the **ptx** command. The **mptx** macro package provides the *.xx* macro definition.

The *before* keyword, and *keyword*, and *after* keyword fields incorporate as much of the line as can fit around the keyword when it is printed. The first field and last field, at least one of which is always the empty string, are wrapped to fit in the unused space at the opposite end of the line.

Notes:

1. Line-length counts do not account for overstriking or proportional spacing.
2. Lines that contain a ~ (tilde) do not work, because the **ptx** command uses that character internally.
3. The **ptx** command does not discard non-alphanumeric characters.

Flags

Item	Description
-b <i>BreakFile</i>	Uses the characters in the specified break file to separate words. Tab characters, new-line characters, and spaces are always used as break characters.
-f	Folds uppercase and lowercase characters for sorting.
-g <i>Number</i>	Uses the specified number as the number of characters that the ptx command reserves for each gap among the four parts of the line as it is printed. The default <i>Number</i> variable value is 3.
-i <i>Ignore</i>	Does not use any words specified in the <i>Ignore</i> file as keywords. If the -i and -o flags are not used, the <i>/usr/lib/eign</i> file is the default <i>Ignore</i> file.
-o <i>Only</i>	Uses only the words specified in the <i>Only</i> file as keywords.
-r	Considers any leading non-blank characters of each input line as reference identifiers separate from the text of the line. Attaches the identifier as a fifth field on each output line.
-t	Prepares the output for the phototypesetter.
-w <i>Number</i>	Uses the specified number as the length of the output line. The default line length is 72 characters for the nroff command and 100 for the troff command.
--	(double dash) Indicates the end of flags.

Parameters

Item	Description
<i>Infile</i>	Specifies the English-language text. Standard input is the default file. The ptx command searches the specified file for keywords, sorts the lines, and generates the file <i>Outfile</i> .
<i>Outfile</i>	Specifies the file to which the ptx command writes the index created from the <i>Infile</i> file. Standard output is the default file. The <i>Outfile</i> file can be processed with the nroff or troff command to produce a rearranged index.

Files

Item	Description
<code>/usr/lib/eign</code>	Contains the default <i>Ignore</i> file.
<code>/usr/share/lib/tmac/tmac.ptx</code>	Contains the macro file.

pvcauth command

Purpose

The **pvcauth** command is used to authenticate with an IBM Power Virtualization Center (PowerVC) and get a token. This token is required to use the PowerVC services for the AIX Live Update operation. This token is valid only for a set time period. This command can also be used to invalidate a token.

Syntax

To authenticate with PowerVC and to get a token, use the following syntax:

```
pvcauth [ -u user_name ] [ -p password ] -a pvc [ -o project ] [ -P port ]
```

To invalidate and remove a previously generated token, use the following syntax:

```
pvcauth -r -a pvc
```

To list all the known PowerVC authentication tokens, use the following syntax:

```
pvcauth -l
```

Description

You can use the **pvcauth** command if you have access to all types of object and if you have appropriate PowerVC administrative authority. The **pvcauth** command generates a token that can be used by an AIX partition administrator to perform the Live Update operation. If the command succeeds, a token is stored in the kernel. You can now use the **geninstall** command to perform the Live Update operation.

To use this command, you must have authority to perform the following tasks:

- Power on a managed partition.
- Shut down a managed partition.
- Create a managed partition.
- Remove a managed partition.
- Manage storage volumes.
- Manage network adapters.

Parameters

password

A string of up to 64 characters that specifies a password.

port

A string of up to 16 characters that specifies a port number to contact PowerVC. The default value of this parameter is 5000.

project

A string of up to 64 characters that specifies the PowerVC project name.

pvc

A string of up to 64 characters that specifies either the host name or the IP address of the PowerVC for authentication.

user_name

A string of up to 64 characters that specifies the PowerVC user name.

Flags**-a pvc**

Specifies the host name or the IP address of PowerVC for authentication.

-o project

Specifies a PowerVC project name that is used to authenticate with PowerVC. If you do not specify the **-o** flag, the project name is set to the default name as **ibm-default**.

-l

Lists all the known PowerVC authentication tokens. The information that is listed includes the current Time To Live (TTL) value for the token.

-p password

Specifies the PowerVC password for authentication. If you do not specify the **-p** flag, you are prompted for the password after you run the **pvcauth** command.

-P port

Specifies a port number that can be used to contact PowerVC.

-r

Removes the token that is generated by PowerVC.

-u user_name

Specifies the PowerVC user name that can be used for authentication. You must have access to all types of objects and appropriate PowerVC administrative authority.

Examples

1. To authenticate with an HMC, called `apollo`, which has a firewall and in which the PowerVC port 5000 is not accessible, a rebound proxy node can be set up to use a different port that is open. To authenticate a logical partition called `mylpar` and to use the SSH client with port 14111 on a proxy node that is called `proxy1`, enter the following commands:

```
root @ proxy1: /
# ssh -R localhost:14111:apollo:5000 root@mylpar

root @ mylpar: /
# pvcauth -a localhost -u hscroot -P 14111
Enter HMC password:
```

You can specify the `management_console` attribute as `localhost` in the `pvc` stanza of the **lvupdate.data** file to initiate the Live Update operation.

2. To authenticate with PowerVC that has an IP address 5.5.55.121 with password prompt, enter the following command:

```
# pvcauth -a 5.5.55.121 -u root
Enter password for root:
```

3. To invalidate a previous authentication with PowerVC that has an IP address 5.5.55.121, enter the following command:

```
# pvcauth -r -a 5.5.55.121
```

pvi Command

Purpose

Provides a privileged editor so that you can access privileged files.

Syntax

```
pvi [ -l ] [ -R ] [ -w Number ] [ -c | + [ Subcommand ] ] [ File ]
```

Description

The **pvi** command calls the **pvi** editor, a privileged version of the **vi** editor, to edit the file specified by the *File* parameter. Only one file can be opened at a time, and this file must have the security attributes that are defined in the privileged file database. You can display the file in the editor only when at least one of the authorizations matches at least one of the authorizations in the **readauths** or the **writeauths** attribute for the file. The contents of the buffer can then be modified. You can write to the file using the editor only when at least one of the authorizations matches at least one of the authorizations in the **writeauths** attribute for the file. Files opened by the **pvi** command can only be written to the same path they were opened from.

You enter and leave the **pvi** editor in command mode, but to add or change text, you must enter the text input mode. See the **text input mode** for information about the subcommands that initiate the text input mode. You can save the text to a file with one of the **:w** commands, and exit the **pvi** editor using the **:q** command.

The full-screen display editor, which is started by the **pvi** command, is based on the **ex** editor. You can use the **ex** subcommands within the **pvi** editor. Subcommands function at the cursor position on the display screen.

The **pvi** editor makes a copy of the file that you are editing in an edit buffer. The contents of the file are not changed until you save the changes.

Note: There are several functions of the **vi** editor that you cannot use with the **pvi** editor. If you refer to the 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 **pvi** editor. Only one buffer is opened at a time and a file can only be written to the same path from which it was opened.

Editor Limitations

The maximum limits of the **pvi** editor assume single-byte characters:

- 256 characters per a 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 **pvi** editor operates in the following modes:

Item	Description
command mode	The pvi editor starts in the command mode. Any subcommand can be called except those that only correct text during the text input mode. To see a description of the subcommands, refer to the topics in " Subcommands for the pvi editor ". To identify the subcommands that cannot be called from the command mode, refer to " Changing Text While in Input Mode ". The pvi editor returns to the command mode when the subcommands and other modes end. Press the Esc key to cancel a partial subcommand.
text input mode	The pvi editor enters the 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 the 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 the text input mode. To see a list of the subcommands, refer to the topics in " Subcommands for the pvi Editor ". To return to command mode from text input mode, press Esc 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 pvi editor places the cursor at the bottom of the screen so you can enter the remaining command characters. To run the subcommand, press Enter . To cancel the subcommand, press Ctrl + C to create an INTERRUPT signal. When you use the colon (:) to enter the last line mode, the following characters have special meaning when used before the commands that specify counts: <ul style="list-style-type: none"> % All lines regardless of the cursor position \$ Last line . Current line

Customizing the **pvi Editor**

You can customize the **pvi** editor on a temporary basis by following the directions in "[Setting vi Editor Options](#)".

Subcommands for the **pvi Editor**

You can find information about the **vi** editor subcommands that are applicable to the **pvi** editor in the following list:

- [vi General Subcommand Syntax](#).
- [vi Subcommands for Adjusting the Screen](#).
- [Editing Text with the vi 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 the editing begins. This provides a line-oriented text editor. When you specify a null operand for the <i>Subcommand</i> parameter, for example, <code>-c ''</code> , the editor places the cursor on the last line of the file.
-l	Enters the editor in the list processing (LISP) mode. In this mode, the editor indents appropriately for LISP mode, and the (, { , [, and] subcommands are modified to act in 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 <i>Number</i>	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.
+ [<i>Subcommand</i>]	Same as the -c Subcommand.

Security

Access Control: This command grants the execute (x) access to all users.

Role-Based Access Control: The command grants read access to a file if the user has an authorization that matches one in the **readauths** or the **writeauths** authorization list in the privileged file database. The command only grants the write access to a file if the user has an authorization that matches one in the **writeauths** authorization list in the privileged file 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.

Examples

1. To call a privileged editor to edit the **plans** file, enter:

```
pvi plans
```

This command puts the **pvi** editor into the command mode. To add or change text, you must enter the text input mode or use a command accepted in the command mode.

2. To save the text that you create with the **pvi** editor, leave the text input mode by pressing **Esc**, and then enter:

```
:w
```

3. To exit the **pvi** editor from the text input mode, press **Esc** to enter the command mode, and then enter:

```
:q!
```

If the editor is already in the command mode, you do not need to press **Esc** before giving the quit (**q!**) command.

File

Item	Description
/usr/bin/pvi	Contains the pvi command.

Item	Description
<code>/etc/security/privfiles</code>	Contains the security attributes for the privileged files.

pwchange Command

Purpose

Change user authentication and privacy keys dynamically.

Syntax

```
pwchange [ -e ] [ -d DebugLevel ] [ -p Protocol ] [ -u KeyUsage ] [ -s ] [ OldPassword NewPassword ]
[ IPAddress | HostName | EngineID ]
```

Description

The **pwchange** command is provided to facilitate dynamic changes of user authentication and privacy keys. Dynamic configuration of authentication and privacy keys is done by doing **set** commands to objects of syntax `keyChange`. The `keyChange` syntax provides a way of changing keys without requiring that the actual keys (either new or old) be flowed directly across the wire, which would not be secure. Instead, if an object, such as **usmUserAuthKeyChange** (for example) is to be set, the `keyChange` value must be derived from the old and new passwords and the `engineID` of the agent at which the key will be used. The **pwchange** command is used to generate the `keyChange` values.

The **pwchange** command generates different output, depending on which protocol and what key usage is selected. Keychange values are typically twice as long as the key to be changed.

Flags

Item	Description
<code>-d <i>DebugLevel</i></code>	This flag indicates what level of debug information is desired. Debug tracing is either on or off: 1 causes debug tracing to be generated to the screen of the command issuer (sysout). Debug tracing is off (0) by default.
<code>-e</code>	This flag indicates that the agent for which the keychange value is being defined is identified by <code>engineID</code> rather than by IP address or host name.
<code>-p <i>Protocol</i></code>	This flag indicates the protocols for which the keychange values should be generated. Valid values are: <ul style="list-style-type: none"> HMAC-MD5 Generates keychange values for use with the HMAC-MD5 authentication protocol. HMAC-SHA Generates keychange values for use with the HMAC-SHA authentication protocol. all Generates both HMAC-MD5 and HMAC-SHA keychange values. The default is that keychange values for the HMAC-MD5 protocol are generated.

Item	Description
-s	This flag indicates that output should be displayed with additional spaces to improve readability. By default, data is displayed in a condensed format to facilitate cut-and-paste operations on the keychange values onto command lines in shell scripts.
-u <i>KeyUsage</i>	This flag indicates the usage intended for the keychange value. Valid values are: <p>auth An authentication keychange value.</p> <p>priv A privacy keychange value.</p> <p>all Both authentication and privacy keychange values.</p> <p>Note: There is no difference between a keychange value generated for authentication and a keychange value generated for privacy. However, the length of privacy keychange values depends on whether the keychange value is localized.</p>

Parameters

Item	Description
<i>EngineID</i>	Specifies the engineID (1-32 octets, 2-64 hex digits) of the destination host at which the key is to be used. The engineID must be a string of 1-32 octets (2-64 hex digits). The default is that the agent identification is not an engineID.
<i>HostName</i>	Specifies the destination host at which the key is to be used.
<i>IPAddress</i>	Specifies an IPv4 or an IPv6 address of the agent at the destination host at which the key is to be used.
<i>NewPassword</i>	Specifies the password that will be used in generating the new key. The password must be between eight and 255 characters long.
<i>OldPassword</i>	Specifies the password that was used in generating the key originally. The password must be between eight and 255 characters long.

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

The **pwchange** command generates different output depending on which protocol and what key usage is selected. Key change values are typically twice as long as the key to be changed.

1. The following command demonstrates how the **pwchange** command can be used:

```
pwchange oldpassword newpassword 9.67.113.79
```

The output of this command looks similar to:

```
Dump of 32 byte HMAC-MD5 authKey keyChange value:  
3eca6ff34b59010d262845210a401656  
78dd9646e31e9f890480a233dbe1114d
```

The value to be set should be passed as a hex value with the **clsnmp** command (all on one line):

```
clsnmp set usmUserAuthKeyChange.12.0.0.0.2.0.0.0.9.67.113.79.2.117.49  
\ '3eca6ff34b59010d262845210a40165678dd9646e31e9f890480a233dbe1114d\ 'h
```

Note: The backslash in the preceding example is required before the single quotation mark to enable AIX to correctly interpret the hexadecimal value.

The index of the usmUserTable is made up of the EngineID and the ASCII representation of the user name. In this case it is 2 characters long and translates to 117.49.

Note: **pwchange** incorporates a random component in generating keys and keyChange values. The output from multiple commands with the same input does not produce duplicate results.

2. The following command demonstrates how the **pwchange** command can be used with IPv6 address:

```
pwchange oldpassword newpassword 2000:1:1:1:209:6bff:feae:6d67
```

The output of this command looks similar to:

```
Dump of 32 byte HMAC-MD5 authKey keyChange value:  
0000774adc53ba4b0427dc2f65568435  
721847d1b5cb597daa85d003033afba3
```

The value to be set should be passed as a hex value with the **clsnmp** command (all on one line):

```
clsnmp set usmUserAuthKeyChange.21.128.0.0.2.2.32.0.0.1.0.1.0.1.2.9.107.255.254.174.  
109.103.6.105.112.118.54.117.49  
\ '36133c694155026620637761f835ef616de294f37f758c74ff1544ca3de279b8\ 'h
```

Note: The backslash in the preceding example is required before the single quotation mark to enable AIX to correctly interpret the hexadecimal value.

The index of the usmUserTable is made up of the EngineID, in this case 21 octets: 128.0.0.2.2.32.0.0.1.0.1.0.1.2.9.107.255.254.174.109.103; And the ASCII representation of the user name, in this case it is 6 characters long and translates to 105.112.118.54.117.49.

Note: The **pwchange** command incorporates a random component in generating keys and keyChange values. The output from multiple commands with the same input does not produce duplicate results.

pwck Command

Purpose

Verifies the correctness of local authentication information.

Syntax

pwck

Description

The **pwck** command verifies the correctness of the password information in the user database files by checking the definitions for all users. The **pwck** command internally calls the **pwdck** command with **-n** and **ALL** options.

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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To verify that all the users and administrators exist in the user database, and have any errors reported (but not fixed), enter:

```
pwck
```

Files

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

pwd Command

Purpose

Displays the path name of the working directory.

Syntax

```
pwd [ -L | -P ]
```

Description

The **pwd** command writes to standard output the full path name of your current directory (from the root directory). All directories are separated by a / (slash). The root directory is represented by the first /, and the last directory named is your current directory.

Flags

-L

Displays the value of the PWD environment variable if the PWD environment variable contains an absolute path name of the current directory that does not contain the file names . (dot) or .. (dot-dot). Otherwise, the **-L** flag behaves the same as the **-P** flag.

-P

Displays the absolute path name of the current directory. The absolute path name displayed with the **-P** flag does not contain file names that, in the context of the path name, refer to files of type symbolic link.

Exit Status

This command returns the following exit values:

Item	Description
-------------	--------------------

0	Successful completion.
----------	------------------------

>0	An error occurred.
--------------	--------------------

Examples

Entering:

```
pwd
```

displays the current directory as:

```
/home/thomas
```

Files

Item	Description
/usr/bin/pwd	Contains the pwd command.

pwdadm Command

Purpose

Administers users' passwords.

