AIX Version 6.1

Commands Reference, Volume 1, a - c

SC23-5243-02

AIX Version 6.1

Commands Reference, Volume 1, a - c

SC23-5243-02

Note
Before using this information and the product it supports, read the information in “Notices,” on page 705.

Third Edition (October 2009)

This edition applies to AIX 5L Version 5.3 and to all subsequent releases of this product until otherwise indicated in new editions.

A reader’s comment form is provided at the back of this publication. If the form has been removed, address comments to Information Development, Department 04XA-905-6B013, 11501 Burnet Road, Austin, Texas 78758-3400. To send comments electronically, use this commercial Internet address: pserinfo@us.ibm.com. Any information that you supply may be used without incurring any obligation to you.

© Copyright International Business Machines Corporation 1997, 2009.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About This Book ix
How to Use This Book ix
ISO 9000 xi
32-Bit and 64-Bit Support for the Single UNIX Specification xii
Related Information xii

Alphabetical Listing of Commands 1
ac Command 1
accept, reject Command 2
acctcms Command 2
acctcom Command 4
acctcon1 or acctcon2 Command 8
acctctl Command 10
acctdisk or acctdusg Command 14
acctmerg Command 16
acctprc1, acctprc2, or accton Command 19
acctrpt Command 20
acctwtmp Command 26
aclconvert Command 27
acledit Command 28
aclget Command 30
aclgettypes Command 31
aclput Command 32
adb Command 34
addbib Command 36
addrpnode Command 38
addX11input Command 40
adfutil Command 40
admin Command (SCCS) 42
aixmibd Daemon 47
aixpert Command 49
aixpertldap Command 51
aixterm Command 53
ali Command 90
alias Command 91
alog Command 93
alstat Command 96
alt_disk_copy Command 97
alt_disk_install Command 101
alt_disk_mksysb Command 108
alt_rootvg_op Command 112
amepat Command 115
anno Command 121
ap Command 124
apply Command 125
apropos Command 126
ar Command 127
arithmetic Command 130
arp Command 132
artexdiff Command 135
artexget Command 136
artexlist Command 138
artexmerge Command 139

© Copyright IBM Corp. 1997, 2009

artexremset Command . . . 140
artexset Command . . . 142
as Command . . . 144
asa or fpr Command . . . 148
at Command . . . 149
ate Command . . . 154
atmstat Command . . . 166
atq Command . . . 168
atrm Command . . . 169
attachrset Command . . . 170
audit Command . . . 171
auditbin Daemon . . . 175
auditcat Command . . . 177
auditconv Command . . . 178
auditmerge Command . . . 180
auditpr Command . . . 181
auditselect Command . . . 183
auditstream Command . . . 188
authexec Command . . . 190
authrpt Command . . . 191
authqry Command . . . 193
autoconf6 Command . . . 194
automount Daemon . . . 195
automountd Daemon . . . 197
autopush Command . . . 198
awk Command . . . 200
back Command . . . 216
backsnap Command . . . 217
backup Command . . . 218
banner Command . . . 224
basename Command . . . 224
batch Command . . . 226
battery Command . . . 227
bc Command . . . 228
bdftopcf Command . . . 240
bdiff Command . . . 241
bellmail Command . . . 242
bffcreate Command . . . 245
bfs Command . . . 248
bg Command . . . 251
bicheck Command . . . 253
biff Command . . . 253
bindintcpu Command . . . 255
bindprocessor Command . . . 256
binld Daemon . . . 258
biod Daemon . . . 259
bj Command . . . 260
bootauth Command . . . 261
bootlist Command . . . 261
bootparamd Daemon . . . 266
bootpd Daemon . . . 266
bootptodhcp Command . . . 268
bosboot Command . . . 269
bosdebug Command . . . 272
bs Command . . . 274
bsh Command . . . 283

bterm command . . . 285
bugfiler Command . . . 288
burst Command . . . 291
cachefslog Command . . . 293
cachefsstat Command . . . 294
cachefswssize Command . . . 295
cal Command . . . 296
calendar Command . . . 297
cancel Command . . . 299
canonls Command . . . 302
captoinfo Command . . . 303
capture Command . . . 304
cat Command . . . 305
catman Command . . . 307
cb Command . . . 309
cd Command . . . 309
cdc Command . . . 311
cdcheck Command . . . 313
cdeject Command . . . 315
cdmount Command . . . 316
cdromd Command . . . 317
cdumount Command . . . 318
cdutil Command . . . 319
certadd Command . . . 320
certcreate Command . . . 322
certdelete Command . . . 324
certget Command . . . 325
certlink Command . . . 326
certlist Command . . . 328
certrevoke Command . . . 330
certverify Command . . . 332
cfgif Method . . . 333
cfginet Method . . . 334
cfgmgr Command . . . 335
cfgqos Method . . . 338
cfgvsd Command . . . 339
cflow Command . . . 340
cfsadmin Command . . . 342
chargefee Command . . . 345
chauth Command . . . 345
chauthent Command . . . 347
chC2admin Command . . . 348
chCCadmin Command . . . 349
chcifscred Command . . . 350
chcifsmnt Command . . . 351
chclass Command . . . 352
chcod Command . . . 356
chcomg Command . . . 357
chcondition Command . . . 361
chcons Command . . . 365
chcore Command . . . 367
chcosi Command . . . 368
chdev Command . . . 370
chdisp Command . . . 373
checkeq or checkmm Command . . . 374
checknr Command . . . 374

chfilt Command 375
chfn Command 377
chfont Command 379
chfs Command 380
chgif Method 388
chginet Method 390
chgroup Command 391
chgrp Command 394
chgrpmem Command 396
chhwkbd Command 398
chiscsi Command 399
chitab Command 401
chkbd Command 403
chkey Command 404
chlang Command 404
chlicense Command 407
chlpclacl Command 408
chlpcmd Command 412
chlpracl Command 415
chlpriacl Command 420
chlprsacl Command 424
chlv Command 429
chlvcopy Command 433
chmaster Command 434
chmod Command 436
chmp Command 439
chnamsv Command 441
chndaf Command 443
chnlspath Command 445
chnfs Command 446
chnfsdom Command 448
chnfsexp Command 448
chnfsim Command 452
chnfsmnt Command 456
chnfsrtd Command 458
chnfssec Command 459
chown Command 461
chpasswd Command 462
chpath Command 464
chprtsv Command 466
chps Command 469
chpv Command 471
chque Command 473
chquedev Command 474
chresponse Command 476
chrole Command 480
chroot Command 482
chrsrc Command 484
chsec Command 488
chsensor Command 491
chserver Command 494
chservices Command 495
chsh Command 497
chslave Command 499
chssys Command 500
chsubserver Command 503

chtcb Command 505
chtun Command 506
chtz Command 509
chuser Command 509
chusil Command 521
chvfs Command 521
chvg Command. 522
chvirprt Command. 527
chvmode Command 528
chwpar Command. 529
chypdom Command 536
ckauth Command 537
ckfilt Command. 537
ckpacct Command 540
ckprereq Command 541
cksum Command 543
clear Command 544
clogin Command 545
clsnmp Command. 546
cmp Command 552
col Command 554
colcrt Command 555
colrm Command 556
comb Command (SCCS) 557
comm Command 558
command Command. 560
comp Command 562
compare_report Command 564
compress Command. 567
comsat Daemon 569
configassist Command 570
conflict Command. 570
confsetcntrl Command 571
confsrc Command. 576
cp Command 578
cpcosi Command 582
cpio Command 583
cplv Command 592
cpp Command 594
cpupstat Command 598
craps Command 599
createvsd Command. 600
crfs Command 606
cron Daemon 611
cronadm Command 613
crontab Command. 614
crvfs Command. 618
csh Command 619
csmstat Command 621
csplit Command 623
csum Command 625
ct Command. 627
ctaclfck Command. 629
ctags Command 633
ctcasd Daemon. 635
ctctrl Command. 636

cthactrl Command 642
cthagsctrl Command 643
cthagstune Command 646
cthatsctrl Command 647
cthatstune Command 650
ctlvsd Command 652
ctmsskf Command 654
ctscachgen Command 657
ctsidmck Command 660
ctskeygen Command. 663
ctsnap Command 665
ctsthl Command 667
ctsvhbac Command 670
ctsvhbal Command 674
ctsvhbar Command 677
cu Command 680
curt Command 684
custom Command. 693
cut Command 699
cw or checkcw Command 701
cxref Command 703

Appendix. Notices 705
Trademarks 706

Index 709

About This Book

This book provides end users with complete detailed information about commands for the AIX operating system. The commands are listed alphabetically and by category, and complete descriptions are given for commands and their available flags. If applicable, each command listing contains examples. This volume contains AIX commands that begin with the letters a through c. This publication is also available on the documentation CD that is shipped with the operating system.

How to Use This Book

A command is a request to perform an operation or run a program. You use commands to tell the operating system what task you want it to perform. When commands are entered, they are deciphered by a command interpreter (also known as a shell) and that task is processed.

Some commands can be entered simply by typing one word. It is also possible to combine commands so that the output from one command becomes the input for another command. This is known as *pipelining*.

Flags further define the actions of commands. A flag is a modifier used with the command name on the command line, usually preceded by a dash.

Commands can also be grouped together and stored in a file. These are known as shell procedures or shell scripts. Instead of executing the commands individually, you execute the file that contains the commands.

Some commands can be constructed using Web-based System Manager applications or the System Management Interface Tool (SMIT).

Highlighting

The following highlighting conventions are used in this book:

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.
Italics	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.

Format

Each command may include any of the following sections:

Purpose	A description of the major function of each command.
Syntax	A syntax statement showing command line options.
Description	A discussion of the command describing in detail its function and use.
Flags	A list of command line flags and associated variables with an explanation of how the flags modify the action of the command.
Parameters	A list of command line parameters and their descriptions.
Subcommands	A list of subcommands (for interactive commands) that explains their use.
Exit Status	A description of the exit values the command returns.
Security	Specifies any permissions needed to run the command.
Examples	Specific examples of how you can use the command.
Files	A list of files used by the command.
Related Information	A list of related commands in this book and related discussions in other books.

© Copyright IBM Corp. 1997, 2009

Reading Syntax Statements

Syntax statements are a way to represent command syntax and consist of symbols such as brackets ([ ]), braces ({ }), and vertical bars (|). The following is a sample of a syntax statement for the **unget** command:

unget [-**r***SID*] [-**s**] [-**n**] *File* ...

The following conventions are used in the command syntax statements:

- Items that must be entered literally on the command line are in **bold**. These items include the command name, flags, and literal charactors.
- Items representing variables that must be replaced by a name are in *italics*. These items include parameters that follow flags and parameters that the command reads, such as *Files* and *Directories*.
- Parameters enclosed in brackets are optional.
- Parameters enclosed in braces are required.
- Parameters not enclosed in either brackets or braces are required.
- A vertical bar signifies that you choose only one parameter. For example, [a | b] indicates that you *can* choose a, b, or nothing. Similarly, { a | b } indicates that you *must* choose either a or b.
- Ellipses (...) signify the parameter can be repeated on the command line.
- The dash (-) represents standard input.

Listing of Installable Software Packages

To list the installable software package (fileset) of an individual command use the **lslpp** command with the **-w** flag. For example, to list the fileset that owns the **installp** command, enter:

```
lslpp -w /usr/sbin/installp
```

Output similar to the following displays:

```
File                            Fileset                 Type
-----------------------------------------------------------------------
/usr/sbin/installp                bos.rte.install           File
```

To list the fileset that owns all file names that contain `installp`, enter:

```
lslpp -w "*installp*"
```

Output similar to the following displays:

```
File                            Fileset                 Type
-----------------------------------------------------------------------
/usr/sbin/installp                bos.rte.install           File
/usr/clvm/sbin/linstallpv         prpq.clvm                 File
/usr/lpp/bos.sysmgt/nim/methods/c_installp
                                  bos.sysmgt.nim.client     File
```

Running Commands in the Background

If you are going to run a command that takes a long time to process, you can specify that the command run in the background. Background processing is a useful way to run programs that process slowly. To run a command in the background, you use the & operator at the end of the command:

```
Command&
```

Once the process is running in the background, you can continue to work and enter other commands on your system.

At times, you might want to run a command at a specified time or on a specific date. Using the **cron** daemon, you can schedule commands to run automatically. Or, using the **at** and **batch** commands, you can run commands at a later time or when the system load level permits.

Entering Commands

You typically enter commands following the shell prompt on the command line. The shell prompt can vary. In the following examples, $ is the prompt.

To display a list of the contents of your current directory, you would type `ls` and press the Enter key:

```
$ ls
```

When you enter a command and it is running, the operating system does not display the shell prompt. When the command completes its action, the system displays the prompt again. This indicates that you can enter another command.

The general format for entering commands is:

Command Flag(s) Parameter

The flag alters the way a command works. Many commands have several flags. For example, if you type the **-l** (long) flag following the **ls** command, the system provides additional information about the contents of the current directory. The following example shows how to use the **-l** flag with the **ls** command:

```
$ ls -l
```

A parameter consists of a string of characters that follows a command or a flag. It specifies data, such as the name of a file or directory, or values. In the following example, the directory named **/usr/bin** is a parameter:

```
$ ls -l /usr/bin
```

When entering commands, it is important to remember the following:

- Commands are usually entered in lowercase.
- Flags are usually prefixed with a - (minus sign).
- More than one command can be typed on the command line if the commands are separated by a **;** (semicolon).
- Long sequences of commands can be continued on the next line by using the \ (backslash). The backslash is placed at the end of the first line. The following example shows the placement of the backslash:

  ```
  $ cat /usr/ust/mydir/mydata > \
  /usr/usts/yourdir/yourdata
  ```

When certain commands are entered, the shell prompt changes. Because some commands are actually programs (such as the **telnet** command), the prompt changes when you are operating within the command. Any command that you issue within a program is known as a subcommand. When you exit the program, the prompt returns to your shell prompt.

The operating system can operate with different shells (for example, Bourne, C, or Korn) and the commands that you enter are interpreted by the shell. Therefore, you must know what shell you are using so that you can enter the commands in the correct format.

Stopping Commands

If you enter a command and then decide to stop that command from running, you can halt the command from processing any further. To stop a command from processing, press the Interrupt key sequence (usually Ctrl-C or Alt-Pause). When the process is stopped, your shell prompt returns and you can then enter another command.

ISO 9000

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

32-Bit and 64-Bit Support for the Single UNIX Specification

Beginning with Version 5.2, the operating system is designed to support The Open Group's Single UNIX Specification Version 3 (UNIX 03) for portability of UNIX-based operating systems. Many new interfaces, and some current ones, have been added or enhanced to meet this specification, making Version 5.2 even more open and portable for applications, while remaining compatible with previous releases of AIX.
To determine the proper way to develop a UNIX 03-portable application, you may need to refer to The Open Group's UNIX 03 specification, which can be accessed online or downloaded from http://www.unix.org/ .

Related Information

The following books contain information about or related to commands:

- *AIX 5L Version 5.3 Commands Reference, Volume 2*
- *AIX 5L Version 5.3 Commands Reference, Volume 3*
- *AIX 5L Version 5.3 Commands Reference, Volume 4*
- *AIX 5L Version 5.3 Commands Reference, Volume 5*
- *AIX 5L Version 5.3 Commands Reference, Volume 6*
- *AIX 5L Version 5.3 Files Reference*
- *Printers and printing*
- *Installation and migration*
- *AIX 5L Version 5.3 AIX Installation in a Partitioned Environment*
- *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*
- *Performance management*
- *AIX 5L Version 5.3 Performance Tools Guide and Reference*
- *Security*
- *Networks and communication management*
- *Operating system and device management*
- *AIX 5L Version 5.3 Technical Reference: Base Operating System and Extensions Volume 1*
- *AIX 5L Version 5.3 Technical Reference: Base Operating System and Extensions Volume 2*
- *AIX 5L Version 5.3 Technical Reference: Communications Volume 1*
- *AIX 5L Version 5.3 Technical Reference: Communications Volume 2*
- *AIX 5L Version 5.3 Technical Reference: Kernel and Subsystems Volume 1*
- *AIX 5L Version 5.3 Technical Reference: Kernel and Subsystems Volume 2*
- *AIX 5L Version 5.3 Web-based System Manager Administration Guide*
- *Performance Toolbox Version 2 and 3 for AIX: Guide and Reference*

Alphabetical Listing of Commands

ac Command

Purpose

Prints connect-time records.

Syntax

/usr/sbin/acct/ac [**-d**] [**-p**] [**-w** *File*] [*User* ...]

Description

The **ac** command prints the total connect time for all users or the connect time for specified users. Records are based on who logged in during the life of the current **wtmp** data file.

Connect-time records are created by the **init** and the **login** programs and are collected in the **/var/adm/wtmp** file, if that file exists. The root user or a member of the **adm** group should create the **/var/adm/wtmp** file with an initial record length of 0 (zero). Records should be processed periodically to keep the file from becoming too full. If the file has not been created, the following error message is returned:

```
No /var/adm/wtmp
```

If the file becomes too full, additional **wtmp** files are created. These files can be printed, if specified with the **-w** flag.

Flags

-d	Creates a printout for each day, from midnight to midnight.
-p	Prints connect-time totals by individual login. Without this flag, a total for the time period is printed.
-w *File*	Specifies a **wtmp** file other than the **/var/adm/wtmp** file.

Security

Access Control: This command should grant execute (x) access to all users.