Syntax

```
pwdadm [ -R load_module ] [ -f Flags | -q | -c ] User
```

Description

The **pwdadm** command administers users' passwords. The root user or a member of the security group can supply or change the password of the user specified by the *User* parameter. The invoker of the command must provide a password when queried before being allowed to change the other user's password. When the command executes, it sets the **ADMCHG** attribute. This forces the user to change the password the next time a **su** command is given for the user.

Note: The behavior described for this command is for a local user. For users defined in a remote domain, attributes will be retrieved and stored in the remote domain rather than in the local files.

Root users and members of the security group should not change their personal password with this command. The **ADMCHG** attribute would require them to change their password again the next time a **login** command or an **su** command is given for the user. Only the root user or a user with PasswdAdmin authorization can change password information for administrative users, who have the **admin** attribute set to true in the **/etc/security/user** file.

Only the root user, a member of the security group, or a user with PasswdManage authorization can supply or change the password of the user specified by the *User* parameter.

When this command is executed, the password field for the user in the **/etc/passwd** file is set to ! (exclamation point), indicating that an encrypted version of the password is in the **/etc/security/passwd** file. The **ADMCHG** attribute is set when the root user or a member of the security group changes a user's password with the **pwdadm** command.

A new password must be defined according to the rules in the **/etc/security/user** file, unless the **-f NOCHECK** flag is included. Only 7-bit characters are supported in passwords. By including the **-f** flag with the **pwdadm** command, the root user or a member of the security group can set attributes that change the password rules. If there is no password entry in the **/etc/security/passwd** file when the **-f** flag is used, the password field in the **/etc/passwd** file is set to ! (exclamation point) and an * (asterisk) appears in the password= field to indicate that no password has been set.

The **-q** flag permits the root user or members of the security group to query password information. Only the status of the **lastupdate** attribute and the **flags** attribute appear. The encrypted password remains hidden.

The **-c** flag clears all password flags for the user.

Flags

Item	Description
-c	Clears all password flags for the user.
-f <i>Flags</i>	Specifies the flags attribute of a password. The <i>Flags</i> variable must be from the following list of comma-separated attributes: NOCHECK Signifies that new passwords need not follow the guidelines established in the /etc/security/user file for password composition. ADMIN Specifies that password information may be changed only by the root user. Only the root user can enable or disable this attribute. ADMCHG Resets the ADMCHG attribute without changing the user's password. This forces the user to change passwords the next time a login command or an su command is given for the user. The attribute is cleared when the user specified by the <i>User</i> parameter resets the password.
-q	Queries the status of the password. The values of the lastupdate attribute and the flags attribute appear.
-R <i>load_module</i>	Specifies the loadable I&A module that is used to change the user's attributes.

Security

Access Control: Only the root user and members of the security group should have execute (x) access to this command. The command should have the **trusted computing base** attribute and be **setuid** to the root user to have write (w) access to the **/etc/passwd** file, the **/etc/security/passwd** file, and other user database files.

Files Accessed:

Mode	File
rw	/etc/passwd
rw	/etc/security/passwd
r	/etc/security/user

Auditing Events:

Event	Information
PASSWORD_Change	user
PASSWORD_Flags	user, flags

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 set a password for user susan, a member of the security group enters:

```
pwdadm susan
```

When prompted, the user who invoked the command is prompted for a password before Susan's password can be changed.

2. To query the password status for user susan, a member of the security group enters:

```
pwdadm -q susan
```

This command displays values for the **lastupdate** attribute and the **flags** attribute. The following example shows what appears when the **NOCHECK** and **ADMCHG flags** attributes are in effect:

```
susan:
      lastupdate=
      flags= NOCHECK,ADMCHG
```

Files

Item	Description
/usr/bin/pwdadm	Contains the pwdadm command.
/etc/security/passwd	Contains password information.
html	

pwdck Command

Purpose

Verifies the correctness of local authentication information.

Syntax

```
pwdck { -p | -n | -t | -y } [-l]{ ALL | User ... }
```

Description

The **pwdck** command verifies the correctness of the password information 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.

Note: This command writes its messages to **stderr**.

You must select a flag to indicate whether the system should try to fix erroneous attributes. The following attributes are checked for locally defined users in the **/etc/passwd** file:

Item	Description
entry	Ensures that each entry is readable and that it contains at least two : (colons). If you indicate that the system should fix errors, the entire entry is discarded.
passwd	Ensures that the password field is an ! (exclamation point). If you indicate that the system should fix errors, it transfers the information in the password field to the /etc/security/passwd file, updates the lastupdate attribute in the /etc/security/passwd file, and then replaces the password field in the /etc/passwd file with an !. In general, passwords are required if the minalpha , minother , or minlen password restriction is set to a nonzero value in the /etc/security/user file.
user	Ensures that the user name is a unique string of 8 bytes or less. It cannot begin with a + (plus sign), a : (colon), a - (minus sign), or a ~ (tilde). It cannot contain a : (colon) in the string and cannot be the ALL , default , or * keywords. If you indicate that the system should fix errors, it removes this user's entry line from the /etc/passwd file. If the user name starts with a + or a - symbol, the user is not locally defined, and checks are not performed.

Attributes checked in the **/etc/security/passwd** file are:

Item	Description
line	Ensures that each line is readable and is part of a stanza. Any invalid line is discarded.
password	Ensures that the password attribute exists and is not blank, if passwords are required on the system. If you indicate that the system should fix errors, the password is set to * (asterisk), and the lastupdate attribute is discarded. In general, passwords are required if either of the minalpha or minother password restrictions are set to nonzero values in the /etc/security/user file. If a user's flags attribute specifies the NOCHECK keyword, a password is not required for this user, and the check is ignored.
lastupdate	Ensures that the lastupdate attribute exists for a valid non-blank password, and that its time is prior to the current time. If you indicate that the system should fix errors, the lastupdate attribute is discarded or updated, depending on the password attribute. The lastupdate attribute is discarded if the password attribute doesn't exist, or equals a blank or an * (asterisk). Otherwise, the lastupdate time is set to the current time.
flags	Ensures that the flags attribute contains only the keywords ADMIN , ADMCHG , and NOCHECK . If you indicate that the system should fix errors, it deletes any undefined flags.

Attributes checked in the **/etc/security/user** file are:

Item	Description
auth1	Ensures that each SYSTEM;username entry defined for a local user has an username entry in the /etc/security/passwd file. If you indicate that the system should fix errors, a stanza is added to the /etc/security/passwd file for each missing entry, in the following format:

```
username:  
    password = *
```

If a user's entry and a default entry both are missing from the **/etc/security/user** file, the system assumes the following values and the check on **auth1** is performed:

```
auth1 = SYSTEM;user
```

Note: The **auth1** attribute is deprecated and should not be used.

Item Description

auth2 Ensures that each `authname;username` entry defined for a local user has an `username` entry in the `/etc/security/passwd` file. If you indicate that the system should fix errors, an entry is added for each missing entry.

If a user's entry and a default entry both are missing from the `/etc/security/user` file, the system assumes the following values and the check on **auth2** is performed:

```
auth2 = NONE
```

When ALL is specified, the **pwdck** command ensures that each stanza in the `/etc/security/passwd` file corresponds to an authentication name of a local user as a `SYSTEM;username` entry in the `/etc/security/user` file. If you indicate that the system should fix errors, a stanza which does not correspond to an username entry in the `/etc/security/user` file is discarded from the `/etc/security/passwd` file.

The **pwdck** command locks the `/etc/passwd` file and the `/etc/security/passwd` file when it updates them. If either of these files are locked by another process, the **pwdck** command waits a few minutes for the files to be unlocked, and terminates if this does not happen.

The **pwdck** command checks to see if the `/etc/passwd` file and the `/etc/security/passwd` file are modified by another process while the current **pwdck** process is running. If you indicate that the system should fix errors, the **pwdck** command updates the `/etc/passwd` file and the `/etc/security/passwd` file, and may overwrite any changes made by the other process.

Note: The **pwdck** command disables any Extended Access Control Lists (ACLs) on the files when it fixes errors and reports them.

The **pwdck** command also 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 alright 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.

Generally, the **sysck** command calls the **pwdck** command as part of the verification of a trusted-system installation. In addition, the root user or a member of the security group can enter the command.

Note: The **auth2** attribute is deprecated and should not be used.

Flags

Item Description

m

- l** Locks file during entire run.
- 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.

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, to read and write the authentication information, and have the **trusted computing base** attribute.

Files Accessed:

Mode	File
rw	/etc/passwd
r	/etc/security/user
rw	/etc/security/passwd
r	/etc/security/login.cfg

Auditing Events:

Event	Information
PASSWORD_Check	user, error/fix, status
PASSWORD_Ckerr	file/user, 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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To verify that all local users have valid passwords, enter:

```
pwdck -y ALL
```

This reports errors, and fixes them.

2. To ensure that user ariel has a valid stanza in the **/etc/security/passwd** file, enter:

```
pwdck -y ariel
```

Files

Item	Description
/usr/bin/pwdck	Contains the pwdck command.
/etc/security/passwd	Contains actual passwords and security information.
/etc/security/login.cfg	Contains configuration information and password restrictions.

pwtokey Command

Purpose

Converts passwords into localized and non localized authentication and privacy keys.

Syntax

```
pwtokey [-e] [ -d DebugLevel ] [ -p Protocol ] [ -u KeyUsage ] [ -s ] Password [ EngineID | HostName | IPAddress ]
```

Description

AIX provides a facility called **pwtokey** that allows conversion of passwords into localized and nonlocalized authentication and privacy keys. The **pwtokey** procedure takes as input a password and an identifier of the agent and generates authentication and privacy keys. Since the procedure used by the pwtokey facility is the same algorithm used by the **clsnmp** command, the person configuring the SNMP agent can

generate appropriate authentication and privacy keys to put in the **snmpd.conf** file for a user, given a particular password and the IP address at which the agent will run.

If the IP address or the hostname is specified, the SNMP agent must be an AIX agent. The engineID will be created using a vendor-specific formula that incorporates the IP address of the agent and an enterprise ID representing AIX.

Flags

Item	Description
-d <i>DebugLevel</i>	This flag indicates what level of debug information is desired. Debug tracing is either on or off, so a value of 1 causes debug tracing to be generated to the screen of the command issuer (sysout), and a value of 0 specifies that no debug tracing be generated. Debug tracing is off (0) by default.
-e	This flag indicates that the agent for which the key is being defined is identified by engineID rather than by IP address or host name.
-p <i>Protocol</i>	This flag indicates the protocols for which the keys should be generated. Valid values are: HMAC-MD5 Generates keys for use with the HMAC-MD5 authentication protocol. HMAC-SHA Generates keys for use with the HMAC-SHA authentication protocol all Generates both HMAC-MD5 and HMAC-SHA keys. The default is that keys for the HMAC-MD5 protocol are generated.
-s	This flag indicates that output data should be displayed with additional spaces to improve readability. By default, data is displayed in a condensed format to facilitate cut-and-paste operations on the keys into configuration files or command lines.
-u <i>KeyUsage</i>	This flag indicates the usage intended for the key. Valid values are: auth An authentication key. priv A privacy key. all Both authentication and privacy keys. Note: There is no difference between a key generated for authentication and a key generated for privacy. However, the length of privacy keys depends on whether the key is localized or not.

Parameters

Item	Description
<i>EngineID</i>	Specifies the engineID of the SNMP agent at which the key will be used. The engineID is determined at SNMP agent initialization from the <code>snmpd.boots</code> file. The engineID must be a string of 1-32 octets (2-64 hex digits). The default is that the agent identification is not an engineID.
<i>HostName</i>	Specifies the SNMP agent at which the key will be used on an SNMP request.
<i>IPAddress</i>	Specifies an IPv4 or an IPv6 address of the SNMP agent at which the key will be used on an SNMP request.
<i>Password</i>	Specifies the text string to be used in generating the keys. The password must be in the range of 8-255 characters long. In general, while any printable characters can be used in the passwords, the AIX shell may interpret some characters rather than passing them to the <code>pwtokey</code> command. Include passwords in single quotes to avoid interpretation of the characters by the AIX shell. Note: This password is not related to the community name (or "password") used with community-based security (SNMPv1 and SNMPv2c). This password is used only to generate keys for user-based security, an entirely different security scheme.

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. This example shows a simple invocation of the `pwtokey` command:

```
pwtokey testpassword 9.67.113.79
```

The output from this command looks similar to the following:

```
Display of 16 byte HMAC-MD5 authKey:  
775b109f79a6b71f94cca5d22451cc0e  
  
Display of 16 byte HMAC-MD5 localized authKey:  
de25243d5c2765f0ce273e4bcf941701
```

As this example shows, `pwtokey` generates two keys—one that is localized (has been tailored to be usable only at the agent identified) and one that has not been localized. Typically, the localized key is used in the configuration for the SNMP agent. The nonlocalized key is used in the configuration for the `clsnmp` command.

- The **pwtokey** can be invoked requesting HMAC-SHA keys for both authentication and privacy, as in the following example:

```
pwtokey -p HMAC-SHA -u all testpassword 9.67.113.79
```

The output of this command looks similar to the following:

```
Display of 20 byte HMAC-SHA authKey:  
b267809aee4b8ef450a7872d6e348713f04b9c50  
  
Display of 20 byte HMAC-SHA localized authKey:  
e5438092d1098a43e27e507e50d32c0edaa39b7c  
  
Display of 20 byte HMAC-SHA privKey:  
b267809aee4b8ef450a7872d6e348713f04b9c50  
  
Display of 16 byte HMAC-SHA localized privKey:  
e5438092d1098a43e27e507e50d32c0e
```

The output for the privacy keys is the same as the output for the authentication keys, except that the localized privacy key has been truncated to 16 bytes, as is required for DES.

Note: If encryption is used, it is more secure to use different passwords for authentication and privacy.

- The following example shows that the **pwtokey** command is using an IPv6 address:

```
pwtokey testpassword 2000:1:1:1:209:6bff:feae:6d67
```

The output from this command looks similar to the following:

```
Display of 16 byte HMAC-MD5 authKey:  
775b109f79a6b71f94cca5d22451cc0e  
  
Display of 16 byte HMAC-MD5 localized authKey:  
2a30fe53690fa6b62dba3f9ea30e11fb
```

As this example shows, the **pwtokey** command generates two keys: one that is localized (has been tailored to be usable only at the agent identified) and one that has not been localized. Typically, the localized key is used in the configuration for the SNMP agent. The non-localized key is used in the configuration for the **clsnmp** command. SNMP agent at which the key will be used on an SNMP request is an IPv6 address.

pxed Command

Purpose

Implements a Preboot Execution Environment (PXE) Proxy Dynamic Host Configuration Protocol (DHCP) server.

Syntax

To start the **pxed** daemon using the system resource controller:

```
startsrc -s pxed [ -a]
```

To start the **pxed** daemon without using the system resource controller:

```
pxed [ -f ConfigurationFile]
```

Description

The Preboot Execution Environment defines a protocol and mechanism through which network-connected client systems can automatically download boot images from a network server to start their operating system. As an extension to the BOOTP and DHCP protocols, it provides the configuration ability for administrators that are not necessarily DHCP or network administrators to manage the operating systems installed on the PXE-capable client systems.

Like a DHCP server, the PXE Proxy DHCP server provides information needed by a PXE client to locate and download its appropriate boot files from a network server. However, the PXE Proxy DHCP server does not administer client IP addresses or other DHCP client options.

The PXE Proxy DHCP server is intended to be used when the management of the system boot images must be separated from the management of the DHCP addresses and DHCP client network configurations. The **pxed** daemon can be configured to run on a system that is the DHCP server or is not the DHCP server.

Flags

Item	Description
-a	The argument to be supplied.
-f <i>ConfigurationFile</i>	Specifies the path and name of the configuration file that is to be used by the server. If unspecified, the default is /etc/pxed.cnf .

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.

The PXE protocol allows for a nonprivileged user to be the administrator of the PXE client boot images because the **pxed** daemon listens to client messages on ports other than the well-known, protected DHCP server port. However, to configure such an environment, the DHCP server must be running on the same server system as the **pxed** daemon, and the file permissions on the **pxed** daemon must be changed for non-root execution.

Files

Item	Description
/usr/sbin/pxed	Contains the PXE Proxy DHCP server daemon.
/usr/sbin/db_file.dhcpcd	Implements a database to be used by the PXE Proxy DHCP server and the DHCP server to store, retrieve, and manage configuration information.
/etc/pxed.cnf	The default configuration file for the pxed daemon.

q

The following AIX commands begin with the letter *q*.

qadm Command

Purpose

Performs system administration functions for the printer spooling system.