Examples

1. To obtain a printout of the connect time for all users who logged in during the life of the current **wtmp** data file, enter:

   ```
   /usr/sbin/acct/ac
   ```

2. To obtain a printout of the total connect time for users `smith` and `jones`, as recorded in the current **wtmp** data file, enter:

   ```
   /usr/sbin/acct/ac smith jones
   ```

3. To obtain a printout of the connect-time subtotals for users `smith` and `jones`, as recorded in the current **wtmp** data file, enter:

   ```
   /usr/sbin/acct/ac  -p smith jones
   ```

Files

/usr/sbin/acct/ac	Contains the **ac** command.
/var/adm/wtmp	Contains the active data file for the collection of connect-time records.

© Copyright IBM Corp. 1997, 2009

Related Information

The **init** and **login** commands.

For more information about the Accounting System, the preparation of daily and monthly reports, and the accounting files, see System accounting in *Operating system and device management*.

Setting up an accounting subsystem in *Operating system and device management* explains the steps you must take to establish an accounting system.

---

accept, reject Command

Purpose

Accepts/rejects print requests.

Syntax

accept *Destinations*

reject [**-r** *Reason*] *Destination*

Description

The **accept** command allows the queuing of print requests for the named *Destinations*. A *Destination* can be either a printer or a class of printers. To find out the status of a destination, run **lpstat -a** command.

The **reject** command prevents queuing of print requests for the named *destinations*. A *destination* can be either a printer or a class of printers. To find out the status of a destination, run **lpstat -a** command.

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

Flags

-r *Reason* Assigns a *Reason* for rejection of requests. The *Reason* applies to all of the specified *Destination*s. The **lpstat -a** command reports the reason. If it contains blanks, *Reason* must be enclosed in quotes. The default reason is `unknown reason` for existing destinations, and `new destination` for destinations just added to the system but not yet accepting requests.

Security

RBAC Disclaimer

Files

/var/spool/lp/*

Related Information

The **enable** command, **lpadmin** command, and **lpsched** command.

---

acctcms Command

Purpose

Produces command-usage summaries from accounting records.

Syntax

/usr/sbin/acct/acctcms [**-t** | **-a** [**-o**] [**-p**]] [**-c**] [**-j**] [**-n**] [**-s**] [*File ...*]

Description

The **acctcms** command reads each file specified by the *File* parameter, adds and sorts all records for identically named processes, and writes the records to standard output. By default, the output file is in binary format. Input files are usually in the **acct** file format.

When you use the **-o** and **-p** flags together, the **acctcms** command produces a report that combines prime and nonprime time. Prime and nonprime times are defined by entries in the **/etc/acct/holidays** file. Prime times are assumed to be the period when the system is most active, such as weekdays. Saturdays and Sundays are always nonprime time for the accounting systems, as are any holidays that you specify in the **/etc/acct/holidays** file. All the output summaries are of total usage, except for number of times run, CPU minutes, and real minutes, which are split into prime and nonprime minutes.

Flags

-a Displays output in ASCII summary format rather than binary summary format. Each output line contains the command name, the number of times the command was run, total kcore time (memory measurement in kilobyte segments), total CPU time, total real time, mean memory size (in K-bytes), mean CPU time per invocation of the command, and the CPU usage factor. The listed times are all in minutes. The **acctcms** command normally sorts its output by total kcore minutes. The unit kcore minutes is a measure of the amount of memory used (in kilobytes) multiplied by the amount of time it was in use. This flag cannot be used with the **-t** flag.

Use the following options only with the -a option:

-o Displays a command summary of non-prime time commands.

-p Displays a command summary of prime time commands.

When you use the **-o** and **-p** flags together, the **acctcms** command produces a report that combines prime and non-prime time. Prime and non-prime times are defined by entries in the **/etc/acct/holidays** file. Prime times are assumed to be the period when the system is most active, such as weekdays. Saturdays and Sundays are always non-prime time for the accounting systems, as are any holidays that you specify in the **/etc/acct/holidays** file. All the output summaries are of total usage, except for number of times run, CPU minutes, and real minutes, which are split into prime and non-prime minutes.

The default items have the following headings in the output:

```
         TOTAL COMMAND SUMMARY


COMMAND  NUMBER  TOTAL     TOTAL    TOTAL
NAME     CMDS    KCOREMIN  CPU-MIN  REAL-MIN


MEAN    MEAN     HOG     CHARS   BLOCKS
SIZE-K  CPU-MIN  FACTOR  TRNSFD  READ
```

-c Sorts by total CPU time rather than total kcore minutes. When this flag is used with the **-n** flag, only the **-n** flag takes effect.

-j Combines all commands called only once under the heading `other`.

-n Sorts by the number of times the commands were called. When this flag is used with the **-c** flag, only the **-n** flag takes effect.

-o Displays a command summary of nonprime time commands. You can use this flag only when the **-a** flag is used.

-p Displays a command summary of prime time commands. You can use this flag only when the **-a** flag is used.

-s Assumes that any named files that follow this flag are already in binary format.

-t Processes all records as total accounting records. The default binary format splits each field into prime and nonprime time sections. This option combines the prime and non-prime time parts into a single field that is the total of both, and provides upward compatibility with old style acctcms binary summary format records. This flag cannot be used with the **-a** flag.

Security

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

Examples

To collect daily command accounting records in a `today` file and maintain a running total in a `total` file, add the following to a shell script:

```
acctcms File . . . > today
cp total previoustotal
acctcms -s today previoustotal > total
acctcms -a -s total
```

The *File* parameters that you specify are redirected to a file called `today`, added to the previous total (in a file renamed `previoustotal`) to produce a new total (called `total`). All files are binary files. In the last line, the **-a** flag displays the `total` file in ASCII format so you can view the report.

Files

/etc/acct/holidays	Specifies prime and nonprime time for accounting records.
/usr/sbin/acct/acctcms	Contains the **acctcms** command.

Related Information

The **lastcomm** command, **runacct** command.

The **acct** file format, **utmp**, **wtmp**, **failedlogin** file format.

The **acct** subroutine.

For more information about the Accounting System, the preparation of daily and monthly reports, and the accounting files, see the System accounting in *Operating system and device management*.

Setting up an accounting subsystem in *Operating system and device management* describes the steps you must take to establish an accounting system.

Accounting commands in *Operating system and device management*.

Monitoring and tuning commands and subroutines in *Performance management*.

---

acctcom Command

Purpose

Displays summaries of process-accounting records for selected processes.

Syntax

/usr/sbin/acct/acctcom [[**-q** | **-o** *File*] | [**-a**] [**-b**] [**-c** *Classname*] [**-f**] [**-h**] [**-i**] [**-k**] [**-m**] [**-r**] [**-t**] [**-v**] [**-w** [**-X**] [**-W**]] [**-C** *Seconds*] [**-g** *Group*] [**-H** *Factor*] [**-I** *Number*] [**-l** *Line*] [**-n** *Pattern*] [**-O** *Seconds*] [**-u** *User*] [**-e** *Time*] [**-E** *Time*] [**-s** *Time*] [**-S** *Time*] [**-@** [*WparName*]] [*File ...*]

Description

The **acctcom** command reads process accounting records from files specified by the *File* parameter from standard input or from the **/var/adm/pacct** file. Then the **acctcom** command writes the records you request to standard output. This command is stored in the **/usr/sbin/acct** directory, for access by all users.

If you do not specify a *File* parameter and if standard input is assigned to a workstation or to the **/dev/null** file, as when a process runs in the background, the **acctcom** command reads the **/var/adm/pacct** file.

If you specify a *File* parameter, the **acctcom** command reads each file chronologically by process completion time. Usually, the **/var/adm/pacct** file is the current file that you want the **acctcom** command to examine. Because the **ckpacct** procedure keeps this file from growing too large, a busy system may have several **pacct** files. All but the current file have the path name **/var/adm/pacct***?*, where *?* (question mark) represents an integer.

Each record represents one completed process. The default display consists of the command name, user name, tty name, start time, end time, real seconds, CPU seconds, and mean memory size (in kilobytes). These default items have the following headings in the output:

```
COMMAND                    START   END    REAL    CPU     MEAN
NAME    USER    TTYNAME    TIME    TIME   (SECS)  (SECS)  SIZE(K)
```

If a process was run by the root user, the process name is prefixed with a # (pound sign). If a process is not assigned to a known workstation (for example, when the **cron** daemon runs the process), a ? (question mark) appears in the `TTYNAME` field.

Notes:

1. The **acctcom** command only reports on processes that have finished. Use the **ps** command to examine active processes.
2. If a specified time is later than the current time, it is interpreted as occurring on the previous day.

Flags

-a Shows some average statistics about the processes selected. The statistics are displayed after the output records.

-b Reads backwards, showing the most recent commands first. This flag has no effect when the **acctcom** command reads standard input.

-c *Classname* Selects processes belonging to the specified class.
Note: Accounting data cannot be retrieved for a deleted class.

-C *Seconds* Shows only processes whose total CPU time (system time + user time) exceeds the value specified by the *Seconds* variable.

-e *Time* Selects processes existing at or before the specified time. You can use the current locale to specify the order of hours, minutes, and seconds. The default order is *hh:mm:ss*.

-E *Time* Selects processes ending at or before the specified time. You can use the current locale to specify the order of hours, minutes, and seconds. The default order is *hh:mm:ss*. If you specify the same time for both the **-E** and **-S** flags, the **acctcom** command displays the processes that existed at the specified time.

-f Displays two columns related to the `ac_flag` field of the **acct.h** file: the first indicates use of the **fork** command to create a process, the second indicates the system exit value. Refer to the **acct** structure described in the **acct** file format in *AIX 5L Version 5.3 Files Reference*.

-g *Group* Selects processes belonging to the specified group. You can specify either the group ID or the group name.

-h Instead of mean memory size, shows the fraction of total available CPU time consumed by the process (hog factor). This factor is computed as:

```
(total CPU time) / (elapsed time)
```

-H *Factor* Shows only the processes that exceed the value of the *Factor* parameter. This factor, called the hog factor, is computed as:

```
(total CPU time) / (elapsed time)
```

-i Displays columns showing the number of characters transferred in read or write operations (the I/O counts).

-k Instead of memory size, shows total kcore minutes (memory measurement in kilobyte segments used per minute of run time).

-l *Line* (lowercase L) Shows only processes belonging to workstation **/dev/***Line*.

-I *Number* (uppercase i) Shows only processes transferring more than the specified number of characters.

-m Shows mean main-memory size. This is the default. The **-h** flag or **-k** flag turn off the **-m** flag.

-n *Pattern* Shows only commands matching the value of the *Pattern* variable, where *Pattern* is a regular expression. Regular expressions are described in the **ed** command. In addition to the usual characters, the **acctcom** command allows you to use a + (plus sign) as a special symbol for the preceding character.

-o *File* Copies selected process records to the specified file, keeping the input data format. This flag suppresses writing to standard output. This flag cannot be used with the **-q** flag.

-O *Seconds* Shows only processes with CPU system time exceeding the specified number of seconds.

-q Displays statistics but not output records. The statistics are the same as those displayed using the **-a** flag. The **-q** flag cannot be used with the **-o** flag.

-r Shows CPU factor. This factor is computed as:

```
(user-time) / (system-time + user-time)
```

-s *Time* Shows only those processes that existed on or after the specified time. You can use the current locale to specify the order of hours, minutes, and seconds. The default order is *hh:mm:ss*.

-S *Time* Shows only those processes starting at or after the specified time. You can use the current locale to specify the order of hours, minutes, and seconds. The default order is *hh:mm:ss*.

-t Shows separate system and user CPU times.

-u *User* Shows only processes belonging to the specified user. Enter one of the following for the *User* variable: a user ID, a login name to be converted to a user ID, a # (pound sign) to select processes run by the root user, or a ? (question mark) to select processes associated with unknown user IDs.

-v Eliminates column headings from the output.

-w Displays the class names to which the processes belong.

-W Prints all available characters of each user name instead of truncating to the first 8 characters. The output is also widened to 132 characters allowing the user name to use the additional space. The **-W** option is mutually exclusive with the **-X** option. When both flags are used the second flag is ignored.

-X Print all available characters of each user name instead of truncating to the first 8 characters. The user name is also moved to the last column of the output. The **-X** option is mutually exclusive with the **-W** option. When both flags are used the second flag is ignored.

-@ [*WparName*] Displays summaries of process-accounting records for selected processes per workload partition. If a workload partition is specified using the *WparName* parameter, the accounting records for the specified workload partition are displayed. If no workload partition is specified, the accounting records for all of the workload partitions are displayed. A workload partition name is displayed for each record.

The -@ option is not supported when executed within a workload partition.

Security

RBAC Disclaimer

Examples

1. To display information about processes that exceed 2 seconds of CPU time, enter:

   ```
   /usr/sbin/acct/acctcom -O 2 < /var/adm/pacct
   ```

   The process information is read from the **/var/adm/pacct** file.
2. To display information about processes belonging to the `finance` group, enter:

   ```
   /usr/sbin/acct/acctcom -g Finance < /var/adm/pacct
   ```

The process information is read from the **/var/adm/pacct** file.

3. To display information about processes that belong to the **/dev/console** workstation and that run after 5 p.m., enter:

   ```
   /usr/sbin/acct/acctcom -l /dev/console -s 17:00
   ```

   The process information is read from the **/var/adm/pacct** file by default.

4. To display all information about processes on a machine that has greater than 8 character user names, enter:

   ```
   /usr/sbin/acct/acctcom -X < /var/adm/pacct
   ```

   The process information is read from the **/var/adm/pacct** file.

5. To display information about processes that are run inside the `warpath` WPAR, use the following command:

   ```
   acctcom -@ warpath < /var/adm/pacct
   ```

   The process information is read from the **/var/adm/pacct** file.

6. To display information about processes that are run on all WPARs, use the following command:

   ```
   acctcom -@ < /var/adm/pacct
   ```

   The process information is read from the **/var/adm/pacct** file.

Files

/usr/sbin/acct/acctcom	Contains the **acctcom** command.
/var/adm/pacct	Contains the current process accounting file.
/etc/group	Contains the basic group attributes of groups.
/etc/passwd	Contains the basic attributes of users.

Related Information

The **ed** command, **ps** command, **runacct** command, **su** command.

The **cron** daemon.

The **acct** subroutine.

The **acct** file format, **utmp**, **wtmp, failedlogin** file format.

Accounting commands in *Operating system and device management*.

For more information about the accounting system, the preparation of daily and monthly reports, and the accounting files, see the System accounting in *Operating system and device management*.

Setting up an accounting subsystem in *Operating system and device management* describes the steps you must take to establish an accounting system.

Monitoring and tuning commands and subroutines in *Performance management*

The environment File describes environment variables and their functions.

acctcon1 or acctcon2 Command

Purpose

Performs connect-time accounting.

Syntax

acctcon1 [**-l** *File*] [**-o** *File*] [**-p**] [**-t**] [**-X**]

acctcon2 [**-X**]

Description

acctcon1

The **acctcon1** command is called by the **runacct** command to convert a sequence of login and logoff records (read from standard input) to a sequence of login session records (written to standard output). Input is normally redirected from the **/var/adm/wtmp** file. The input file can be a file other than **/var/adm/wtmp**, as long as it is in the correct format.

The **acctcon1** command displays the following in ASCII format:

- Login device
- User ID
- Login name
- Prime connect time (seconds)
- Non-prime connect time (seconds)
- Session starting time (numeric)
- Starting date and time (in date/time format)

The **acctcon1** command also maintains a list of ports on which users are logged in. When the **acctcon1** command reaches the end of its input, the command writes a session record for each port that still appears to be active. Unless the **-t** flag is used, the **acctcon1** command assumes that input is a current file and uses the current time as the ending time for each session still in progress.

The summary file generated with the **-l** flag helps an administrator track line usage and identify bad lines. All hang-ups, terminations of the **login** command, and terminations of the login shell cause the system to write logoff records. Consequently, the number of logoffs is often much higher than the number of sessions.

acctcon2

The **acctcon2** command, also called by the **runacct** command, converts a sequence of login session records produced by the **acctcon1** command into connect-time total accounting records. These records are merged with other total accounting records by the **acctmerg** command to produce a daily report.

Flags

Note: The following flags are used with the **acctcon1** command.

-l *File* (lowercase L) Writes a line-usage summary file showing the line name, the number of minutes used, the percentage of total elapsed time, the number of sessions charged, the number of logins, and the number of logoffs. If you do not specify a file name, the system creates the information in the **/var/adm/acct/nite/lineuse** file.

-o *File*	Writes to the specified file an overall record for the accounting period, giving starting time, ending time, number of restarts, and number of date changes. If you do not specify a file name, the system creates the **/var/adm/acct/nite/reboots** file.
-p	Displays only input. Line name, login name, and time are shown in both numeric and date/time formats. Without the **-p** flag specified, the **acctcon1** command would display input, converting input to session records, and write reports.
-t	Uses the last time found in the input as the ending time for any current processes. This, rather than current time, is necessary in order to have reasonable and repeatable values for files that are not current.
-X	Prints and processes all available characters for each user name instead of truncating to the first 8 characters. **Note:** The following flag can be used with both the **acctcon1** and **acctcon2** commands.

Security

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

Examples

1. To convert a sequence of login records (in the **/var/adm/wtmp** file) to a sequence of login session records (stored in the **/var/adm/logsess** file), include the following in a shell script:

   ```
   acctcon1 -t -l/var/adm/acct/nite/lineuse \
   -o/var/adm/acct/nite/reboots \
   </var/adm/wtmp > /var/adm/logsess
   ```

   The login session reports show an ending time that corresponds with the last time input was provided. Two reports are generated: a line-usage summary file named **/var/adm/acct/nite/lineuse**, an overall record for the accounting period, reported in the **/var/adm/acct/nite/reboots** file.
2. To convert a series of login session records (in the **/var/adm/acct/nite/ctmp** file) to a total accounting record (stored in the **/var/adm/logacct** file), include the following in a shell script:

   ```
   acctcon2 < /var/adm/acct/nite/ctmp \
    > /var/adm/logacct
   ```

Files

/usr/sbin/acct/acctcon1	Contains the **acctcon1** command.
/usr/sbin/acct/acctcon2	Contains the **acctcon2** command.
/var/adm/wtmp	Contains connect-time accounting data, including login, logout, and shutdown records.