Syntax

```
qadm { -G } | { { -D Printer } [ -K Printer ] [ -U Printer ] [ -X Printer ] }
```

Description

The **qadm** command is a front-end command to the **enq** command. This command brings printers, queues, and the spooling system up or down and also cancels jobs. The **qadm** command translates the requested flags into a format that can be run by the **enq** command.

The **qadm** command works only on local print jobs. Remote print is not supported.

Note: You must either have root user authority or belong to the printq group to run this command.

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

Flags

Item	Description
-D Printer	Brings down the printer you name in the <i>Printer</i> variable. The qdaemon process stops sending jobs to the device. Entering the qchk -P Printer command, where <i>Printer</i> matches the <i>Printer</i> variable in the -D flag, reports the device is <i>down</i> . The qadm command allows current jobs to finish before stopping the printer.
-G	Gracefully brings down the queuing system. This flag temporarily interrupts the qdaemon process after all currently running jobs on all queues are finished. Use of this flag is the only way to bring the system down without causing such problems as jobs hanging up in the queue.
-K Printer	Brings down the printer that you name in the <i>Printer</i> variable, ending all current jobs immediately. Jobs remain in the queue and run again when the printer is brought back.
-U Printer	Brings up the printer that you name in the <i>Printer</i> variable. The qdaemon process sends jobs to the printer again. Entering the qchk -P Printer command, where <i>Printer</i> matches the <i>Printer</i> variable in the -U flag, reports the device is <i>ready</i> .
-X Printer	Cancels all the jobs of the user that executed the command. If you have root user privileges or are a member of the printq group, then all jobs on the queue system will be canceled.

Note: When **-U** and **-D** flags are used together, the **-U** flag has higher priority.

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 bring the queuing system down gracefully, enter:

```
qadm -G
```

2. To cancel all of a particular user's jobs on printer lp0, or all jobs on printer lp0 if you are have root user authority, enter:

```
qadm -X lp0
```

3. To bring up the printer lp0 attached to queue lp0, enter:

```
qadm -U lp0:lp0
```

Files

Item	Description
<code>/usr/sbin/qdaemon</code>	Contains the qdaemon daemon.
<code>/var/spool/lpd/qdir/*</code>	Contains the job description files.
<code>/var/spool/lpd/stat/*</code>	Contains information on the status of the devices.
<code>/var/spool/qdaemon/*</code>	Contains the temporary copies of enqueued files.
<code>/etc/qconfig</code>	Contains the configuration file.
<code>/etc/qconfig.bin</code>	Contains the digested, binary version of the <code>/etc/qconfig</code> file.

qcan Command

Purpose

Cancels a print job.

Syntax

```
qcan [ -X ] [ -x JobNumber ] [ -P Printer ]
```

Description

The **qcan** command cancels either a particular job number or all jobs in a print queue.

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

The **qstatus** was enhanced to improve the administration of local queues showing duplicate 3-digit job numbers. You can use the **-W** flag with the **enq**, **qchk**, **lpstat**, and **lpq** status commands to display more job number digits.

If your queue display shows duplicate 3-digit job numbers, use **qchk -W** to list job numbers with greater precision. You can then cancel a specific job.

For example, **qchk** might display job number 123 twice while, **qchk -W** would display job number 1123 and 2123. If you want to cancel job number 2123, specifying **qcan -x 123**, causes the **qdaemon**

to cancel the first matching job number it finds in its internal list, which may be 1123. By having the additional information that the **-W** flag provides, you can cancel a specific job number.

Flags

Item	Description
-P <i>Printer</i>	Specifies the <i>Printer</i> where either all jobs or the selected job number will be canceled.
-x <i>JobNumber</i>	Specifies that only the job number specified by the <i>JobNumber</i> variable be canceled.
-X	Cancels all jobs or all jobs for the specified printer. If you have root user authority, all jobs on that queue are deleted. If you do not have root user authority, only jobs you submitted will be canceled. This flag is only valid for local print jobs.

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 cancel all jobs queued on printer lp0, enter:

```
qcan -X -P lp0
```

2. To cancel job number 123 on whatever printer the job is on, enter:

```
qcan -x 123
```

Files

Item	Description
/usr/sbin/qdaemon	Contains the qdaemon daemon.
/var/spool/lpd/qdir/*	Contains the job description files.
/var/spool/lpd/stat/*	Contains information on the status of the devices.
/var/spool/qdaemon/*	Contains the temporary copies of enqueued files.
/etc/qconfig	Contains the configuration file.
/etc/qconfig.bin	Contains the digested, binary version of the /etc/qconfig file.

qchk Command

Purpose

Displays the status of a print queue.

Syntax

```
qchk [ -A ] [ -L ] [ -W ] [ -P Printer ] [ -# JobNumber ] [ -q ] [ -u UserName ] [ -w Delay ]
```

Description

The **qchk** command displays the current status information regarding specified print jobs, print queues, or users. Use the appropriate flag followed by the requested name or number to indicate specific status information. If you run the **qchk** command with no flags, the status of the default queue is returned.

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

Flags

Item	Description
-# <i>JobNumber</i>	Requests the status of the job number specified by the <i>JobNumber</i> variable. The qchk command looks for <i>JobNumber</i> on the default queue when the -#JobNumber flag is used alone. To search for <i>JobNumber</i> on all queues -# flag must be used with the -A flag. The -# flag may also be used in conjunction with the -P Queue flag. Notes: <ol style="list-style-type: none">1. Specify the -P Queue to override the default destination printer.2. If jobs 1, 2, and 3 are in the printer queue, and you specify that you want the status of job 3 while job 1 is running, the status information will show job 1 and job 3, not only job 3.3. If you specify a job number that does not exist, the system displays the current job number on the queue instead of an error message.
-A	Requests the status of all queues.
-L	Displays information in a long-form mode. If the -L and -W flags are used simultaneously, the -L flag displays status of the print job in a semicolon-separated format.
-P Printer	Requests the status of the printer specified by the <i>Printer</i> variable.
-q	Requests the status of the default print queue.
-u <i>UserName</i>	Requests the status of all print jobs sent by the user specified by the <i>UserName</i> variable.
-W	Displays information in a wide-form mode with longer queue names, device names, and job numbers. Larger job number information is supported. If the -W and -L flags are used simultaneously, the -W flag displays the status of the print job in a semicolon-separated format.
-w <i>Delay</i>	Updates requested status information at intervals, in seconds, as specified by the <i>Delay</i> variable until all print jobs are finished.

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 display the default print queue, enter:

```
qchk -q
```

2. To display the long status of all queues until empty, while updating the screen every 5 seconds, enter:

```
qchk -A -L -w 5
```

3. To display the status for printer lp0, enter:

```
qchk -P lp0
```

4. To display the status for job number 123, enter:

```
qchk -# 123
```

5. To display the status of all print jobs while restricting the queue status to only printer lp0, enter:

```
qchk -A -P lp0
```

6. To display the wide status of the default print queue, enter:

```
qchk -W -q
```

Files

Item	Description
<code>/usr/sbin/qdaemon</code>	Contains the qdaemon daemon.
<code>/var/spool/lpd/qdir/*</code>	Contains the job description files.
<code>/var/spool/lpd/stat/*</code>	Contains information on the status of the devices.
<code>/var/spool/qdaemon/*</code>	Contains the temporary copies of enqueued files.
<code>/etc/qconfig</code>	Contains the configuration file.
<code>/etc/qconfig.bin</code>	Contains the digested, binary version of the <code>/etc/qconfig</code> file.

qdaemon Command

Purpose

Schedules jobs enqueued by the **enq** command.

Syntax

```
qdaemon
```

Description

The **qdaemon** command is a background process (usually started by the **startsrc** command) that schedules printing jobs enqueued by the **enq** 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.

Security

Privilege Control: Only the root user and members of the printq group should have execute (x) access to this command.

Auditing Event	Information
ENQUE_exec	Queue name, job name, host name, file name, user name

Files

Item	Description
<code>/usr/sbin/qdaemon</code>	Contains the qdaemon daemon.
<code>/var/spool/lpd/qdir/*</code>	Contains the job description files.
<code>/var/spool/lpd/pio/@local/fullmsg</code>	Contains a flag file whose existence activates qdaemon messages to contain complete information.
<code>/var/spool/lpd/stat/*</code>	Contains information on the status of the devices.
<code>/var/spool/qdaemon/*</code>	Contains the temporary copies of enqueued files.
<code>/etc/qconfig</code>	Contains the configuration file.
<code>/etc/qconfig.bin</code>	Contains the digested, binary version of the <code>/etc/qconfig</code> file.

qhld Command

Purpose

Holds and releases a spooled print job.

Syntax

```
qhld [ -r ] { -#JobNumber [ -PQueue ] | -PQueue | -uUser [ -PQueue ] }
```

Description

The **qhld** command holds print jobs in a spooled state. The job to be held is designated by job number, queue, or user name. The **-r** flag releases the hold on the print job.

The **qstatus** was enhanced to improve the administration of local queues showing duplicate 3-digit job numbers. You can use the **-W** flag with the **enq**, **qchk**, **lpstat**, and **lpq** status commands to display more job number digits.

If your queue display shows duplicate 3-digit job numbers, use **qchk -W** to list job numbers with greater precision. You can then hold a specific job.

For example, **qchk** might display job number 123 twice while, **qchk -W** would display job number 1123 and 2123. If you want to hold job number 2123, specifying **qhld -# 123**, causes the **qdaemon** to hold the first matching job number it finds in its internal list, which may be 1123. By having the additional information that the **qstatus -W** provides, you can hold a specific job number.

Flags

Item	Description
<code>-#JobNumber</code>	Specifies the print job number to be held.
<code>-PQueue</code>	Specifies the print queue to be held.
<code>-r</code>	Releases the print job by number, queue, or user name.
<code>-uUser</code>	Specifies the name of user whose print jobs are to be held.

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 hold the print job number 300, enter:

```
qhld -#300
```

2. To hold all print jobs on queue lp0, enter:

```
qhld -P lp0
```

3. To hold all jobs that belong to user fred, enter:

```
qhld -u fred
```

4. To release job number 300, enter:

```
qhld -#300 -r
```

5. To release all the jobs on queue lp0, enter:

```
qhld -Plp0 -r
```

6. To release all jobs that belong to user fred, enter:

```
qhld -u fred -r
```

Files

Item	Description
/usr/sbin/qdaemon	Contains the qdaemon daemon.
/var/spool/lpd/qdir/*	Contains the job description files.
/var/spool/lpd/stat/*	Contains information on the status of the devices.
/var/spool/qdaemon/*	Contains the temporary copies of enqueued files.
/etc/qconfig	Contains the configuration file.
/etc/qconfig.bin	Contains the digested, binary version of the /etc/qconfig file.

qmov Command

Purpose

Moves spooled print jobs to another queue.

Syntax

```
qmov -mNewQueue { -#JobNumber [ -PQueue ] | -PQueue | -uUser [ -PQueue ] }
```

Description

The **qmov** command moves spooled print jobs to another print queue. The print job to be moved is identified by job number, queue, or user name. The format of the command requires the queue where the job is to be moved to as the first argument and the name of the job to move as the second argument.

The **qstatus** was enhanced to improve the administration of local queues showing duplicate 3-digit job numbers. You can use the **-W** flag with the **enq**, **qchk**, **lpstat**, and **lpq** status commands to display more job number digits.

If your queue display shows duplicate 3-digit job numbers, use **qchk -W** to list job numbers with greater precision. You can then move a specific job.

For example, **qchk** might display job number 123 twice while, **qchk -W** would display job number 1123 and 2123. If you want to move job number 2123, specifying **qmov -# 123**, causes the **qdaemon** to move the first matching job number it finds in its internal list, which may be 1123. By having the additional information that the **qstatus -W** provides, you can move a specific job number.

Flags

Item	Description
-#JobNumber	Specifies the job number of the print job to be moved.
-mNewQueue	Specifies the name of the destination print queue.
-PQueue	Specifies the present print queue of the job to be moved.
-uUser	Specifies the name of the user whose print jobs are to be moved.

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 move job number 280 to queue lp0, enter:

```
qmov -mlp0 -#280
```

2. To move all print jobs on queue lp1 to queue lp0, enter:

```
qmov -mlp0 -Plp1
```

3. To move all of Mary's print jobs to queue lp0, enter:

```
qmov -mlp0 -u mary
```

Files

Item	Description
/usr/sbin/qdaemon	Contains the qdaemon daemon.
/var/spool/lpd/qdir/*	Contains the job description files.
/var/spool/lpd/stat/*	Contains information on the status of the devices.
/var/spool/qdaemon/*	Contains the temporary copies of enqueued files.
/etc/qconfig	Contains the configuration file.
/etc/qconfig.bin	Contains the digested, binary version of the /etc/qconfig file.

qosadd Command

Purpose

Adds a QoS (Quality of Service) Service Category or Policy Rule.

Syntax

To add a Service Category:

```
qosadd [ -s ServiceCategory] [ -t OutgoingTOS] [ -b MaxTokenBucket] [ -f FlowServiceType] [ -m MaxRate]  
service
```

To add a Policy Rule:

```
qosadd [ -s ServiceCategory] [ -r ServicePolicyRules] [ -l PolicyRulePriority] [ -n ProtocolNumber] [ -A SrcAddrRange] [ -a DestAddrRange] [ -P SrcPortRange] [ -p DestPortRange] policy
```

Description

The **qosadd** command adds the specified Service Category or Policy Rule entry in the **policyd.conf** file and installs the changes in the QoS Manager.

Flags

Flags with service add:

Item	Description
-s	The name of the ServiceCategory attribute, which is mandatory.
-t	The OutgoingTOS attribute, specified as an 8 bit binary number.
-b	The MaxTokenBucket attribute, specified in Kb (Kilobits).
-f	The FlowServiceType attribute, which is ControlledLoad or Guaranteed.
-m	The MaxRate attribute, which is specified in Kbps (Kilobits per second).

Flags with policy add:

Item	Description
-s	The name of the ServiceCategory attribute, which is mandatory.
-r	The name of the ServicePolicyRules attribute, which is mandatory.
-l	The PolicyRulePriority attribute, which is a positive integer.
-n	The ProtocolNumber attribute, which is defined in the /etc/protocols file.
-A	The SrcAddrRange attribute, which is the Source IP address range from a1 to a2 where a2 >= a1.
-a	The DestAddrRange attribute, which is the Destination IP address range from i1 to i2 where i2 >= i1.
-P	The SrcPortRange attribute, which is the Source Port range from a1 to a2 where a2 >= a1.
-p	The DestPortRange attribute, which is the Destination Port range from i1 to i2 where i2 >= i1.

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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To add the sc01 service, type:

```
qosadd -s sc01 -t 10000001 -b 81 -f ControlledLoad -m 41 service
```

2. To add the pr01 policy, type:

```
qosadd -s sc01 -r pr01 -l 2 -n 17 -A 9.3.25.1-9.3.25.10 -a 9.3.25.33-9.3.25.33  
-p 9001-9010 -P 9000-9000 policy
```

3. To add the sc02 service, type:

```
qosadd -s sc02 -t 10000001 -b 81 service
```

4. To add the pr02 policy, type:

```
qosadd -s sc02 -r pr02 -l 2 -n 17 policy
```

qoslist Command

Purpose

Lists a specific QoS (Quality of Service) Service Category or Policy Rule or lists all of them.

Syntax

To list a Service Category:

```
qoslist [ServiceCategory] service
```

To list a Policy Rule:

```
qoslist [ServicePolicyRule] policy
```

Description

The **qoslist** command lists the specified Service Category or Policy Rule. The **qoslist** command lists all Service Categories or Policy Rules if no specific name is given.

Exit Status

Item	Description
0	Successful completion

Item	Description
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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To list the sc01 service, type:

```
qoslist sc01 service
```

2. To list the the pr01 policy, type:

```
qoslist pr01 policy
```

3. To list all of the QoS ServiceCategories, type:

```
qoslist service
```

4. To list all of the QoS PolicyRules, type:

```
qoslist policy
```

qosmod Command

Purpose

Modifies an existing QoS (Quality of Service) Service Category or Policy Rule.

Syntax

To modify an existing Service Category:

```
qosmod [ -s ServiceCategory] [ -t OutgoingTOS] [ -b MaxTokenBucket] [ -f FlowServiceType] [ -m MaxRate] service
```

To modify an existing Policy Rule:

```
qosmod [ -s ServiceCategory] [ -r ServicePolicyRules] [ -l PolicyRulePriority] [ -n ProtocolNumber] [ -A SrcAddrRange] [ -a DestAddrRange] [ -P SrcPortRange] [ -p DestPortRange] policy
```

Description

The **qosmod** command modifies the specified Service Category or Policy Rule entry in the **policyd.conf** file and installs the changes in the QoS Manager.