Related Information

The **acctmerg** command, **fwtmp**, **acctwtmp**, or **wtmpfix** command, **init** command, **login** command, **runacct** command.

The **acct** file format, **utmp**, **wtmp**, **failedlogin** file format.

The **acct** subroutine.

For more information about the Accounting System, the preparation of daily and monthly reports, and the accounting files, see the System accounting in *Operating system and device management*.

Setting up an accounting subsystem in *Operating system and device management* describes the steps you must take to establish an accounting system.

acctctl Command

Purpose

Controls advanced accounting.

Syntax

acctctl fadd *file size*

acctctl frm *file*

acctctl freset *file*

acctctl fquery [*file*]

acctctl fswitch [*file*]

acctctl isystem {*time*|off}

acctctl iprocess {*time*|off}

acctctl agproc {on|off}

acctctl agke {on|off}

acctctl agarm {on|off}

acctctl trquery [*trid*] [**-@** [*wpar*]]

acctctl tron *trid* [**-@** *wpar*]

acctctl troff *trid* [**-@** *wpar*]

acctctl email {on|off|*addr*}

acctctl on [**-@** [*wpar*]]

acctctl off [**-@** [*wpar*]]

acctctl [**-@** [*wpar*]]

acctctl turacct {on|off}

Description

The administration of Advanced Accounting (AACCT) is organized around the following high level tasks, which are mostly performed by the **acctctl** command.

- Manage Accounting Data Files.
- Manage Project Definitions and Assignments.
- Manage Transactions.
- Manage Advanced Accounting Subsystem.

The -@ option is not supported when executed within a workload partition.

Managing Accounting Data Files

The first task is centered around file management. Files are pre-allocated and registered with the AACCT subsystem, so that it can continuously stream accounting data to these files. When an accounting file is filled, AACCT automatically switches to the next available registered file. If there is no such file, then incoming data might be lost, unless the administrator or the billing application quickly reacts to the problem.

Messages are sent alerting the administrator to the status of files, so that he can avoid these types of problems before they occur. The best approach is to allocate sufficient file space up front. Messages are sent, when a file approaches the full state, and when the system automatically switches to another file. Messages are sent by way of the syslog facility and email. These subsystems have to be correctly configured in order to receive messages.

When the system runs out of accounting files, it internally buffers accounting data, so data is not immediately lost. If the administrator does not respond in time and data is lost, then the system internally maintains some statistics about the outage, which it logs to the accounting subsystem, after the condition has been corrected.

Before starting AACCT, the system administrator should create the accounting files that will be needed on the system. The number and size of these files is workload dependent, so the administrator should choose values that are appropriate for the specific installation. The only recommendation is that at least two files be created, so that AACCT can remain active at all times.

The following commands are provided for managing files:

acctctl fadd *file size*	Allocates and defines an accounting file with specified filename and size. The size is in megabytes.
acctctl frm *file*	Removes the specified accounting file from the accounting subsystem. This will not remove the file from the file system.
acctctl freset *file*	Indicates that the specified file can now be reused by the accounting subsystem.
acctctl fquery [*file*]	Queries the state and current utilization of the specified file, if supplied, or all accounting files otherwise.
acctctl fswitch [*file*]	Forces accounting to switch to a new accounting file. The new file can be optionally specified.

All files must be fully qualified path names. When creating a file, ensure that the file system has enough space.

Managing Project Definitions and Assignments

The second task, Manage Project Definitions and Assignments, is supported through the **projctl** command. Projects are optional. For a description of this capability, see the **projctl** command in *AIX 5L Version 5.3 Commands Reference, Volume 4*.

Managing Transactions

The third task, Manage Transactions, is designed to control the type of accounting data that is produced, which is configuration dependent, because applications and middleware can provide transactions. The following types of accounting are supported on all systems:

- Process
- Disk
- Network interfaces
- File systems
- System (provides global CPU and memory use)

Administrative control over these sources of accounting data is provided by enabling or disabling the accounting records that they produce. Each accounting record is assigned a unique identifier, so that report and analysis commands can apply the appropriate templates when processing the accounting file. These identifiers also serve to name the different types of accounting that is supported and are specified as parameters to the transaction specific commands. Identifiers are listed in the **sys** file.

The following commands are provided for managing transactions:

acctctl trquery [*trid*] [**-@** [*wpar*]]	Queries the state and name of the specified *trid*, if supplied, or of all trids, otherwise. If you specify the **-@** option without the *wpar* parameter, query trids in all active workload partitions. If you specify the **-@** option with the *wpar* parameter, query trids for the specified workload partition only.
acctctl tron *trid* [**-@** *wpar*]	Enables the specified transaction. If you specify the **-@** option with the *wpar* parameter, enable the transaction in the specified workload partition only.
acctctl troff *trid* [**-@** *wpar*]	Disables the specified transaction. If you specify the **-@** option with the *wpar* parameter, disable the transaction in the specified workload partition only.

By default, all transactions identifiers are enabled.

Not all transaction identifiers can be disabled, because some of them are derived types and are dependent on other transactions. For example, the process aggregation record is dependent on the process record, so it can't be disabled by itself. Aggregation can be enabled or disabled, and process accounting can be enabled or disabled, but the transaction identifier that corresponds to the aggregated process record can't be disabled. Aggregation is a convenience in the sense that it sums up data internally, so that fewer records are produced. In some cases, data aggregation is provided to simplify data management.

Managing the Advanced Accounting Subsystem

The fourth task, Manage Advanced Accounting Subsystem, is concerned with controlling the execution environment of the subsystem itself. Sub-tasks are oriented towards configuring, running, stopping, and querying AACCT.

The following commands are provided for managing the subsystem:

acctctl email {on|off|*addr*}	Sets up e-mail notifications. If given the **on** subcommand, the last used e-mail address will be used. The e-mail address is limited to 80 characters. Mail must be configured for e-mail notification to function.
acctctl iprocess {*time*|off}	Enables process interval accounting every *time* minutes or disables process interval accounting entirely.
acctctl isystem {*time*|off}	Enables system interval accounting every *time* minutes or disables system interval accounting entirely.
acctctl agproc {on|off}	Enables or disables system-wide aggregation for processes.
acctctl agke {on|off}	Enables or disables system-wide aggregation for third party kernel extensions.
acctctl agarm {on|off}	Enables or disables system-wide aggregation for ARM transactions.
acctctl dump *pid*	Writes the accounting record for the named process into the accounting file.

acctctl on [**-@** [*wpar*]]	Starts Advanced Accounting. If you specify the **-@** option without the *wpar* parameter, start Advanced Accounting for all active workload partitions. If you specify the **-@** option with the *wpar* parameter, start Advanced Accounting for the specified workload partition only.
acctctl off [**-@** [*wpar*]]	Stops Advanced Accounting. If you specify the **-@** option without the *wpar* parameter, stop Advanced Accounting for all active workload partitions. If you specify the **-@** option with the *wpar* parameter, stop Advanced Accounting for the specified workload partition only.
acctctl [**-@** [*wpar*]]	Queries overall accounting state. If you specify the **-@** option without the *wpar* parameter, query the Advanced Accounting state for all active workload partitions. If you specify the **-@** option with the *wpar* parameter, query the Advanced Accounting state of the specified workload partition only.
acctctl turacct {on|off}	Enables or disables the accounting based on Scaled Performance Utilization Resources Register (SPURR) in turbo mode.

Exit Status

This command returns the following exit values:

0	The command executed successfully.
>0	An error occurred.

Security

Root authority is required to use this command.

Data files are created by this command. These files are owned by root, but are readable by members of the adm group.

RBAC Disclaimer

Examples

1. To display status, type:

   ```
   acctctl
   ```

   Output similar to the following is displayed:

   ```
   Advanced Accounting is not running.
   Email notification is off.
   The current email address to be used is not set.
   Process Interval Accounting is off.
   System Interval Accounting is off.
   System-wide aggregation of process data is off.
   System-wide aggregation of third party kernel extension data is off.
   System-wide aggregation of ARM transactions is off.
   Files: 0 defined, 0 available.
   ```

2. To turn on accounting, type:

   ```
   acctctl on
   ```

3. To add a 200 MB data file, type:

   ```
   acctctl fadd /var/aacct/acctdata1 200
   ```

4. To enable the process interval so that it collects data every 2 hours, type:

   ```
   acctctl iprocess 120
   ```

5. To set process aggregation, type:
   ```
   acctctl agproc on
   ```
6. To enable e-mail notification, type:
   ```
   acctctl email on
   ```
7. To specify an e-mail address for notification, type:
   ```
   acctctl email user@company.com
   ```
8. To turn on accounting for WPARs on system, use the following command:
   ```
   acctctl on -@
   ```
9. To list trids specific to a WPAR that is named `wpar1`, use the following command:
   ```
   acctctl trquery -@ wpar1
   ```

   A similar result will be displayed as follows:
   ```
   NUMBER        STATE           NAME
    33           disabled       wpar-proc
    34           disabled       wpar-agg_proc
    35           disabled       wpar-agg_app
    36           enabled        wpar-system
    38            enabled        wpar-file
    39           enabled        wpar-netif
    44           disabled       wpar-agg_KE
   ```

Location

/usr/bin/acctctl

Files

/var/aacct	Default directory for accounting data files.
/var/aacct/acctdata	Default accounting data file.

Data files can be created in other locations by the system administrator.

Related Information

The **projctl** command in *AIX 5L Version 5.3 Commands Reference, Volume 4*.

AIX 5L Version 5.3 Advanced Accounting Subsystem.

Application transactions are supported through the Application Response Measurement (ARM) APIs, which are documented in *Application Response Measure (ARM) Issue 4.0 - C Binding*, The Open Group. This document is available at http://www.opengroup.org/tech/management/arm.

---

acctdisk or acctdusg Command

Purpose

Performs disk-usage accounting.

Syntax

/usr/sbin/acct/acctdisk

/usr/sbin/acct/acctdusg [**-u** *File*] [**-p** *File*] [**-X**]

Description

The **acctdisk** and **acctdusg** commands are called by the **dodisk** command to perform disk-usage accounting. Usually, this procedure is initiated when the **cron** daemon runs the **dodisk** command.

Normally, the output of the **diskusg** command becomes the input of the **acctdisk** command. If a more thorough but slower version of disk accounting is needed, use the **dodisk -o** command to call the **acctdusg** command instead of the **diskusg** command.

Accounting is only done for files on the local file system for local users. System administrators who want to count remote users (such as YP clients or diskless clients) should use the **acctdusg -p** command.

acctdisk

The **acctdisk** command reads the output lines of the **diskusg** or **acctdusg** commands from standard input, converts each individual record into a total accounting record, and writes the records to standard output. These records are merged with other accounting records by the **acctmerg** command to produce the daily accounting report.

acctdusg

The **acctdusg** command is called by using the **dodisk -o** command, when a slow and thorough version of disk accounting is needed. Otherwise, the **dodisk** command calls the **diskusg** command.

The **acctdusg** command reads a list of files from standard input (usually piped from a **find / -print** command), computes the number of disk blocks (including indirect blocks) allocated to each file owner, and writes an individual record for each user to standard output. By default, the command searches for login names and numbers in the **/etc/passwd** file. You can search other files by specifying the **-p** *File* flag and variable. Each output record has the following form:

```
uid login #blocks
```

The `#blocks` value is the number of 1KB blocks utilized by the user.

Flags

-p *File*	Searches the specified file for login names and numbers, instead of searching the **/etc/passwd** file.
-u *File*	Places, in the specified file, records of the file names that are exempt from charges.
-X	Turns on long username support.

Security

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

Examples

1. To start normal disk accounting procedures, add a line similar the following to a **crontab** file so that the **cron** daemon runs disk accounting commands automatically:

   ```
   0 2 * * 4 /usr/sbin/acct/dodisk
   ```

   In this example, the **dodisk** procedure runs at 2 a.m. (`0 2`) every Thursday (`4`) and the **dodisk** procedure calls the **diskusg** and **acctdisk** commands to write disk usage records to the **/usr/adm/acct/nite/dacct** file.
2. To start a thorough disk accounting procedure, add a line similar the following to a **crontab** file so that the **cron** daemon runs disk accounting commands automatically:

   ```
   0 2 * * 4 /usr/sbin/acct/dodisk -o
   ```

In this example, the **dodisk** procedure runs at 2 a.m. (`0 2`) every Thursday (`4`) and the **dodisk** procedure calls the **acctdusg** and **acctdisk** commands to write disk usage records to the **/var/adm/acct/nite/dacct** file.

Files

/usr/sbin/acct/acctdisk	Contains the **acctdisk** command.
/usr/sbin/acct/acctdusg	Containsthe **acctdusg** command.
/etc/passwd	Contains the basic attributes of user.
/usr/sbin/acct	Directory holding all accounting commands.

Related Information

The **acctmerg** command, **diskusg** command, **dodisk** command, **runacct** command.

The **cron** daemon.

The **acct** file format, **utmp**, **wtmp**, **failedlogin** file format.

The **acct** subroutine.

System accounting in *Operating system and device management* provides more information about the accounting system, the preparation of daily and monthly reports, and the accounting files.

Setting up an accounting subsystem in *Operating system and device management* describes the steps you must take to establish an accounting system.

---

acctmerg Command

Purpose

Merges total accounting files into an intermediary file or a daily report.

Syntax

/**usr/sbin/acct/acctmerg** [**-a** [*Specification*]] [**-h** [*Specification*]] [**-i** [*Specification*]] [**-p** [*Specification*]] [**-q** *Filename*] [**-v** [*Specification*]] [**-X**] [**-t**] [**-u**] [*File ...*]

Description

The **acctmerg** command merges process, connect-time, fee, disk-usage, and queuing (printer) total accounting records (in **tacct** binary or **tacct** ASCII format, **tacctx** binary, or **tacctx** ASCII format) and then writes the results to standard output. (See the **tacct** structure in the **acct** File Format for a description of the total accounting format or **/usr/include/sys/tacct.h** for a description of the **tacctx** format). The **acctmerg** command reads the total accounting records from standard input and from the additional files (up to nine) specified by the *File* parameter. The **acctmerg** command then merges the records by identical keys, usually a user ID and name. To facilitate storage, the **acctmerg** command writes the output in binary format unless you use either the **-a**, **-v**, or **-p** flag.

The **acctmerg** command is called by the **runacct** command to produce either an intermediate report when one of the input files is full, or to merge the intermediate reports into a cumulative total. The intermediate report is stored in the **/var/adm/acct/nite(x)/daytacct** file. The cumulative report is stored in the **/var/adm/acct/sum(x)/tacct** file. The cumulative total is the source from which the **monacct** command produces the ASCII-format monthly summary report. The monthly summary report is stored in the **/var/adm/acct/fiscal** file.

The *Specification* variable allows you to select input or output fields, as illustrated in Example 1. A field specification is a comma-separated list of field numbers, in the order specified in the **tacct(x)** structure in the **acct** File Format. Field ranges may be used, with array sizes taken into account, except for the *ta_name* characters. In the following example:

```
-h2-3,11,15-13,2
```

The **-h** flag causes column headings to display for the following types of data, in this order:

- login name (2)
- prime CPU (3)
- connect time (11)
- fee (15)
- queuing system (14, as implied in the range)
- disk usage data (13)
- the login name again (2)

The default displays all fields, otherwise specified as 1-18 or 1-, and produces wide output lines containing all the available accounting data.

Queueing system, disk usage, or fee data can be converted into **tacct** records by using the **acctmerg -i** *Specification* command.

The **tacct** fields are:

No. Header	Description
1 UID	User ID number.
2 LOGIN NAME	Login name of user.
3 CPU PRIME	Cumulative CPU minutes during prime hours.
4 CPU NPRIME	Cumulative during non-prime hours.
5 KCORE PRIME	Cumulative minutes spent in the kernel during prime hours.
6 KCORE NPRIME	Cumulative during non-prime hours.
7 BLKIO PRIME	Cumulative blocks transferred during prime hours.
8 BLKIO NPRIME	Cumulative during non-prime hours.
9 RW/WR PRIME	Cumulative blocks read/written during prime hours.
10 RW/WR NPRIME	Cumulative during non-prime hours.
11 CONNECT PRIME	Cumulative connect time (minutes) during prime hours.
12 CONNECT NPRIME	Cumulative during non-prime hours.
13 DISK BLOCKS	Cumulative disk usage.
14 PRINT	Queuing system charges. (pages)
15 FEES	Fee for special services.
16 # OF PROCS	Count of processes.
17 # OF SESS	Count of login sessions.
18 # OF SAMPLES	Count of count of disk samples.

Flags

-a[*Specification*]	Produces output in the form of ASCII records.
-h[*Specification*]	Displays column headings. This flag implies the **-a** flag, but is effective with **-p** or **-v**.
-i[*Specification*]	Expects input files composed of ASCII records, which are converted to binary records.
-p[*Specification*]	Displays input without processing. The output is in ASCII format.

-q *Filename*	Reads the specified **qacct** file (**accrec.h** file format) and produces output records sorted by user ID and user name. These records contain the user ID, user name, and number of pages printed.
-t	Produces a single record that contains the totals of all input.
-u	Summarizes by user ID rather than by user name.
-v[*Specification*]	Produces output in ASCII format, with more precise notation for floating-point numbers.
-X	Prints and processes all available characters for each user name instead of truncating to the first 8 characters.