The **qosmod** command clears out all the statistics of the old policy. When a **qosstat** command is executed immediately after **qosmod**, the user may not see all the data connections that were using the older rule shifted to the modified rule. This is because the reclassification of the data connection is delayed until a data packet arrives on that connection.

Note: Modifying the priority or filter spec of the rule only results in reclassification of the data connections which use that particular rule. Connections using other rules maintain their existing classification.

Flags

Flags with service modify:

Item	Description
-s	The name of the ServiceCategory attribute, which is mandatory.
-t	The OutgoingTOS attribute, specified as an 8-bit binary number.
-b	The MaxTokenBucket attribute, specified in Kb (Kilobits).
-f	The FlowServiceType attribute, which is ControlledLoad or Guaranteed.
-m	The MaxRate attribute, which is specified in Kbps (Kilobits per second).

Flags with policy modify:

Item	Description
-s	The name of the ServiceCategory attribute, which is mandatory.
-r	The name of the ServicePolicyRules attribute, which is mandatory.
-l	The PolicyRulePriority attribute, which is a positive integer.
-n	The ProtocolNumber attribute, which is defined in the /etc/protocols file.
-A	The SrcAddrRange attribute, which is the Source IP address range from a1 to a2, where a2 >= a1.
-a	The DestAddrRange attribute, which is the Destination IP address range from i1 to i2, where i2 >= i1.
-P	The SrcPortRange attribute, which is the Source Port range from a1 to a2, where a2 >= a1.
-p	The DestPortRange attribute, which is the Destination Port range from i1 to i2, where i2 >= i1.

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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To modify the sc01 service, type:

```
qosmod -s sc01 -t 10001100 -b 84 -f Guaranteed service
```

2. To modify the pr01 policy, type:

```
qos -s sc01 -r pr01 -l 10 -n 6 -A 9.3.25.15-9.3.25.20 -a 9.3.25.39-9.3.25.39 -p 9015-9020 policy
```

3. To modify the sc02 service, type:

```
qosmod -s sc02 -t 10001111 service
```

4. To modify the pr02 policy, type:

```
qosmod -s sc02 -r pr02 -l 13 -n 6 policy
```

qosremove Command

Purpose

Removes a QoS (Quality of Service) Service Category or Policy Rule.

Syntax

To remove a Service Category:

```
qosremove [ServiceCategory] service
```

To remove a Policy Rule:

```
qosremove [ServicePolicyRule] policy
```

To remove all the Policies and Service categories installed in the kernel:

```
qosremove all
```

Description

The **qosremove** command removes the specified Service Category or Policy Rule entry in the **policyd.conf** file and the associated policy or service in the QoS Manager.

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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To remove the sc01 service, type:

```
qosremove sc01 service
```

2. To remove the pr01 policy, type:

```
qosremove pr01 policy
```

qosstat Command

Purpose

Show Quality of Service (QoS) status.

Syntax

```
qosstat [ -A ] [ -F ] [ -S ]
```

Description

The **qosstat** command displays information about the installed QoS policies. Using **qosstat** without any flags returns filter/flow specification and statistical information for each installed policy.

Flags

Item	Description
-A	Returns the policy rule handle for each installed policy. A unique handle is assigned by the qoS manager for each policy installed.
-F	Returns the flow and filter specification for each policy installed.
-S	Returns the statistical information for each policy installed.

Examples

1. qosstat

```
Policy Rule handle 1:
Filter specification for rule index 1:
  PolicyRulePriority:          0
  protocol:                   TCP
  source IP addr:              INADDR_ANY
  destination IP addr:        INADDR_ANY
  source port:                 80
  destination port:           ANY_PORT
Flow Class for rule index 1:
  service class:              Diff-Serv
  peak rate:                  100000000 bytes/sec
  average rate:               128 bytes/sec
  bucket depth:               4096 bytes
  TOS (in profile):          0
  TOS (out profile):          0
Statistics for rule index 1:
  total number of connections: 0
  total bytes transmitted:     0
  total packets transmitted:   0
  total in-profile bytes transmitted: 0
  total in-profile packets transmitted: 0
Policy Rule Handle 2:
Filter specification for rule index 2:
  PolicyRulePriority:          0
  protocol:                   TCP
  source IP addr:              INADDR_ANY
  destination IP addr:        INADDR_ANY
  source port:                 100
  destination port:           ANY_PORT
Flow Class for rule index 2:
  service class:              Diff-Serv
  peak rate:                  100000000 bytes/sec
  average rate:               128 bytes/sec
  bucket depth:               4096 bytes
  TOS (in profile):          0
  TOS (out profile):          0
Statistics for rule index 2:
  total number of connections: 0
```

```
total bytes transmitted: 0
total packets transmitted: 0
total in-profile bytes transmitted: 0
total in-profile packets transmitted: 0
```

2. qosstat -A

```
Policy Rule Handle 1:
  rule index: 1

Policy Rule Handle 2:
  rule index: 2
```

3. qosstat -F

```
Policy Rule Handle 1:
Filter specification for rule index 1:
  PolicyRulePriority: 0
  protocol: TCP
  source IP addr: INADDR_ANY
  destination IP addr: INADDR_ANY
  source port: 80
  destination port: ANY_PORT
Flow Class for rule index 1:
  service class: Diff-Serv
  peak rate: 100000000 bytes/sec
  average rate: 128 bytes/sec
  bucket depth: 4096 bytes
  TOS (in profile): 0
  TOS (out profile): 0

Policy Rule Handle 2:
Filter specification for rule index 2:
  PolicyRulePriority: 0
  protocol: TCP
  source IP addr: INADDR_ANY
  destination IP addr: INADDR_ANY
  source port: 100
  destination port: ANY_PORT
Flow Class for rule index 2:
  service class: Diff-Serv
  peak rate: 100000000 bytes/sec
  average rate: 128 bytes/sec
  bucket depth: 4096 bytes
  TOS (in profile): 0
  TOS (out profile): 0
```

4. qosstat -S

```
Statistics for rule index 1:
  total number of connections: 0
  total bytes transmitted: 0
  total packets transmitted: 0
  total in-profile bytes transmitted: 0
  total in-profile packets transmitted: 0

Policy Rule Handle 2:
Statistics for rule index 2:
  total number of connections: 0
  total bytes transmitted: 0
  total packets transmitted: 0
  total in-profile bytes transmitted: 0
  total in-profile packets transmitted: 0
```

qpri Command

Purpose

Prioritizes a job in the print queue.

Syntax

qpri -# *JobNumber* -a *PriorityNumber*

Description

The **qpri** command prioritizes a job in a print queue by specifying the job number and giving it a priority number.

The **qpri** command works only on local print jobs and the local side of remote queues. Remote print jobs are not supported. Also, you must have root user authority or belong to the printq group to run this command.

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

The **qstatus** was enhanced to improve the administration of local queues showing duplicate 3-digit job numbers. You can use the **-W** flag with the **enq**, **qchk**, **lpstat**, and **lpq** status commands to display more job number digits.

If your queue display shows duplicate 3-digit job numbers, use **qchk -W** to list job numbers with greater precision. You can then alter the priority of a specific job.

For example, **qchk** might display job number 123 twice while, **qchk -W** would display job number 1123 and 2123. If you want to alter the priority of job number 2123, specifying **qpri -# 123**, causes the **qdaemon** to alter the priority of the first matching job number it finds in its internal list, which may be 1123. By having the additional information that the **qstatus -W** provides, you can alter the priority of a specific job number.

Flags

Item	Description
-# JobNumber	Specifies the job number on which to change priority.
-a PriorityNumber	Specifies the new priority number for the print job specified by the <i>JobNumber</i> variable. The range of priority numbers is 1 through 20, except for the root user or a member of the printq group, who can select priority numbers from 1 through 30.

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 change job number 123 to priority number 18, enter:

```
qpri -# 123 -a 18
```

Files

Item	Description
/usr/sbin/qdaemon	Contains the qdaemon daemon.
/var/spool/lpd/qdir	Contains the job description files.
/var/spool/lpd/stat	Contains information on the status of the devices.
/var/spool/qdaemon	Contains the temporary copies of enqueued files.
/etc/qconfig	Contains the configuration file.
/etc/qconfig.bin	Contains the digested, binary version of the /etc/qconfig file.

qprt Command

Purpose

Starts a print job.

Syntax

```
qprt [ -a PreviewOption ] [ -A Level ] [ -b BottomMargin ] [ -B Value ] [ -c ] [ -C ] [ -d InputDataType ] [ -D "User" ] [ -e EmphasizedOpt ] [ -E DblHigh ] [ -f Filter ] [ -F Name ] [ -g Begin ] [ -G Coord ] [ -h "Header" ] [ -H "HostName" ] [ -i Indent ] [ -I FontPath ] [ -j Init ] [ -J Restore ] [ -k Color ] [ -K Condense ] [ -l Length ] [ -L LineWrap ] [ -m Message ] [ -M MessageFile ] [ -n ] [ -N NumberCopies ] [ -O PaperHand ] [ -p Pitch ] [ -P Queue [ :QueueDevice ] ] [ -Q Value ] [ -q Quality ] [ -r ] [ -R Priority ] [ -s NameType ] [ -S Speed ] [ -t TopMargin ] [ -T "Title" ] [ -u PaperSrc ] [ -U Directional ] [ -v LinesPerIn ] [ -V Vertical ] [ -w PageWidth ] [ -W DblWide ] [ -x LineFeed ] [ -X CodePage ] [ -y DblStrike ] [ -Y Duplex ] [ -z Rotate ] [ -Z FormFeed ] [ -# { j | h | v } ] [ -= OutputBin ] { File | - } ...
```

Description

The **qprt** command creates and queues a print job to print the file specified by the *File* parameter. To print a file from standard input, specify a - (dash) instead of a file name. If you specify multiple files, then they all together make up one print job. The **qprt** command prints the files in the order you specify them.

To print a file, you must have read access to it. Using the **-r** flag you can remove a file after printing it. To remove a file, you must have write access to the directory that contains it. If you want the **qprt** command to notify you when a print job completes, specify the **-n** flag.

You can use the **-B** flag in conjunction with the **-D**, **-H**, and **-T** flags to customize burst pages. Burst pages mark the beginning, end, or both of a print job. To mark the beginning and end of a print job with burst pages, use the **-B aa** flag.

All flags are optional and you can specify them in any order. The **qprt** command ignores spaces between a flag and its argument. You can group flags without arguments after a single - (dash). All flags and their arguments must precede the *File* parameter.

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

Some of the flags and arguments listed in this command are invalid for particular printer types. If you experience problems using an option, you can use SMIT to preview a **qprt** command statement. See the *System management interface tool (SMIT) in General Programming Concepts: Writing and Debugging Programs*. Alternatively, consult your printer manual to find out what options your printer supports.

Flags

Item	Description
-#<i>{j h v}</i>	<p>Specifies a special functionality. The possible values for the <i>Value</i> variable are:</p> <ul style="list-style-type: none">j Displays a job number for the specified print job.h Queues the print job but holds it in a HELD state.v Validates the specified printer backend flag values. As part of the validation process, the command performs legality checking for illegal flag values, type checking, range checking, list checking, and other types of validation. Typically, the validation of backend flag values is useful because illegal flags are identified when the print job is submitted rather than at a later stage when the print job is processed.
-= <i>OutputBin</i>	<p>Specifies the output bin destination for a print job. If you do not specify this flag, it uses the default value from the printer driver.</p> <p>The possible values for <i>OutputBin</i> are:</p> <ul style="list-style-type: none">0 Top printer bin.1 - 49 High Capacity Output (HCO) bins 1 - 49.>49 Printer specific output bins. <p>Note: Valid output bins are printer dependent.</p>
-a <i>PreviewOption</i>	<p>Previews parameter values for a print job without actually printing any files. You can specify a 0 or a 1 for the <i>PreviewOption</i> variable. If you specify 0, the qprt command preview displays normal print processing. If you specify a 1, the command returns a list of the flag values and the filter pipeline that would be used to convert the input data type to the data type expected by the printer. These flag values are the default command line flag values from the configuration database, overridden by any flag parameters specified on the command line.</p> <p>Only flags that are valid for the <i>InputDataType</i> variable specified (or defaulted) for the -d flag are shown. Flag values related only to the spooling of your print job, instead of the actual printing, are not shown. The default values for the spooling flags are included with the flag descriptions. The flag values are not checked to verify that they are supported by the printer.</p> <p>The pipeline of filters shows the filter commands (and the flag values passed to the filter commands) that would process the data from your print files before it is sent to the printer. You can review the description for each of the filter commands to determine the type of filtering that is performed.</p>

Item**Description****-A Level**

Sets the level of diagnostic output. Diagnostic output is useful for diagnosing errors encountered by a filter pipeline that is processing a print file, a header page, or a trailer page. Diagnostic output is mailed to the user who submitted the print job. You can specify one of the following levels:

0

Discards any standard error output produced.

1

Returns flag values, the standard error output, and the complete pipeline that produced any standard error output.

2

Returns the flag values, standard error output (if any), and complete pipelines, regardless of whether an error is detected. If an error is detected, the print job is terminated.

3

Similar to a value of **2**, except that the file is not printed.

A value of **1** is recommended. A value of **0** is useful if a filter in a pipeline produces output to standard error, even if no error is encountered (for example, status information). A value of **2** or **3** is useful for diagnosing a problem, even if the problem does not cause any output to standard error.

-b BottomMargin

Specifies the bottom margin, the number of blank lines to be left at the bottom of each page.

-B Value

Prints burst pages. The *Value* variable consists of a two-character string. The first character applies to header pages. The second character applies to trailer pages. The following values are valid:

a

Always print the (header or trailer) page for each file in each print job.

n

Never print the (header or trailer) page.

g

Print the (header or trailer) page once for each print job (group of files).

For example, the **-B ga** flag prints a header page at the beginning of each print job and a trailer page after each file in each print job.

Note: In a remote print environment, the default is determined by the remote queue on the server.

-c

Copies each print file and prints from the copy. Specify this flag if you plan to modify the print file or files after the **qprt** command is issued, but before the print job completes.

If this flag is not specified and the print job is printed on the same node where it was submitted, copies of the print file or files are not made. Printing occurs directly from the file or files you specified with the *File* parameter.

Item	Description
-C	<p>Mails messages generated by your print job to you, even if you are logged in. By default, the qprt command displays messages on the console.</p> <p>The -C flag only applies to local print jobs. If you want to be notified when a job sent to a remote printer is completed, use the -n flag to receive a mail message.</p> <p style="padding-left: 40px;">Note: You cannot redirect certain messages from the qdaemon and the printer backend in any way. They are sent directly to the /dev/console file.</p>
-d InputDataType	<p>Identifies the input data type of the file or files to print. Based on the input data type and the data type expected by the printer, the print files are passed through filters (if necessary) before being sent to the printer. You can specify any of the following input data types:</p> <ul style="list-style-type: none"> a Extended ASCII c PCL d Diablo 630 g Hewlett-Packard GL p Pass-through (sent to printer unmodified) s PostScript <p>If the printer you select does not support the specified input data type, and if filters are not available to convert the data type of your print file or files to a data type supported by the printer, the print job terminates with an error message.</p>
-D "User"	<p>Labels the output for delivery to <i>User</i>. Normally the output is labeled for delivery to the user name of the person issuing the qprt command request. The value of <i>User</i> must be a single word meeting the same requirements of a regular user ID.</p>
-e EmphasizedOpt	<p>Sets emphasized print to one of the following:</p> <ul style="list-style-type: none"> + Use emphasized print. ! Do not use emphasized print.
-E DbtHigh	<p>Sets double-high print to one of the following:</p> <ul style="list-style-type: none"> + Use double-high print. ! Do not use double-high print.