Security

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

Examples

1. To merge disk accounting file `dacct` with field specification `-i1-2,13,18` into an existing total accounting file, `tacct`, enter:

   ```
   acctmerg -i1-2,13,18 <dacct | acctmerg tacct >output
   ```

   The **acctmerg** command reads the field specifications for the user ID, login name, number of blocks, and number of disk samples (`i1-2,13,18`) from the **dacct** file, merges this information with a **tacct** record, and writes the result to standard output.
2. To make repairs to the **tacct** format file `jan2.rpt`, first enter:

   ```
   acctmerg  -v <Jan.2.rpt >jan2.tmp
   ```

   Now edit the file `jan2.tmp` as desired. This command redirects the content of `Jan2.rpt` to `Jan2.tmp`, with the output in ASCII format.
3. To redirect `Jan2.tmp` to `Jan2.rpt`, with the output in binary record format, enter the following command:

   ```
   acctmerg  -i <jan2.tmp >jan2.rpt
   ```

Files

/usr/sbin/acct/acctmerg	Contains the **acctmerg** command.
/usr/include/sys/acct.h	Contains the **acct** and **tacct** file formats.
/var/adm/acct/nite/daytacct	Contains an intermediate daily total accounting report in binary format.
/var/adm/acct/sum/tacct	Contains the cumulative total accounting report for the month in binary format.
/var/adm/acct/fiscal	Contains the monthly accounting summary report, produced from the records in the **/var/adm/acct/sum/tacct** file.

Related Information

The **acctcms** command, **acctcom** command, **acctcon1** or **acctcon2** command, **acctdisk** command, **acctprc1**, **acctprc2**, or **accton** command, **fwtmp** command, **runacct** command.

The **acct** file format, **utmp**, **wtmp**, **failedlogin** file format.

The **acct** subroutine.

System accounting in *Operating system and device management*.

Print spooler in *Printers and printing*.

Setting up an accounting subsystem in *Operating system and device management* describes the steps you must take to establish an accounting system.

---

acctprc1, acctprc2, or accton Command

Purpose

Performs process-accounting procedures.

Syntax

/usr/sbin/acct/acctprc1 [*InFile*]

/usr/sbin/acct/acctprc2 [**-X**]

/usr/sbin/acct/accton [[-@] *OutFile*]

Description

The three **acctprc** commands, **acctprc1**, **acctprc2**, and **accton,** are called by the **runacct** command to perform process-accounting shell procedures.

The **acctprc1** command reads records from standard input that are in the **acct** format, adds the login names that correspond to user IDs, and then writes an ASCII record to standard output. This record contains the user ID, login name, prime CPU time, nonprime CPU time, the total number of characters transferred (in 1024-byte units), the total number of blocks read and written, and mean memory size (in 64-byte units) for each process.

If specified, the *InFile* parameter contains a list of login sessions in **utmp** format, sorted by user ID and login name. If the *File* parameter is not specified, **acctprc1** gets login names from the**/etc/passwd** password file. The information in the *InFile* parameter helps distinguish among different login names that share the same user ID.

The **acctprc2** command reads (from standard input) the records written by the **acctprc1** command, summarizes them by user ID and name, and writes the sorted summaries to standard output as total accounting records.

When the **accton** command is used without parameters, process accounting is turned off. If you specify the *OutFile* parameter (an existing file), process accounting is turned on, and the kernel adds records to that file. You must specify the *OutFile* parameter for process accounting to start. The *OutFile* parameter is not created by the **accton** command. The file specified by the *OutFile* parameter must already exist with the proper group, owner, and permissions. Many shell scripts expect the **/var/adm/pacct** file.

Flags

-X Process all available characters for each use rname instead of truncating to the first 8 characters. This flag also causes the **acctprc2** command to produce **tacctx** formatted binary records instead of **tacct** binary records.
Note: This flag can only be used with the **acctprc2** command.

-@ Include workload partition process accounting records in the global workload partition's accounting output file. This option is not valid inside a workload partition.

Security

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

Examples

1. To add a user name to each process-accounting record in a binary file and convert the records to an ASCII file named `out.file`, enter the following commands or use the lines in a shell script:

   ```
   /usr/sbin/acct/acctprc1 < /var/adm/pacct >out.file
   ```

2. To produce a total accounting record of the ASCII output file in example 1, enter the following commands or use the lines in a shell script:

   ```
   /usr/sbin/acct/acctprc2 < out.file > \
   /var/adm/acct/nite/daytacct
   ```

   The resulting file is a binary total accounting file in **tacct** format, containing individual records sorted by user ID. The file `/var/adm/acct/nite/daytacct` is merged with other total accounting records by the **acctmerg** command to produce the daily summary record in the **/var/adm/acct/sum/tacct** file.

3. To turn off process accounting, enter:

   ```
   /usr/sbin/acct/accton
   ```

Files

/usr/sbin/acct/acctprc1	Contains the **acctprc1** command.
/usr/sbin/acct/acctprc2	Contains the **acctprc2** command.
/usr/sbin/acct/accton	Contains the **accton** command.
/etc/accton	Symbolic link to the actual **accton** command directory.
/etc/passwd	Contains the basic user attributes, including the user IDs used by the **acctprc1** command.

Related Information

The **acctmerg** command,**runacct** command.

The **acct** file format, **utmp** file format.

For more information about the accounting system, the preparation of daily and monthly reports, and the accounting files, see the System accounting in *Operating system and device management*.

Setting up an accounting subsystem in *Operating system and device management* describes the steps you must take to establish an accounting system.

Accounting commands in *Operating system and device management*.

Monitoring and tuning commands and subroutines in *Performance management*

---

acctrpt Command

Purpose

Generates advanced accounting subsystem data reports.

Syntax

acctrpt [**-f** *filename*] [**-F**] [**-U** *uid*] [**-G** *gid*] [**-P** *projID*] [**-C** *command*] [**-b** *begin_time*] [**-e** *end_time*] [**-p** *projfile*] [**-n**]

acctrpt [**-f** *filename*] [**-F**] **-L** *resource* [**-b** *begin_time*] [**-e** *end_time*]

acctrpt [**-f** *filename*] [**-F**] **-T** [**-b** *begin_time*] [**-e** *end_time*]

acctrpt { **-c** | **-x** } [**-f** *filename*] [**-p** *projfile*] [**-n**]

acctrpt [**-b** *begin_time*] [**-e** *end_time*] [[[**-U** *uid*] [**-G** *gid*] [**-C** *command*] [**-@** *wpar*]] | [**-L** *resource* [**-@** *wpar*]]] [**-n**] [**-f** *filename*]

Description

The **acctrpt** command displays the advanced accounting statistics. advanced accounting subsystem supports process accounting, LPAR accounting, and transaction accounting.

For process accounting, users can generate accounting reports by projects, by groups, by users, by commands, or by a combination of these four identifiers. The command arguments **-U**, **-G**, **-P**, and **-C** command arguments are used to generate process accounting reports. The order in which these arguments are specified affects the order in which the data is displayed in the report. For example, the **acctrpt -U ALL -P ALL** command sorts by UID first and project second.

For LPAR accounting, users can generate accounting reports that describe the system-level use of resources, such as processors, memory, file systems, disks, and network interfaces. The system accounting interval must be enabled to collect accounting statistics for system resources. The **-L** command argument is used to generate LPAR accounting reports.

Note: The **-L** argument provides OS image level statistics, so it can also be used on systems that are not LPAR systems.

For transaction accounting, users can generate accounting reports describing application transactions. Transaction reports provide scheduling and accounting information, such as transaction resource usage requirements. These reports consume data that is produced by applications that are instrumented with the application response and measurement application programming interface (APIs). The **-T** command argument is used to generate transaction accounting reports.

If the **-U**, **-G**, **-P**, **-C**, **-L**, and **-T** command arguments are not specified, individual process accounting records are displayed.

Flags

-@ *wpar*
Specifies the workload partition for which the report is generated.

The -@ option is not supported when executed within a workload partition.

-b *begin_time*
Specifies the begin time of an interval. The *begin_time* parameter is a 10-character string in the *MMDDhhmmyy* format, where *MM* is month, *DD* is day, *hh* is hour, *mm* is minute, and *yy* is the last 2 digits of the year. All characters are numeric. If *begin_time* is not specified, all encountered records that were written before *end_time* are considered. If neither *end_time* or *begin_time* is specified, all records are considered.

-C *command*
Displays process accounting statistics for the specified command. More than one command name can be specified using a comma-separated list. Only the first 12 characters of the base command name are considered. To display all commands, specify **-C ALL**.

-c
Displays the project definitions in human readable format.

-e *end_time*
Specifies the end time of an interval. The *end_time* parameter is a 10-character string in the *MMDDhhmmyy* format, where *MM* is month, *DD* is day, *hh* is hour, *mm* is minute, and *yy* is the last 2 digits of the year. All characters are numeric. If *end_time* is not specified, all encountered records that were written after *begin_time* are considered. If neither *end_time* or *begin_time* is specified, all records are considered.