Item	Description
-f <i>Filter</i>	Identifies the filter to pass your print files through before sending them to the printer. The identifiers are similar to the filter flags available with the lpr command. The available filter identifiers are p , which invokes the pr filter, n , which processes output from the troff command, and l , which allows control characters to be printed.
-F <i>Name</i>	Specifies the list of X font files containing the image of characters to be used for printing. Items in the list must be separated by commas. The <i>Name</i> parameter value can be full path names, font alias names, or XLFD names. The -F Name flag is effective only for MBCS printer queues.
-g <i>Begin</i>	Sets the page number to begin printing. This flag is recognized only if the print files are to be formatted (for example, with the -d a flag). It is not recognized for pass-through (the -d p flag), PostScript (the -d s flag), and other types of data that are already formatted.
-G <i>Coord</i>	Indicates how to print pages on laser printers that cannot print to the edge of the paper. Use one of the following for the <i>Coordinate</i> variable: <ul style="list-style-type: none"> + Whole page coordinate system ! Print page coordinate system
-h " <i>Header</i> "	Specifies the header text for use by the pr command when the -f p flag is also specified. If this flag is not specified, the pr command uses the print file name as the header. This flag is useful if you also specified the -c flag. With the -c flag, the print file name used by the pr command as the default header is the name of a temporary file generated by the spooler, instead of the file name you specified with the qprt command.
-H " <i>HostName</i> "	Sets the host name on the header page.
-i <i>Indent</i>	Indents each line the specified number of spaces. You must include the <i>Indent</i> variable in the page width specified by the -w flag.
-I <i>FontID</i>	(uppercase i) Specifies a font identifier. Specifying a font identifier overrides the pitch (the -p flag) and type style (the -s flag). The -IFontID command is effective for single byte code set print queues only.
-IFontPath	(uppercase i) Specifies the comma-separated list of font paths required for the -F flag when the font files are designated with a font alias name or an XLFD name. The <i>FontPath</i> flag is effective only for MBCS printer queues.
-j <i>Init</i>	Initializes the printer before each file is printed. You can specify any of the following: <ul style="list-style-type: none"> 0 No initialization 1 Full initialization 2 Emulator selection only

Item	Description
-J Restore	Restores the printer at the end of the print job. You can specify one of the following: <ul style="list-style-type: none"> + Restore at the end of the print job. ! Do not restore at the end of the print job.
-k Color	Specifies the print color. Typical values are black, red, blue, green, and so on. Consult your printer manual for colors supported and the ribbon position assigned to a particular color.
-K Condense	Sets condensed print to one of the following: <ul style="list-style-type: none"> + Use condensed print. ! Do not use condensed print.
-l Length	(lowercase L) Sets the page length. If the <i>Length</i> variable is 0, page length is ignored, and the output treated as one continuous page. The page length includes the top and bottom margins and indicates the printable length of the paper.
-L LineWrap	Sets line wrap for lines wider than the page width to one of the following: <ul style="list-style-type: none"> + Wrap long lines to the next line. ! Truncate long lines at the right margin.

Item	Description
-m "Message"	Displays the specified message on the console when the print job is assigned a printer and is ready to begin printing. The print job does not proceed until the message is acknowledged at the console.
-M MessageFile	Identifies a file containing message text. This text is displayed on the console when the print job is assigned a printer and is ready to begin printing. The print job does not proceed until the message is acknowledged at the console.
-n	Notifies you when the print job completes. If the -D "User" flag is also specified, the specified user is notified as well. By default, you are not notified when the print job completes.
-N NumberCopies	Specifies the number of copies to print. If this flag is not specified, one copy is printed.
-O PaperHand	Sets the type of input paper handling to one of the following: <ul style="list-style-type: none"> 1 Manual (insert one sheet at a time) 2 Continuous forms 3 Sheet feed

Item	Description
-p <i>Pitch</i>	<p>Sets the number of characters per inch. Typical values for <i>Pitch</i> are 10 and 12. The actual pitch of the characters printed is also affected by the values for the -K (condensed) flag and the -W (double-wide) flag.</p> <p>If you are printing an ASCII file on a PostScript printer, this flag determines the character point size. You can specify positive numbers greater than or equal to 1.</p>
-P <i>Queue[:QueueDevice]</i>	<p>Specifies the print queue name and the optional queue device name. If this flag is not specified, the following conditions occur:</p> <ul style="list-style-type: none"> • If the LPDEST environment variable is set, the qprt command uses the queue name specified by the LPDEST variable. If set, this value is always used, even if the PRINTER variable is also set. • If the PRINTER variable is set and no LPDEST variable is set, the qprt command uses the queue name specified by the PRINTER environment variable. Any destination command-line options override both the LPDEST and PRINTER environment variables. • If neither the LPDEST nor the PRINTER variable is set, the qprt command uses the system default queue name. (The system default queue name is the name of the first queue defined in the /etc/qconfig file.) If the <i>QueueDevice</i> variable is not specified, the first available printer configured for the queue is used. <p style="margin-left: 40px;">Note: If multiple printers are configured for the same print queue and one or more of the printers is not suitable for printing your files, you should use the <i>QueueDevice</i> variable. Otherwise, the spooler assigns the first available printer.</p>
-q <i>Quality</i>	<p>Sets the print quality to one of the following:</p> <p>0 Fast font</p> <p>1 Draft quality</p> <p>2 Near letter quality</p> <p>3 Enhanced quality</p> <p>300 300 dots per inch (dpi)</p> <p>600 600 dpi</p>
-Q <i>Value</i>	<p>Sets the paper size. The <i>Value</i> for paper size is printer-dependent. Typical values are: 1 for letter-size paper, 2 for legal, and so on. Consult your printer manual for the values assigned to specific paper sizes.</p>
-r	<p>Removes the print files after the print job completes. If this flag is not specified, the print files are not removed.</p>

Item	Description
-R <i>Priority</i>	Sets the priority for the print job. Higher values for the <i>Priority</i> variable indicate a higher priority for the print job. The default priority value is 15 . The maximum priority value is 20 for most users and 30 for users with root user privilege and members of the system group (group 0). Note: You cannot use this flag when requesting remote print jobs.
-s <i>NameType</i>	Specifies a type style with the <i>NameType</i> variable. Examples are courier and prestige. The particular type style choices differ depending on the printer type.
-S <i>Speed</i>	Sets high-speed printing to one of the following: + Use high-speed printing. ! Do not use high-speed printing.
-t <i>TopMargin</i>	Sets the top margin, the number of blank lines left at the top of each page.
-T " <i>Title</i> "	Specifies a print job title with the <i>Text</i> variable. If this flag is not specified, the first file name on the qprt command line is used as the print job title. The print job title is displayed on the header page and on responses to inquiries about queue status.
-u <i>PaperSrc</i>	Sets the paper source to one of the following: 1 Primary 2 Alternate 3 Envelopes
-U <i>Directional</i>	Sets unidirectional printing to one of the following: + Use unidirectional printing. ! Do not use unidirectional printing.
-v <i>LinesPerIn</i>	Sets the line density to a number of lines per inch. Typical values for the <i>LinesPerIn</i> variable are 6 and 8 .
-V <i>Vertical</i>	Sets vertical printing to one of the following: + Use vertical printing. ! Do not use vertical printing.
-w <i>PageWidth</i>	Sets the page width in number of characters. The page width must include the number of indentation spaces specified with the -i flag.

Item	Description
-W <i>DblWide</i>	Sets double-wide print to one of the following: + Use double-wide print. ! Do not use double-wide print.
-x <i>LineFeed</i>	Specifies automatic line feed or automatic carriage return: 0 Do not change line feeds, vertical tabs, and carriage returns. 1 Add a line feed for each carriage return. 2 Add a carriage return for each line feed and each vertical tab.
-X <i>CodePage</i>	Provides the code page name. Valid values for the <i>CodePage</i> variable are ISO8859-1 through ISO8859-9, IBM-943, IBM-eucJP, IBM-eucKR, IBM-eucTW, and UTF-8. The code page in the user's locale definition is the default.
-y <i>DblStrike</i>	Sets double-strike print to one of the following: + Use double-strike print. ! Do not use double-strike print.
-Y <i>Duplex</i>	Sets duplexed output. Duplexed output uses both the front and back of each sheet of paper for printing. You can set one of the following: 0 Simplex 1 Duplex, long edge binding 2 Duplex, short edge binding
-z <i>Rotate</i>	Rotates page printer output the number of quarter-turns clockwise as specified by the <i>Value</i> variable. The length (-l) and width (-w) values are automatically adjusted accordingly. 0 Portrait 1 Landscape right 2 Portrait upside-down 3 Landscape left

Item	Description
-Z <i>FormFeed</i>	Sends a form feed to the printer after each print file. You can specify either of the following: <ul style="list-style-type: none"> + Send a form feed command. ! Do not send a form feed command to the printer. Use this option carefully since it can result in the next print job beginning on the last output page generated by this print job. Printers printing on continuous forms cannot determine the top of the form for subsequent 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 associated with this command, see the **lssecattr** command or the **getcmdattr** subcommand.

Examples

1. To queue the `myfile` file to print on the first available printer configured for the default print queue using the default values, enter:

```
qprt myfile
```

2. To queue a file on a specific queue, to print the file without using nondefault flag values, and to validate the flag values at the time of the print job submission, enter:

```
qprt -f p -e + -P fastest -r -n -C -#v somefile
```

This command line passes the `somefile` file through the `pr` command (the `-f p` flag) and prints it using emphasized mode (the `-e +` flag) on the first available printer configured for the queue named **fastest** (the `-P fastest` flag). The `-#v` flag verifies that all flags associated with this command are valid before passing the print job to the printer backend. After the file is printed, it is removed (the `-r` flag), and the user who submitted the print job is notified (the `-n` flag) by mail (the `-C` flag) that the print job completed.

3. To print `myfile` on legal size paper, enter:

```
qprt -Q2 myfile
```

4. To enqueue the `myfile` file and return the job number, enter:

```
qprt -#j myfile
```

5. To queue `MyFile` and hold it, enter:

```
qprt -#h MyFile
```

Files

Item	Description
/etc/qconfig	Contains the queue and queue device configuration file.
/usr/bin/qprt	Contains the qprt command.

qstatus Command

Purpose

Provides printer status for the print spooling system.

Syntax

```
qstatus [ -# JobNumber ] [ -A ] [ -L ] [ -W ] [ -P Printer ] [ -e ] [ -q ] [ -u UserName ] [ -w DelaySeconds ]
```

Description

The **qstatus** command performs the actual status function for the print-spooling system. This command is never entered on the command line; it is called by the **enq** command. The **qstatus** command generates status information on specified jobs, printers, queues, or users.

The display generated by the **qstatus** command contains two entries for remote queues. The first entry contains the client's local queue and local device name and its status information. The second entry follows immediately; it contains the client's local queue name (again), followed by the remote queue name. Any jobs submitted to a remote queue are displayed first on the local side and are moved to the remote device as the job is processed on the remote machine.

Since the status commands communicate with remote machines, the status display may occasionally appear to hang while waiting for a response from the remote machine. The command will eventually time-out if a connection cannot be established between the two machines.

Flags

All flags are optional. If flags are not specified, the **qstatus** command returns the status of the following:

- The printer specified by the **LPDEST** variable, if the **LPDEST** environment variable is set. If set, this value is always used, even if the **PRINTER** variable is also set.
- The printer specified by the **PRINTER** environment variable, if the **PRINTER** variable is set and no **LPDEST** variable is set.
- The default printer, if neither the **LPDEST** nor the **PRINTER** variable is set.

Note: Any destination command line options override both the **LPDEST** and the **PRINTER** environment variables.

Item	Description
-# <i>JobNumber</i>	Displays current status information for the job specified by the <i>JobNumber</i> variable. Normally, the status of all queued jobs is displayed. <ol style="list-style-type: none">1. Specify the -P <i>Queue</i> to override the default destination printer.2. If jobs 1, 2, and 3 are in the printer queue, and you specify that you want the status of job 3 while job 1 is running, the status information will show job 1 and job 3, not only job 3.3. If you specify a job number that does not exist, the system displays the current job number on the queue instead of an error message.
-A	Displays status information on all queues defined in the /etc/qconfig file.
-e	Excludes status information from queues that are not under the control of the qdaemon command. The status from such queues may appear in different formats. The -e flag can be used with any combination of flags.
-L	Displays status information in a long and detailed version. If the -L flag and the -W flag are used simultaneously, the -L flag displays the long status of the print job in a semicolon-separated format.

Item	Description
-P Printer	Displays current status information for the printer specified by the <i>Printer</i> variable. Normally, the default printer is used, or the value of either the LPDEST or PRINTER environment variable is used. The LPDEST variable always takes precedence over the PRINTER variable.
-q	Displays the current status of the default queue. The default queue is specified by the LPDEST variable, or if a LPDEST value does not exist, by the PRINTER environment variable. If neither variable exists, the qstatus command uses the first queue listed in the <i>/etc/qconfig</i> file.
-u UserName	Displays current status information for all jobs submitted by the user specified by the <i>UserName</i> variable. Normally, the status of all queued jobs is displayed.
-W	Displays a wide version of the status information with longer queue names, device names, and job numbers. Longer job number information is supported. If the -L flag and the -W flag are used simultaneously, the -W flag displays the long status of the print job in a semicolon-separated format.
-w DelaySeconds	Displays requested queue information at intervals specified by the <i>DelaySeconds</i> variable. When the queue is empty, the display ends.

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 display the default print queue, enter:

```
qstatus -q
```

2. To display the long status of all queues until empty, while updating the screen every 5 seconds, enter:

```
qstatus -A -L -w 5
```

3. To display the status for printer lp0, enter:

```
qstatus -P lp0
```

4. To display the status for job number 123, enter:

```
qstatus -# 123 -P lp0
```

5. To display the status of all queues in wide format, enter:

```
qstatus -A -W
```

Files

Item	Description
/var/spool/lpd/qdir/*	Contains the job description files.
/etc/qconfig	Contains the configuration file.
/etc/qconfig.bin	Contains the digested, binary version of the <i>/etc/qconfig</i> file.
/usr/lib/lpd/rembak	Contains the remote back end.
/usr/lib/lpd/qstatus	Contains the command file.

Item	Description
<code>/var/spool/lpd/stat/*</code>	Contains the status files for the qstatus command.

quiz Command

Purpose

Tests your knowledge.

Syntax

```
quiz { -i File | -t | Category1 Category2 }
```

Description

The **quiz** command gives associative knowledge tests on various selectable subjects. It asks about items chosen from *Category1* and expects answers from *Category2*. If you do not specify the categories, the **quiz** command lists the available categories, provides instructions, and returns to the shell prompt.

The game provides the correct answer whenever you press the Enter key. When questions run out or when you press the Interrupt (Ctrl-C) or End Of File (Ctrl-D) key sequences, the game reports a score and ends.

Flags

Item	Description
-i <i>File</i>	Substitutes the named <i>File</i> for the standard index file. <p>Note: In the following syntax description, brackets are normally used to indicate that an item is optional. However, a bold-faced bracket or brace should be entered as a literal part of the syntax. A vertical list of items indicates that one must be chosen. The lines in <i>File</i> must have the following syntax:</p> <pre> line = category [:category] . . . category = alternate [alternate] . . . alternate = [primary] primary = character [category] option option = {category}</pre> <p>In an index file, the first category of each line must specify the name of an information file. The information file contains the names of files with quiz material. The remaining categories specify the order and contents of the data in each line of the information file. The quiz data in an information file follows the same syntax.</p> <p>A \ (backslash) is an escape character that allows you to quote syntactically significant characters or to insert a new-line character (\n) into a line. When either a question or its answer is blank, the quiz command does not ask the question. The construct a ab does not work in an information file. Use a{b}.</p>
-t	Provides a tutorial. Repeats missed questions and introduces new material gradually.

Examples

- To start a Latin-to-English quiz, enter:

```
/usr/games/quiz latin english
```

The game displays Latin words and waits for you to enter what they mean in English.

2. To start an English-to-Latin quiz, enter:

```
/usr/games/quiz english latin
```

3. To set up a Latin-English quiz, add the following line to the index file:

```
/usr/games/lib/quiz/latin:latin:english
```

This line specifies that the **/usr/games/lib/quiz/latin** file contains information about the categories Latin and English.

You can add new categories to the standard index file, **/usr/games/lib/quiz/index**, or to an index file of your own. If you create your own index file, run the **quiz** command with the **-iFile** flag and enter your list of quiz topics.

4. The following is a sample information file:

```
cor:heart
sacerdos:priest{ess}
quando:when|since|because
optat:{s}he |it |[desires|wishes]\\
desire|wish
alb[us|a|um]:white
```

This information file contains Latin and English words. The : (colon) separates each Latin word from its English equivalent. Items enclosed in { } (braces) are optional. A | (vertical bar) separates two items when entering either is correct. The [] (brackets) group items separated by vertical bars.

The first line accepts only the answer `heart` in response to the Latin word `cor`. The second accepts either `priest` or `priestess` in response to `sacerdos`. The third line accepts `when`, `since`, or `because` for `quando`.

The \ (backslash) at the end of the fourth line indicates that this entry continues on the next line. In other words, the fourth and fifth lines together form one entry. This entry accepts any of the following in response to `optat`:

```
she desires it desires desire
she wishes it wishes wish
he desires desires
he wishes wishes
```

If you start a Latin-to-English quiz, the last line of the sample information file instructs the **quiz** command to ask you the meaning of the Latin word `albus`. If you start an English-to-Latin quiz, the **quiz** command displays `white` and accepts `albus`, `alba`, or `album` for the answer.

If any of the characters { (left brace), } (right brace), [(left bracket) ,], (right bracket) or | (vertical bar) appear in a question item, the **quiz** command gives the first alternative of every | group and displays every optional group. Thus, the English-to-Latin question for the fourth definition in this sample is `she desires`.

Files

Item	Description
/usr/games/lib/quiz/index	Default index file for quiz categories.
/usr/games/lib/quiz/*	Used to specify the contents of a given file.
/usr/games	Location of the system's games.

quot Command

Purpose

Summarizes file system ownership.

Syntax

```
quot [ -c ] [ -f ] [ -h ] [ -n ] [ -v ] [ FileSystem ... ]
```

```
quot -a [ -c ] [ -f ] [ -h ] [ -n ] [ -v ]
```

Description

The **quot** command summarizes file system ownership for JFS file systems by displaying the number of 512-byte blocks currently owned by each user in the specified file system (*FileSystem*). If no file system is specified, the **quot** command displays the same information for each of the JFS file systems in the `/etc/filesystems` file.

Flags

It	Description
----	-------------

- | | |
|-----------|---|
| -a | Generate a report on all mounted systems. |
| -c | Displays a three-column report. The first column specifies the file size in 512-byte blocks. The second column specifies the number of files of that size. Finally, the third column specifies the cumulative total of 512-byte blocks in all files of that size or smaller.

Note: Files greater than or equal to 500 blocks are grouped under a block size of 499. However, their exact block count contributes to the cumulative total of blocks. |
| -f | Displays the total number of blocks, the total number of files, and the user name associated with these totals. |
| -h | Estimates the number of blocks used by the file. This estimation is based on the file size and may return greater than actual block usage when used on files with holes. |
| -n | Produces a list of all files and their owners by running the following pipeline:
<pre>ncheck filesystem sort +0n quot -n filesystem</pre> |
| -v | Displays output in three columns containing the number of blocks not accessed in the last 30, 60, and 90 days. |

Security

Access Control: This command is owned by the bin user and bin 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 display the number of files and bytes owned by each user in the `/usr` file system, enter:

```
quot -f /usr
```

The system displays the following information:

```
/usr:
63056    3217    bin
20832    390     root
1184     42      uucp
56       5       adm
8        1       guest
8        1       sys
```

2. To display file size statistics, the number of files of each size, and a cumulative total, enter:

```
quot -c /usr
```

The system displays the following information:

```
/usr:
8        103     824
16       2       856
499     0       856
```

3. To generate a report of all mounted file systems, type:

```
quot -a
```

4. To generate a report of the **/var** file system, type:

```
#quot -v /var
/var:
45695    root          12852    11878    11774
2569     guest         2567     1280     960
2121     adm           92       91       91
1343     bin           465      233     193
14      uucp          0        0        0
5       daemon        0        0        0
1       invscout      1        1        1
1       nuucp         1        1        1
1       sys           0        0        0
```

Files

Item	Description
/etc/passwd	Contains user names.
/etc/filesystems	Contains file system names and locations.

quota Command

Purpose

Displays disk usage and quotas.

Syntax

```
quota [ -u [ User ] ] [ -g [ Group ] ] [ -v | -q ]
```

Description

The quota command displays disk usage and quotas. By default, or with the **-u** flag, only user quotas are displayed. The quota command reports the quotas of all file systems listed in the **/etc/filesystems** file. If the quota command exits with a non-zero status, one or more file systems are over quota.

A root user may use the **-u** flag with the optional *User* parameter to view the limits of other users. Users without root user authority can view the limits of groups of which they are members by using the **-g** flag with the optional *Group* parameter.

Note:

1. In a JFS file system, if a particular user has no files in a file system on which that user has a quota, this command displays `quota: none` for that user. The user's actual quota is displayed when the user has files in the file system, or when the `-v` flag is specified. For JFS2, a user's actual quota is displayed in all cases.
2. In JFS2 systems, because the root user is not limited by quotas, limits for the root user are always displayed as zero (unlimited).
3. The `rpc.rquotad` protocol does not support the group quota for NFS. Thus, it does not return group quota information for NFS.

Flags

Item Description

m

- `-g` Displays the quotas of the user's group.
- `-u` Displays user quotas. This flag is the default option.
- `-v` Displays quotas on file systems with no allocated storage.
- `-q` Prints a terse message, containing only information about file systems with usage over quota.

Note: The `-q` flag takes precedence over the `-v` flag.

Security

Access Control: This command is owned by the root user and the bin group.

Privilege Control: This program is **setuid** in order to allow non-privileged users to view personal quotas.

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 display your quotas as user `keith`, type:

```
quota
```

The system displays the following information:

```
User quotas for user keith (uid 502):
Filesystem blocks quota limit grace Files quota limit grace
/u          20     55    60      20     60     65
```

2. To display quotas as the root user for user `davec`, type:

```
quota -u davec
```

The system displays the following information:

```
User quotas for user davec (uid 2702):
Filesystem blocks quota limit grace files quota limit grace
/u          48     50    60       7     60     60
```

Files

Item	Description
<code>quota.user</code>	Specifies user quotas.
<code>quota.group</code>	Specifies group quotas.
<code>/etc/filesystems</code>	Contains file system names and locations.

quotacheck Command

Purpose

Checks file system quota consistency.

Syntax

```
quotacheck [ -d ] [ -g ] [ -u ] [ -v ] { -a | Filesystem ... }
```

Description

The `quotacheck` command examines a file system specified by the *FileSystem* parameter, builds a table of current disk usage, and compares the information in the table to that recorded in the file system's disk quota files. If any inconsistencies are detected, the quota files are updated. By default, both user and group quotas are checked.

The optional `-g` flag specifies that only group quotas are checked. The optional `-u` flag specifies that only user quotas are checked. Specifying both `-g` and `-u` flags is equivalent to the default behavior which checks both user and group quotas. The `-a` flag specifies that all file systems in the `/etc/filesystem` file with disk quotas enabled are checked.

For both JFS and JFS2 file systems, the optional `-d` flag deletes Usage statistics for any user or group ID that does not exist in `/etc/passwd` or `/etc/group`, and which has no allocation in the file system. The affected users or groups will no longer have statistics displayed by the `repquota` command.

The `quotacheck` command normally operates silently. If the `-v` flag is specified, the `quotacheck` command reports discrepancies between the calculated and recorded disk quotas.

For JFS, the `quotacheck` command determines the quota file names from the `/etc/filesystems` file (by default, the files are named `quota.user` and `quota.group` and are located at the root of the file system); for JFS2, the names and location of these files are predetermined and cannot be changed. If these files do not exist, the `quotacheck` command creates them.

Note: Do not run the `quotacheck` command against an active file system. If the file system has any current activity, running `quotacheck` may result in incorrect disk usage information.

Flags

Item	Description
<code>-a</code>	Checks all file systems with disk quotas enabled in <code>/etc/filesystems</code> .
<code>-d</code>	Deletes Usage statistics for undefined IDs with no allocation (both JFS and JFS2).
<code>-g</code>	Checks group quotas only.
<code>-u</code>	Checks user quotas only.
<code>-v</code>	Reports discrepancies between the calculated and recorded disk quotas.

Security

Access Control: Only a user with root user authority 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

1. To check the user and group quotas in the `/usr` file system, type:

```
quotacheck /usr
```

2. To check only the group quotas in the `/usr` file system, type:

```
quotacheck -g /usr
```

Files

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

quotaon or quotaoff Command

Purpose

Turns on and off file system quotas.

Syntax

```
quotaon [ -g ] [ -u ] [ -v ] { -a | FileSystem ... }
```

```
quotaoff [ -g ] [ -u ] [ -v ] { -a | FileSystem ... }
```

Description

The `quotaon` command enables disk quotas for one or more file systems specified by the *FileSystem* parameter. The specified file system must be defined with quotas in the `/etc/filesystems` file, and must be mounted. The `quotaon` command looks for the `quota.user` and `quota.group` files in the root directory of the associated file system, and will return an error if not found.

Note: For JFS only, the default quota file names (`quota.user` and `quota.group`) may be overridden in the `/etc/filesystems` file. The quota files can be external to the quota enabled file system by specifying full paths in the `/etc/filesystems` file. For JFS2 file systems, the file names may not be overridden and must reside in the root directory of the file system.

By default, both user and group quotas are enabled. The `-u` flag enables only user quotas; the `-g` flag enables only group quotas. Specifying both `-g` and `-u` flags is equivalent to the default (no option specified). The `-a` flag specifies that all file systems with disk quotas, as indicated by the `/etc/filesystems` file, are enabled.

The `quotaoff` command disables disk quotas for one or more file systems. By default, both user and group quotas are disabled. The `-a`, `-g`, and `-u` flags operate as with the `quotaon` command. The `-v` flag prints a message for each quota type (user or group) in every file system in which quotas are turned on or off with the `quotaon` and `quotaoff` commands, respectively.

An error (**EPERM**) will be returned if the `quota.user` and `quota.group` files are not owned by user **root** and group **system**. Ownership changes on these files are not permitted while quotas are active.

Flags

Item	Description
------	-------------

- | | |
|-----------|---|
| -a | Enables or disables all file systems that are read-write and have disk quotas, as indicated by the <code>/etc/filesystems</code> file. When used with the -g flag, only group quotas in the <code>/etc/filesystems</code> file are enabled or disabled; when used with the -u flag, only user quotas in the <code>/etc/filesystems</code> file are enabled or disabled. |
| -g | Specifies that only group quotas are enabled or disabled. |
| -u | Specifies that only user quotas are enabled or disabled. |
| -v | Prints a message for each file system in which quotas are turned on or off. |

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

1. To enable user quotas for the `/usr` file system, enter:

```
quotaon -u /usr
```

2. To disable user and group quotas for all file systems in the `/etc/filesystems` file and print a message, enter:

```
quotaoff -v -a
```

Files

Item	Description
<code>quota.user</code>	Specifies user quotas.
<code>quota.group</code>	Specifies group quotas.
<code>/etc/filesystems</code>	Contains file system names and locations.

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

- /etc/filesystems file
 - listing entries
 - using lsfs command [245](#)
- /etc/inittab file
 - listing records
 - using lsitab command [250](#)
- /etc/vfs file
 - listing entries
 - using lsvfs command [399](#)

A

- accounting system
 - creating system
 - creating basic files
 - using nulladm command [1136](#)
 - displaying information about last commands run [5](#)
 - displaying previous logins and logouts [2](#)
 - displaying session records [1320](#)
 - formatting ASCII report of previous day
 - using prdaily command [1321](#)
 - formatting total accounting records
 - using prtacct command [1378](#)
 - performing periodic accounting
 - using monacct command [756](#)
 - printer records
 - preparing [1177](#)
 - updating last log in records [6](#)
- acct/* commands
 - lastlogin [6](#)
 - monacct [756](#)
 - nulladm [1136](#)
 - prctmp [1320](#)
 - prdaily [1321](#)
 - prtacct [1378](#)
- alias database
 - building [920](#)
- appearance
 - general
 - using mwm command [828](#)
- Appearance resource set
 - using mwm command [828](#)
- asynchronous connection
 - PPP subsystem
 - pppdial command [1312](#)
- attributes
 - displaying characteristics of
 - using Isattr command [186](#)
 - displaying possible values of
 - using Isattr command [186](#)
 - of exported symbols [34](#)

B

- behavior
 - general

- behavior (*continued*)
 - general (*continued*)
 - using mwm command [828](#)
- Behavior resource set
 - using mwm command [828](#)
- bibliography
 - finding references in
 - using lookbib command [89](#)
- binding
 - button
 - using mwm command [863](#)
 - key
 - using mwm command [864](#)
- button
 - binding
 - using mwm command [863](#)

C

- C programming language
 - checking source code for problems [53](#)
- Client-Specific resource set
 - using mwm command [847](#)
- command
 - mwm [814](#)
 - running without hangups [1101](#)
- commands
 - event response resource manager (ERRM)
 - logevent [77](#)
 - event-response resource manager (ERRM)
 - lsevent [239](#)
 - logevent [77](#)
 - lphistory [140](#)
 - lppchk [145](#)
 - lsassocmap [184](#)
 - lsattr [186](#), [679](#)
 - lsaudrec [191](#)
 - lscomg [211](#)
 - lscondition [214](#)
 - lscondresp [219](#)
 - lscons [226](#)
 - lsevent [239](#)
 - lslpclacl [258](#)
 - lslpcmd [263](#)
 - lslpp [267](#)
 - lslpracl [272](#)
 - lslpriacl [278](#)
 - lslprsacl [283](#)
 - lslv [288](#)
 - lsmcode [293](#)
 - lsresponse [335](#)
 - lsrpdomain [343](#)
 - lsrpnodc [346](#)
 - lsrsrc [352](#)
 - lsrsrcassoc [358](#)
 - lssensor [378](#)
 - lssmbcmnt [388](#)

commands (*continued*)

- lssmbcred [389](#)
- lsvsd [413](#)
- lswpar [420](#)
- mail [454](#)
- Mail [454](#)
- mailq [469](#)
- mailx [454](#)
- make [472](#)
- mkboot [518](#)
- mkcfsmnt [530](#)
- mkcimreg [535](#)
- mkclient [541](#)
- mkcomg [546](#)
- mkcondition [551](#)
- mkfs [578](#)
- mkitab [592](#)
- mklpcmd [600](#)
- mklv [604](#)
- mknfs [619](#)
- mknfsmnt [624](#)
- mkprtsv [647](#)
- mkresponse [658](#)
- mkrole [663](#)
- mkrpdomain [665](#)
- mkrsrc [674](#)
- mksensor [691](#)
- mksmbcmnt [699](#)
- mksmbcred [702](#)
- mkssys [703](#)
- mkstr [706](#)
- mkuser [722](#)
- mkuser.sys [726](#)
- mkvg [728](#)
- mkwpar [738](#)
- namerslv [873](#)
- netstat [905](#)
- newform [920](#)
- nfso [936](#)
- nim [953](#)
- nimconfig [1005](#)
- niminit [1013](#)
- nlssrc [1039](#)
- no [1070](#)
- ntpq4 [1130](#)
- oslevel [1170](#)
- parameter type [936](#)
- passwd [1186](#)
- pic [1227](#)
- pkgtrans [1282](#)
- pr [1317](#)
- preparevsd [1322](#)
- preprnode [1323](#)
- printf [1327](#)
- prs [1374](#)
- psroff [1413](#)
- qdaemon [1453](#)
- quot [1479](#)
- quota [1480](#)
- quotacheck [1482](#)
- quotaoff [1483](#)
- quotaon [1483](#)
- setting a different priority for
using nice command [951](#)

commands (*continued*)

- tunable parameters [936](#)
- commands, mtrace [801](#)
- Component Appearance resource set
using mwm command [822](#)
- computer languages
 - C
 - lexical analyzer [45](#)

D

- daemons
 - lockd [74](#)
 - monitord [759](#)
 - mountd [785](#)
 - mrouted [794](#)
 - ndpd-router [880](#)
 - nfsrgyd [945](#)
 - pppattachd [1304](#)
- default window menu
 - using mwm command
using mwm command [816](#)
- delayed login ports [1214](#)
- description file
 - creating
 - using makedev command [483](#)
 - using mwm command [853](#)
- device
 - adding to the system
 - using mkdev command [561](#)
- Device Configuration Database
 - listing acceptable devices from
using Isparent command [313](#)
- devices
 - displaying characteristics of
using lsdev command [230](#)
- Diablo 630 print file
 - converting to PostScript
using ps630 command [1405](#)
- diff listings [1191](#)
- directories
 - creating [564](#)
 - creating a hierarchy [566](#)
 - moving [808](#), [812](#)
 - path name of
 - displaying [1437](#)
 - renaming [812](#)
- directory
 - creating a lost and found
 - using mklost+found command [599](#)
 - mounting
 - using mount command [768](#)
- displays
 - files
 - formatting to screen [1223](#)
 - one screen at a time [760](#)
 - listing currently available on the system
using lsdisp command [237](#)
- document
 - typesetting
 - using mmt command [751](#)
- drivers
 - formatting a printer
using pioformat command [1252](#)

dynamic host configuration protocol
update the DNS server
nsupdate command [1110](#)

E

echo request
sending to a network host
using ping command [1237](#)
enotifyevent Command [1103](#)
enotifyevent script [1103](#)
environment
printing variable values [1327](#)
ERRM
event information
logging [77](#)
ERRM commands
logevent [77](#)
lsevent [239](#)
ERRM scripts
logevent [77](#)
event information
logging [77](#)
event response resource manager (ERRM)
commands
logevent [77](#)
event information
logging [77](#)
scripts
logevent [77](#)
event specification
using mwm command [862](#)
event-response resource manager (ERRM)
commands
lsevent [239](#)

F

file
checking the pathname
using pathchk command [1196](#)
creating a special
using mknod command [630](#)
finding lines in a sorted
using look command [88](#)
resource description
using mwm command [853](#)
file system
constructing a prototype
using mkproto command [639](#)
constructing a prototype file
using proto command [1372](#)
making available for use
using mount command [768](#)
files
archive [9](#)
compressing
using pack command [1178](#)
creating links [67](#)
displaying
one screen at a time [760](#)
displaying contents [1223](#)
displaying in specified format [1149](#)

files (*continued*)
formatting to the display [1223](#)
generating path names from i-node numbers
using ncheck command [876](#)
import and export [10](#)
moving [808](#)
numbering lines [1036](#)
unpacking
using pcat command [1213](#)
writing to standard output
using pr command [1317](#)
flags
ignored and unsupported [11](#)
folders
printing full path names of
using mhpath command [503](#)
font code
adding to the system
using mkfont command [576](#)
font files
creating fonts.dir file
using mkfontdir command [577](#)
font set
changing
using managefonts command [492](#)
updating
using managefonts command [492](#)
fonts
copying
using piofontin command [1251](#)
listing the fonts available for use
using lsfont command [244](#)
format
changing text
using newform command [920](#)
Frame Component resource set
using mwm command [825](#)

G

games
number-guessing game [760](#)
number-writing game [1137](#)
quiz [1477](#)
garbage collection
by using ld command [11](#)
group ID
changing the primary
using newgrp command [923](#)
groups
creating new groups
using mkgroup command [583](#)
displaying attributes of
using lsgroup command [246](#)

H

HCON
System/370 Host Interface Adapter
diagnosing activity [1185](#)
help
using files, editors, and macros [42](#)
hlplearn [42](#)

hlpleave [43](#)
hpline [51](#)
hlplint [53](#)
hlpln [67](#)
hlplorder [92](#)
hlpmkvgdata [734](#)

I

icon
 box menu
 using mwm command [818](#)
 window menu
 using mwm command [817](#)
Icon Component resource set
 using mwm command [825](#)
iconifying window
 using mwm command [817](#)
idnls [69](#)
IMAP commands
 pop3d [1298](#)
 pop3ds [1299](#)
import and export file format
 -bI: and -bE: Flags [32](#)
index
 creating subject-page index
 using ndx command [885](#)
 generating permuted
 using ptx command [1427](#)
input extension records
 listing
 using listX11input command [61](#)
input focus
 keyboard
 using mwm command [819](#)
installable packages,
 producing
 using pkgmk command [1275](#)
installp format
 creating software packages [589](#)
internet
 querying domain name servers [1108](#)
IPv6 neighbor discovery protocol [878](#)
Isallqdev command [182](#)
Isconn command [224](#)
Isdev command [230](#)
ISO 2022 [431](#)
Isparent command
 listing acceptable connection types from
 using Isparent command [313](#)

K

key
 binding
 using mwm command [864](#)
keyboard maps
 listing
 using lskbd command [251](#)

L

labck [1](#)

last command [2](#)
lastcomm command [5](#)
lastlogin command [6](#)
ld command
 archive files [9](#)
 attributes of exported symbols [34](#)
 garbage collection [11](#)
 ignored and unsupported flags [11](#)
 import and export file format [32](#)
 import and export files [10](#)
 libraries [10](#)
 linking mode [9](#)
 options (-bOptions) [15](#)
 processing [10](#)
 runtime linking [31](#)
 symbols [11](#)
ldd command [39](#)
learn command [42](#)
leave command [43](#)
lex command
 definitions [46](#)
 rules [47](#)
 specification file [46](#)
lexical analyzer [45](#)
libraries
 by using ld command [10](#)
 checking for incompatibilities [53](#)
line command [51](#)
line printer
 formatting text for
 using nroff command [1105](#)
 generating a ripple pattern for a
 using lptest command [168](#)
 sending requests to
 using lp command [93](#)
link command [52](#)
link subroutine [52](#)
linking mode
 using ld command [9](#)
links
 hard [67](#)
 symbolic [67](#)
lint command [53](#)
listdgrp command [58](#)
listing
 software products
 using lspp command [267](#)
listvgbackup command [59](#)
listX11input command [61](#)
Live Update
 lvupdateInit command [438](#)
 lvupdateRegKE command [440](#)
 lvupdateRegScript command [441](#)
 lvupdateSafeKE command [443](#)
 lvupdateSetProcs command [445](#)
ln command [67](#)
locale command
 writes information about locales [69](#)
localedef command
 Processes locale and character map files [71](#)
lock command [73](#)
lockd daemon [74](#)
locktrace command
 controls kernel lock tracing [76](#)

- logevent command [77](#)
- logevent script [77](#)
- logform command [78](#)
- logger command [80](#)
- logical volume
 - adding mirrors to
 - using mklvcopy command [613](#)
 - mirroring
 - using mirrorvg command [509](#)
- login command [81](#)
- login name
 - displaying the current process
 - using logname command [87](#)
- login ports
 - delayed
 - enabling [1214](#)
 - listing [1214](#)
 - disabling
 - using pdisable command [1215](#)
 - using phold command [1226](#)
 - enabling
 - using pdelay command [1214](#)
 - using penable command [1221](#)
 - using pshare command [1409](#)
 - using pstart command [1416](#)
 - listing
 - using pdelay command [1214](#)
 - using penable command [1221](#)
 - using phold command [1226](#)
 - using pstart command [1416](#)
 - shared
 - enabling [1409](#)
 - listing [1409](#)
- logins command [84](#)
- logname command [87](#)
- logoffs
 - displaying all previous [2](#)
- logout command [88](#)
- look command [88](#)
- lookbib command [89](#)
- loopmount command [90](#)
- loopumount command [91](#)
- lorder command [92](#)
- lpacl information [101](#)
- lpar_netboot command [114](#)
- lpd command [129](#)
- lphistory command [140](#)
- lppchk command [145](#)
- lppmgr command [148](#)
- lptest command [168](#)
- ls-secldapclntd [176](#)
- lsallq command [181](#)
- lsarm command [183](#)
- lsassocmap command [184](#)
- lsattr command [186](#)
- lsaudrec command [191](#)
- lsauthent command
 - authentication methods [198](#)
- lsC2admin command [199](#)
- lsCCadmin command [200](#)
- lscfg command [200](#)
- lscifscrd command [203](#)
- lscifsmnt command [204](#)
- lsclass command [205](#)
- lscluster
 - configuration [207](#)
- lscomg command [211](#)
- lscondition command [214](#)
- lscondresp command [219](#)
- lscons command [226](#)
- lscore [227](#)
- lscosi command [228](#)
- lsdisp command [237](#)
- lsevent command [239](#)
- lsfilt command
 - listing filter rules [243](#)
- lsfont command [244](#)
- lsfs command [245](#)
- lsgroup command [246](#)
- lsitab command [250](#)
- lskbd command [251](#)
- lsldap command [253](#)
- lslicense command
 - fixed and floating licenses
 - listing number and status of [257](#)
- lslpclacl command [258](#)
- lslpcmd command [263](#)
- lslpp command [267](#)
- lslpriacl command [272](#)
- lslpriacl command [278](#)
- lslprsacl command [283](#)
- lslv command [288](#)
- lsmaster command [292](#)
- lsmcode command [293](#)
- lsmksysb command [295](#)
- lsmp command [298](#)
- lsmpio command [299](#)
- lsnamsv command [307](#)
- lsnfsexp command [308](#)
- lsnfmnt command [309](#)
- lsnim command [310](#)
- lsnlspath command [313](#)
- lspath command [315](#)
- lspprc command [328](#)
- lsprtsv command [321](#)
- lsps command [322](#)
- lspv command [324](#)
- lsque command [330](#)
- lsquedev
 - command [331](#)
- lsquedev command [331](#)
- lsresource command
 - displaying bus resources [332](#)
- lsresponse command [335](#)
- lsrole command [340](#)
- lsrpdomain command [343](#)
- lsrpnod command [346](#)
- lsrset command [350](#)
- lsrsrc command [352](#)
- lsrsrcassoc command [358](#)
- lssavevg command [366](#)
- lssavewpar command [369](#)
- lssec
 - listing attributes of
 - security stanza files [371](#)
- lssecattr command [373](#)
- lssecmode [376](#)
- lssensor command [378](#)

- lssmbcmnt command [388](#)
- lssmbcred command [389](#)
- lssrad command [390](#)
- lssrc command [391](#)
- lsts command [394](#)
- lstun command
 - listing tunnel definitions [395](#)
- lsuser command [396](#)
- lsusil command [399](#)
- lsvfs command [399](#)
- lsvg command [400](#)
- lsvgfs command [404](#)
- lsvirprt command [405](#)
- lsvmode command
 - displaying current video mode [408](#)
- lsvpd command [409](#)
- lsvsd command [413](#)
- lswlmconf command [416](#)
- lswpar command [420](#)
- luit command [431](#)
- lvmo command [66](#), [433](#)
- lvmostat [435](#)
- lvupdateInit command [438](#)
- lvupdateRegKE command [440](#)
- lvupdateRegScript command [441](#)
- lvupdateSafeKE command [443](#)
- lvupdateSetProcs command [445](#)

M

- m4 command
 - preprocessing files [447](#)
- mach command [451](#)
- machstat command [452](#)
- macref command
 - producing cross-reference listing of [453](#)
- macro file [453](#)
- mail
 - sending [502](#)
- mail command [454](#)
- Mail command [454](#)
- Mail commands
 - mailq [469](#)
 - mailstats [471](#)
 - newaliases [920](#)
- mail traffic statistics
 - displaying [471](#)
- mailq command [469](#)
- mailstats command [471](#)
- mailx command [454](#)
- make command [472](#)
- makedbm command [480](#)
- makedepend command [481](#)
- makedev command [483](#)
- man command [486](#)
- man pages
 - displaying information online [486](#)
- manage_disk_drivers command [491](#)
- managefonts command [492](#)
- mant command [494](#)
- manual
 - typesetting pages of
 - using mant command [494](#)
- mark command [496](#)

- mathematical text
 - formatting
 - using neqn command [886](#)
- memorandum macro [749](#)
- memory management
 - displaying system page size [1183](#)
- menu pane
 - window manager
 - using mwm command [864](#)
- mesg command [498](#)
- message
 - showing the previous
 - using prev command [1325](#)
- message facility commands
 - mkcatdefs [521](#)
- message routing [1303](#)
- message sequences
 - creating
 - using mark command [496](#)
 - using pick command [1233](#)
 - displaying
 - using mark command [496](#)
 - modifying
 - using mark command [496](#)
 - using pick command [1233](#)
- message source file
 - preprocessing [521](#)
- messages
 - checking for
 - using msgchk command [797](#)
 - permitting
 - using mesg command [498](#)
 - printing full path names of
 - using mhpath command [503](#)
 - producing formatted lists of
 - using mhl command [499](#)
 - refusing
 - using mesg command [498](#)
 - selecting by content
 - using pick command [1233](#)
 - showing the next
 - using next command [926](#)
 - writing to standard output [925](#)
- MH
 - post command [1303](#)
 - prompter command [1370](#)
- MH shell
 - creating
 - using msh command [798](#)
- mhl command [499](#)
- mhmail command [502](#)
- mhpath command [503](#)
- migratelp [505](#)
- migratepv command [506](#)
- migwpar command [507](#)
- mirror pools [298](#), [324](#)
- mirrorvg command [509](#)
- mirscan command [512](#)
- mkboot command [518](#)
- mkC2admin command [520](#)
- mkcatdefs command [521](#)
- mkCCadmin command [522](#)
- mkcd command [523](#)
- mkcfsmnt command [530](#)

- mkcifscred command [531](#)
- mkcifsmnt command [533](#)
- mkcimreg command [535](#)
- mkclass command [538](#)
- mkclient command [541](#)
- mkcomg command [546](#)
- mkcondition command [551](#)
- mkcosi command [560](#)
- mkdev command [561](#)
- mkdir command [564](#)
- mkdirhier command [566](#)
- mkdom command [566](#)
- mkdvd command [568](#)
- mkfifo command
 - making FIFO special files [574](#)
- mkfilt command
 - activating or deactivating filter rules [575](#)
- mkfont command [576](#)
- mkfontdir command [577](#)
- mkfs command [578](#)
- mkgroup command
 - description of [583](#)
- mkhosts command [586](#)
- mkiba command [587](#)
- mkinstallp command [589](#)
- mkkitab command [592](#)
- mkkeyserv command [594](#)
- mklost+found command [599](#)
- mklpcmd command [600](#)
- mklv command [604](#)
- mklvcopy command [613](#)
- mkmaster command [616](#)
- mknamsv command [617](#)
- mknetid command [618](#)
- mknfs command [619](#)
- mknfsexp command [620](#)
- mknfsmnt command [624](#)
- mknfscopy command [628](#)
- mknod command [630](#)
- mkpasswd command [633](#)
- mkpath command [635](#)
- mkprojldap command [637](#)
- mkproto command [639](#)
- mkprtldap command [644](#)
- mkprtsv command [647](#)
- mkps command [650](#)
- mkque command [653](#)
- mkquedev command [654](#)
- mkramdisk command [656](#)
- mkresponse command [658](#)
- mkrole command [663](#)
- mkrpdomain command [665](#)
- mkrsset [673](#)
- mkrsrc command [674](#)
- mkrtc command [679](#)
- mksecpki command [689](#)
- mk sensori command [691](#)
- mkslave command [698](#)
- mk smbcmnt command [699](#)
- mk smbcred command [702](#)
- mkssys command [703](#)
- mkstr command [706](#)
- mk sysb command [707](#)
- mk sysb_iso command [712](#)
- mk szfile command [715](#)
- mk tcpip command [717](#)
- mkts command [719](#)
- mkuser command [722](#)
- mkuser.sys command [726](#)
- mkusil command [727](#)
- mkvg command [728](#)
- mkvgdata command [734](#)
- mkvirprt command [735](#)
- mkwpar command [738](#)
- mkwpar data command [748](#)
- mm command
 - printing document formatted with using mm command [749](#)
- mmt command [751](#)
- mmtu command [753](#)
- mobip6ctrl commnad [754](#)
- mobip6reqd daemon [756](#)
- monacct command [756](#)
- monitord daemon [759](#)
- mount command [768](#)
- mountd daemon [785](#)
- mounting
 - automatic
 - using mount command [769](#)
- mpstat [788](#)
- mrouted daemon [794](#)
- msgchk command [797](#)
- msh command [798](#)
- mtrace command [801](#)
- multibos command [804](#)
- multicast path
 - from a source to a receiver, printing a using mtrace command [801](#)
- MultiPath I/O
 - lsmpio command [299](#)
 - lspath command [315](#)
 - mkpath command [635](#)
- multiple path I/O (MPIO) [491](#)
- mvt command [813](#)
- MWM [814](#)
- mwm command [814](#)

N

- named daemon
 - description of [867](#)
- named-checkconf [867](#)
- named-checkzone [868](#)
- named-compilezone [868](#)
- named9 Daemon [870](#)
- namerslv command [873](#)
- ncheck command [876](#)
- nddctl command [877](#)
- NDP and RIPng daemon
 - for a router
 - using ndpd-router daemon [880](#)
- ndp Command [878](#)
- ndp daemon [878](#)
- ndpd-host [879](#)
- ndpd-router daemon [880](#)
- ndx command [885](#)
- neighbor discovery protocol [878](#)
- neqn command [886](#)

- netcd daemon [887](#)
- netcdctrl Command [889](#)
- netpmon Command [891](#)
- netrule command [900](#)
- netstat command
 - interface display [905](#)
 - routing table display [905](#)
- network CPU usage [891](#)
- Network Install Management operations
 - using nim command [953](#)
- Network Install Manager [310](#), [1001](#), [1005](#)
- network parameters
 - tuning
 - using no command [1070](#)
- newaliases command
 - Mail [920](#)
- newform command [920](#)
- newgrp command [923](#)
- newkey command
 - NIS [924](#)
- next command [926](#)
- NFS commands
 - mknfs [619](#)
 - mknfsexp [620](#)
 - mknfmnt [624](#)
 - nfsstat [946](#)
 - on [1161](#)
- NFS daemons
 - lockd [74](#)
 - mountd [785](#)
 - nfsd [932](#)
 - portmap [1300](#)
- nfs.clean command [928](#)
- nfs4cl command [929](#)
- nfs4smctl [931](#)
- nfsauthreset [931](#)
- nfsd daemon [932](#)
- nfshostkey [934](#)
- nfshostmap [935](#)
- nfso command [936](#)
- nfsrgyd [945](#)
- nfsstat command [946](#)
- nice command [951](#)
- nim command [953](#)
- NIM commands
 - lsnim [310](#)
 - nim [953](#)
 - nim_clients_setup [969](#)
 - nim_master_recover [970](#)
 - nim_master_setup [973](#)
 - nim_update_all [986](#)
 - nimadapters [988](#)
 - nimclient [1001](#)
 - nimconfig [1005](#)
 - niminit [1013](#)
- NIM objects
 - performing operations
 - using nim command [953](#)
- nim_clients_setup [969](#)
- nim_master_recover [970](#)
- nim_master_setup [973](#)
- nim_move_up command [976](#)
- nim_update_all [986](#)

- nimadapters [988](#)
- nimadm command [994](#)
- nimclient command [1001](#)
- nimconfig command [1005](#)
- nimdef command [1008](#)
- niminit command [1013](#)
- niminv command [1017](#)
- nimol_backup command [1023](#)
- nimol_config command [1024](#)
- nimol_install command [1027](#)
- nimol_lspp command [1029](#)
- nimol_update command [1031](#)
- nimquery [1032](#)
- NIS commands
 - lsmaster [292](#)
 - lsnfsexp [308](#)
 - lsnfmnt [309](#)
 - makedbm [480](#)
 - mkclient [541](#)
 - mkkeyserv [594](#)
 - mkmaster [616](#)
 - mkslave [698](#)
 - newkey [924](#)
- nistoldif [1034](#)
- NLSPATH,
 - secure system configuration variable, showing the value of [313](#)
- nlsrc command [1039](#)
- nm Command
 - object files
 - displaying symbol tables [1041](#)
- no command [1070](#)
- nohup command [1101](#)
- notify object class (SRC)
 - adding a notify method definition [632](#)
- notifyevent Command [1103](#)
- notifyevent script [1103](#)
- nroff command
 - formatting mathematical text for using neqn command [886](#)
- nslookup command [1108](#)
- nsupdate command [1110](#)
- ntp-keygen4 [1126](#), [1135](#)
- ntp-keygen4 Command [1126](#), [1135](#)
- ntpd4 daemon [1114](#)
- ntpd4 [1117](#), [1411](#)
- ntpd4 [1119](#)
- ntpd4 Command [1119](#)
- ntpq4 daemon [1130](#)
- nulladm command [1136](#)

O

- object files
 - displaying profile data [1359](#)
 - reading one or more [92](#)
- objects
 - shared [9](#)
- ODM
 - compacts database [1250](#)
- ODM (Object Data Manager)
 - object classes
 - adding objects to [1154](#)
 - changing objects in [1155](#)

ODM (Object Data Manager) *(continued)*

object classes *(continued)*

- compiling [1156](#)
- deleting [1158](#)
- displaying description [1161](#)
- removing objects [1158](#)
- retrieving objects matching criteria [1159](#)

objects

- adding to object classes [1154](#)
- changing in object classes [1155](#)
- removing from object classes [1158](#)
- retrieving matching criteria from object classes [1159](#)

on command [1161](#)

openpts [1162](#)

openpts Command [1162](#)

options (-bOptions)

- in ld command [15](#)

OS_install command [1164](#)

oslevel command [1170](#)

ospf_monitor Command [1172](#)

output, standard

- writing to
 - using pr command [1317](#)

P

pac command [1177](#)

pack command [1178](#)

packages,

- producing installable
 - using pkgmk command [1275](#)

packf command [1180](#)

pagdel [1182](#)

pagesize Command [1183](#)

paging space

- adding an additional
 - using mkps command [650](#)
- displaying the characteristics of
 - using lsps command [322](#)

paginit [1184](#)

paglist [1184](#)

panel 20 Command [1185](#)

passwd command [1186](#)

password

- changing
 - using passwd command [1186](#)
- reserving a terminal
 - using lock command [73](#)
- verifying information in the user database about a
 - using pwdck command [1440](#)

passwords

- administering user
 - using pwdadm command [1438](#)

paste command [1189](#)

patch command [1191](#)

path name

- displaying [1437](#)

pathchk command [1196](#)

pax command

- extracts, writes, and lists archives and copies files [1197](#)

pcat command [1213](#)

pdelay command [1214](#)

pdisable command [1215](#)

pdlink [1216](#)

pdmkdir [1217](#)

pdmode [1218](#)

pdrmdir [1219](#)

pdset [1220](#)

penable command [1221](#)

perfwb command [1222](#)

permuted index

generating

- using ptx command [1427](#)

personal printer data stream

- formatting text for printers with
 - using proff command [1361](#)

phold command [1226](#)

physical volume

- moving partitions from one volume to another volume
 - using migratepv command [506](#)

pic command [1227](#)

ping command

- description of [1237](#)

piobe command [1244](#)

pioburst command [1246](#)

piocnvt command [1242](#), [1247](#)

pidigest command [1248](#)

piofontin command [1251](#)

pioformat command [1252](#)

piofquote command [1254](#)

pioout command [1263](#)

piopredef command [1265](#)

pkgadd Command [1267](#)

pkgask Command [1269](#)

pkgchk Command [1272](#)

pkginfo Command [1273](#)

pkgmk command [1275](#)

pkgparam Command [1278](#)

pkgproto Command [1279](#)

pkgrm Command [1281](#)

pkgtrans command [1282](#)

platform_dump [1284](#)

plot file

- converting to PostScript
 - using psplot command [1410](#)

plotgbe Command [1286](#)

plotlbe Command [1287](#)

plotter queue

- listing device names with a queue
 - using lsallqdev command [182](#)

plotting HP-GL files [1286](#), [1287](#)

pmctl command [1288](#)

pmcycles Command [1294](#)

pmlist command [1295](#)

pmtu command [1297](#)

policy

- keyboard input
 - using mwm command [819](#)

pop3d daemon [1298](#)

pop3ds daemon [1299](#)

port

- stopping all processes on a
 - using logout command [88](#)

port 515

- monitoring for print requests
 - using lpd command [129](#)

portmap daemon [1300](#)

- portmir command [1301](#)
- ports
 - login [1215](#)
- post command [1303](#)
- PostScript file
 - converting from Diablo 630 file
 - using ps630 command [1405](#)
 - converting from Tektronix 4014 file
 - using ps4014 command [1404](#)
 - converting from troff
 - using psroff command [1413](#)
 - converting from troff intermediate file
 - using psc command [1406](#)
 - using psdit command [1406](#)
 - reverse page order for printing
 - using psrev command [1412](#)
 - select page range for printing
 - using psrev command [1412](#)
- PostScript printers
 - converting certain control characters for
 - using piofquote command [1254](#)
- PPP [1307](#)
- ppp subsystem
 - startup and management
 - pppcontrold daemon [1307](#)
- PPP subsystem
 - asynchronous connection [1312](#)
 - display RAS information
 - pppstat command [1313](#)
- pppattachd daemon [1304](#)
- pppcontrold daemon [1307](#)
- pppdial command [1312](#)
- pppstat command [1313](#)
- pprof Command [1315](#)
- pr command [1317](#)
- praliases command [1320](#)
- prctmp command [1320](#)
- prdaily command [1321](#)
- preparevsd command [1322](#)
- preprnode command [1323](#)
- prev command [1325](#)
- Print Job Manager
 - printing a file
 - using pioout command [1263](#)
 - processing a print job
 - using piobe command [1244](#)
- print queue
 - prioritizing a job in the
 - using qpri command [1463](#)
- print queues
 - creating new [1260](#)
- print server
 - providing on a network
 - using lpd command [129](#)
- printer
 - burst pages
 - pioburst command [1246](#)
 - cancelling a job on the
 - using qcan command [1450](#)
 - creating a predefined data stream definition
 - using piopredef command [1265](#)
 - fonts
 - copying [1251](#)
 - line
 - printer (*continued*)
 - line (*continued*)
 - sending requests to a [93](#)
 - providing the status of the spooling system
 - using qstatus command [1475](#)
 - starting a job on the
 - using qprt command [1465](#)
 - printer backend
 - send message to [1262](#)
 - printer backend commands
 - piodmng [1250](#)
 - piolsvp [1254](#)
 - piomgpdev [1257](#)
 - piomkapqd [1258](#)
 - piomkqpq [1260](#)
 - piomsg [1262](#)
 - printer definition
 - for virtual printer [1248](#)
 - printer definition file
 - expanding and contracting [1247](#)
 - printer formatter
 - driving
 - using pioformat command [1252](#)
 - printer queue
 - adding to the system
 - using mkque command [653](#)
 - using mkqudev command [654](#)
 - displaying the status of a
 - using qchk command [1451](#)
 - listing device names with a queue
 - using lsallqdev command [182](#)
 - printer spooling system
 - performing system administration for the
 - using qadm command [1449](#)
 - printers
 - adding new [1260](#)
 - printf command [1327](#)
 - printing
 - holding a job [1454](#)
 - moving a spooled job [1455](#)
 - printing a multicast path
 - from a source to a receiver
 - using mtrace command [801](#)
 - probevctrl command [1332](#)
 - probevue command [1336](#)
 - proccred command [1340](#)
 - processes
 - stopping all port
 - using logout command [88](#)
 - procfiles command [1341](#)
 - procflags command [1343](#)
 - procldd command [1344](#)
 - procmmap command [1345](#)
 - procrun command [1349](#)
 - procsig command [1350](#)
 - procstack command [1351](#)
 - proctree command [1353](#)
 - proctree command [1354](#)
 - procwait command [1357](#)
 - procwdx command [1358](#)
 - proff command [1361](#)
 - program products
 - listing optional
 - using ls1pp command [267](#)

- programs
 - creating make file dependencies [481](#)
- projctl command [1362](#)
- prompter command [1370](#)
- prompting editor, invocation of, MH [1370](#)
- proto command [1372](#)
- proxymngr Command [1372](#)
- prs command [1374](#)
- prtacct command [1378](#)
- prtconf [1379](#)
- prtgblconfig command [1383](#)
- ps4014 command [1404](#)
- ps630 command [1405](#)
- psc command [1406](#)
- psdit command [1406](#)
- pseudo-device printers [1257](#)
- pshare command [1409](#)
- psplot command [1410](#)
- psrev command [1412](#)
- psroff command [1413](#)
- pstart command [1416](#)
- pstat command [1417](#)
- ptpd daemon [1418](#)
- ptsc [1425](#)
- ptsc Command [1425](#)
- ptsevt [1426](#)
- ptsevt Command [1426](#)
- ptsevtd [1427](#)
- ptsevtd Command [1427](#)
- ptx command [1427](#)
- pvi command [1431](#)
- pwchange command [1434](#)
- pwck command [1436](#)
- pwdadm command [1438](#)
- pwdck command [1440](#)
- pwtokey command [1443](#)
- pxed Command [1446](#)

Q

- qadm command [1449](#)
- qcan command [1450](#)
- qchk command [1451](#)
- qdaemon command [1453](#)
- qhld command [1454](#)
- qmov command [1455](#)
- qosadd Command [1457](#)
- qoslist Command [1458](#)
- qosmod Command [1459](#)
- qosremove Command [1461](#)
- qosstat Command [1462](#)
- qpri command [1463](#)
- qprt command [1465](#)
- qstatus command [1475](#)
- queue
 - displaying the stanza name of
 - using lsque command [330](#)
- queues
 - holding a spooled job [1454](#)
 - moving a spooled job [1455](#)
- quot command [1479](#)
- quota command [1480](#)
- quotacheck command [1482](#)
- quotaoff command [1483](#)

- quotaon command [1483](#)

R

- RAM disk [656](#)
- read operations
 - copying from standard input [51](#)
- remount option [774](#)
- resource
 - client-specific
 - using mwm command [846](#)
- resource description file
 - using mwm command [853](#)
- resource set
 - Client-Specific [847](#)
 - Component Appearance [822](#)
 - Frame and Icon Component [825](#)
 - General Appearance and Behavior [828](#)
- RIPng and NDP daemon
 - for a router
 - using ndpd-router daemon [880](#)
- router
 - NDP and RIPng daemon for
 - using ndpd-router daemon [880](#)
- runtime linking
 - by using ld command [31](#)

S

- SCCS commands
 - prs [1374](#)
- screens
 - displaying text one screen at a time [760](#)
- scripts
 - enotifyevent [1103](#)
 - event response resource manager (ERRM)
 - logevent [77](#)
 - logevent [77](#)
 - notifyevent [1103](#)
- session
 - initiating a user
 - using login command [81](#)
- shared login ports [1409](#)
- shared objects
 - using ld command [9](#)
- slide
 - typesetting
 - using mvt command [813](#)
- SMIT
 - building printer dialogs [1258](#)
 - Creating print queues with [1258](#)
 - Creating printers with [1258](#)
- source files [1191](#)
- SRC
 - adding a definition to subsystem object class
 - mkssys command examples [705](#)
 - adding a subserver object
 - mkserver command example [698](#)
 - adding a subsystem notification method
 - mknotify command examples [632](#)
- SRC configuration commands
 - mknotify [632](#)
 - mkserver [697](#)

- SRC configuration commands (*continued*)
 - mkssys [703](#)
 - standard input
 - reading [51](#)
 - standard output
 - writing [51](#)
 - subserver
 - getting status
 - using lssrc command [391](#)
 - subservers
 - adding a definition to SRC object class
 - mkserver command example [698](#)
 - subsystem
 - getting status
 - using lssrc command [391](#)
 - subsystems
 - adding a definition to SRC object class
 - mkssys command examples [705](#)
 - adding a notification method
 - mknotify command examples [632](#)
 - system
 - adding a device to
 - using mkdev command [561](#)
 - system resource controller [632](#), [697](#), [703](#)
 - system tables
 - interpreting the contents of [1417](#)
 - System/370 Host Interface Adapter
 - checking for proper installation [1185](#)

T

- TCP/IP
 - daemons
 - named [867](#)
 - hosts
 - generating table file [586](#)
 - setting required values [717](#)
 - name service
 - displaying information [307](#)
 - name services
 - configuring [617](#)
 - network options
 - displaying, adding, and deleting [753](#)
 - parameters
 - tuning [1070](#)
 - print services
 - displaying information about [321](#)
 - querying internet domain name servers [1108](#)
- TCP/IP commands
 - lsnamsv [307](#)
 - lsprtsv [321](#)
 - mkhosts [586](#)
 - mknamsv [617](#)
 - mkprtsv [647](#)
 - mktcpip [717](#)
 - mmtu [753](#)
 - namerslv [873](#)
 - netstat [905](#)
 - no [1070](#)
 - nslookup [1108](#)
- TCP/IP smit commands
 - lsnamsv [307](#)
 - lsprtsv [321](#)
 - mknamsv [617](#)

- TCP/IP smit commands (*continued*)
 - mkprtsv [647](#)
 - mktcpip [717](#)
 - namerslv [873](#)
- Tektronix 4014 file
 - converting to PostScript
 - using ps4014 command [1404](#)
- terminal
 - reserving [73](#)
- text
 - changing format of
 - using newform command [920](#)
- time management
 - creating reminders [43](#)
- troff file
 - converting to PostScript
 - using psroff command [1413](#)
- troff intermediate file format
 - converting to PostScript format
 - using psc command [1406](#)
 - using psdit command [1406](#)
- tuning
 - network parameters
 - using no command [1070](#)
- typesetting document
 - using mmt command [751](#)

U

- Unicode [431](#)
- user
 - changing a password
 - using passwd command [1186](#)
 - initiating a session for a
 - using login command [81](#)
 - verifying password information about a
 - using pwdck command [1440](#)
- user database
 - creating a version for searches
 - using mkpasswd command [633](#)
- users
 - administering passwords for
 - using pwdadm command [1438](#)
 - displaying attributes
 - using lsuser command [396](#)
 - displaying role attributes
 - using lsrole command [340](#)

V

- view graph
 - typesetting
 - using mvt command [813](#)
- virtual memory
 - displaying the system page size [1183](#)
- virtual printer
 - displaying the attribute values of
 - using lsvirprt command [405](#)
 - displaying the attribute values of using lsvirprt
 - command [1242](#)
 - making
 - using mkvirprt [735](#)
- virtual printers [1248](#), [1254](#)

- virtual RAM disk
 - making [656](#)
- volume group
 - displaying information about
 - using lsvg command [400](#)
 - displaying information about a physical volume
 - using lspv command [324](#)

W

- window
 - default components [816](#)
 - default menu
 - using mwm command [816](#)
 - iconifying
 - using mwm command [817](#)
 - menu pane
 - using mwm command [864](#)
- window manager
 - event specification
 - using mwm command [862](#)
- Workload Manager (WLM)
 - listing configurations [416](#)
 - lswlmconf command [416](#)

X

- x defaults [820](#)