-f *filename*
Specifies the path name of the accounting data file to be used. More than one file can be specified using a comma-separated list. If the **-f** flag is not specified, the **/var/aacct/aacctdata** file is used by default.

-F
Displays information about the specified accounting data file. The report includes the host name, partition name, machine model, and serial number of the system where the accounting data file was generated.

-G *gid*
Displays process accounting statistics for the specified GIDs. More than one GID can be specified using a comma-separated list. To display all GIDs, specify **-G ALL**.

-L *resource*
Displays LPAR accounting statistics for the specified resource. The *resource* parameter must be one of the following values:

cpumem
CPU and memory statistics

filesys File system statistics

netif Network interface statistics

disk Disk statistics

vtarget VSCSI target statistics

vclient VSCSI client statistics

ALL All LPAR resource statistics

The **-L** argument cannot be specified with the **-U**, **-P**, **-G**, **-C**, or **-T** flags.

-n
Displays the IDs in numbers. By default, names are displayed.

-P *projID*
Displays process accounting statistics for the specified project ID. More than one project ID can be specified using a comma-separated list. To display all projects, specify **-P ALL**.

-p *projfile*
Specifies the project definition file to be used to resolve the projects associated with the transaction records. If **-p** is not specified, the projects are resolved using the currently loaded projects.

-T
Displays transaction accounting statistics. The **-T** argument cannot be specified with **-U**, **-P**, **-G**, **-C**, or **-L** flags.

-U *uid*
Displays process accounting statistics for the specified UIDs. More than one UID can be specified using a comma-separated list. To display all UIDs, specify **-U ALL**.

-x
Displays the project definitions in the project definition file format.

Exit Status

0	Successful completion.
>0	An error occurred.

Security

RBAC Disclaimer

Examples

1. To generate a file header report from the **/var/aacct/acctdata** data file, type:

   ```
   acctrpt -F -f /var/aacct/acctdata
   ```

2. To generate process accounting report by Users from the **/var/aacct/acctdata** data file, type:

   ```
   acctrpt -U ALL -f /var/aacct/acctdata
   ```

3. To generate a process accounting report for user ID 256 and user ID 257 and command **uname** from the **/var/aacct/acctdata** data file, type:

   ```
   acctrpt -U 256 257 -C uname -f /var/aacct/acctdata
   ```

4. To generate a process accounting report by projects and by users from the**/var/aacct/acctdata** data file, type:

   ```
   acctrpt -P ALL -U ALL -f /var/aacct/acctdata
   ```

5. To generate CPU and Memory statistics from the **/var/aacct/acctdata** data file, type:

   ```
   acctrpt -L cpumem -f /var/aacct/acctdata
   ```

6. To display the project definitions associated with the accounting records, type:

   ```
   acctrpt -c -f /var/aacct/acctdata
   ```

   Information similar to the following is displayed:

   ```
   PROJNAME        PROJID   AGGR       ORIGIN


   System          0        ENABLED    LOCAL
   ```

7. To display the associated IDs in numbers, type:

   ```
   acctrpt -P ALL -f /var/aacct/acctdata -n
   ```

Standard Output

Based on the **-f** option, the **acctrpt** command displays the following values in the File Header report.

File Name	The full path name of the accounting data file.
Open Date	The timestamp of first transaction record in the data file.
Last Close Date	The timestamp of last transaction record in the data file.
Host Name	The host where the data file was produced.
Partition Name	The partition where the data file was produced.
Partition ID	The partition number where the data file was produced.
System Model	The system model where the data file was produced.
System ID	The system serial number where the data file was produced.

Based on one or more of the **-P**, **-G**, **-U**, or **-C** options, the **acctrpt** command displays the following values in the Process Accounting report.

PROJID	The project name (Project ID).
UID	The user name (User ID).
GID	The group name (Group ID).

CMD	The base name of the executed command.
CNT	The count of transaction records aggregated per row of accounting report.
CPU	The CPU time (in seconds).
LFILE	The local File I/O (in MB).
DFILE	Other File I/O (in MB).
LSOCKET	The local socket I/O (in MB).
RSOCKET	Other socket I/O (in MB).
DMEM	Page seconds of disk pages.
PMEM	Page seconds of real pages.
VMEM	Page seconds of virtual memory.

Based on the **-L cpumem** option, the **acctrpt** command displays the following values in the CPU and Memory LDAP Accounting report.

CNT	The count of transaction records aggregated per row of accounting report.
IDLE	The CPU idle time (in seconds).
IOWAIT	The CPU I/O wait time (in seconds).
SPROC	The system process time (in seconds).
UPROC	The user process time (in seconds).
INTR	The interrupt time (in seconds).
IO	The number of I/Os.
PGSPIN	The number of page swap-ins.
PGSPOUT	The number of page swap-outs.
LGPGUTIL	The average utilization of large page pool.
PGRATE	The average page rate (per second).
PMEMUTIL	The average amount of physical memory that is allocated to an LPAR (in MB).
IOMEMUTIL	The average utilization of I/O memory entitlement (in MB).

Based on the **-L filesys** option, the **acctrpt** command displays the following values in the File Systems LPAR Accounting report.

CNT	The count of transaction records aggregated per row of accounting report.
DEVNAME	The device name.
MOUNTPT	The mount point name.
FSTYPE	The file system type.
RDWR	The number of reads and writes.
OPEN	The number of file opens.
CREATE	The number of file creates.
LOCKS	The number of file locks.
XFERS	The data transferred (in MB).

Based on the **-L netif** option, the **acctrpt** command displays the following values in the Network Interfaces LPAR Accounting report.

CNT	The count of transaction records aggregated per row of accounting report.
NETIFNAME	The network interface name.
NUMIO	The number of I/Os.
XFERS	The data transferred (in MB).

Based on the **-L disk** option, the **acctrpt** command displays the following values in the Disks LPAR Accounting report.

CNT	The count of transaction records aggregated per row of accounting report.

DISKNAME	The disk name.
BLKSZ	The disk block size (in bytes).
XFERS	The number of disk transfers.
READ	The number of reads from the disk.
WRITE	The number of writes to the disk.

Based on the **-L vtarget** option, the **acctrpt** command displays the following values in the VSCSI Targets LPAR Accounting report.

CNT	The count of transaction records aggregated per row of accounting report.
CLIENT#	The client partition number.
SERVERID	The server Unit ID.
UNITID	The device logical unit ID.
BYTESIN	The data in (in MB).
BYTESOUT	The data out (in MB).

Based on the **-L vclient** option, the **acctrpt** command displays the following values in the VSCSI Clients LPAR Accounting report.

CNT	The count of transaction records aggregated per row of accounting report.
CLIENT#	The client partition number.
SERVERID	The server Unit ID.
UNITID	The device logical unit ID.
BYTESIN	The data in (in MB).
BYTESOUT	The data out (in MB).

Based on the **-T** option, the **acctrpt** command displays the following values in the Transaction Accounting report.

PROJID	The project name (Project ID).
CNT	The count of transaction records aggregated per row of accounting report.
CLASS	The account class.
GROUP	The application group name.
NAME	The application name.
TRANSACTION	The transaction name
USER	The user name.
RESPONSE	The response time (in milliseconds).
QUEUED	The queued time (in milliseconds).
USER	The CPU time (in milliseconds).

If you specify the **-@** flag , the **acctrprt** command displays workload partition names in the process accounting report and the LPAR accounting report.

Note: Some of the transaction records displayed by **-U**, **-G**, **-P** and **-C** cannot be aggregated. For example, the transaction records that belong to the transaction ID `TRID_agg_proc` cannot be aggregated on group IDs and command names because these transaction records do not have the respective fields. For such records, the **acctrpt** command displays a `*` (asterisk) character in the command name field and a value of `-2` in the group ID field. It is an indication that these records are not aggregated and the caller has to look up for the command name.

Files

/usr/bin/acctrpt	Contains the **acctrpt** command.

/var/aacct/acctdata	Contains the default accounting data file.

Related Information

The **libaacct.a** library interfaces in the in *AIX 5L Version 5.3 Technical Reference: Base Operating System and Extensions Volume 1*.

AIX 5L Version 5.3 Advanced Accounting Subsystem.

---

acctwtmp Command

Purpose

Manipulates connect-time accounting records by writing a **utmp** record to standard output.

Syntax

/usr/sbin/acct/acctwtmp *"Reason"*

Description

The **acctwtmp** command is called by the **runacct** command to write a **utmp** record to standard output. The standard output includes the current date and time, plus a *Reason* string of 11 characters or less that you must enter.

Flags

None.

Parameters

Reason	String of 11 characters or less.

Security

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

Files

/usr/sbin/acct/accwtmp	Contains the **acctwtmp** command.
/var/adm/wtmp	Contains records of date changes that include an old date and a new date.
/usr/include/utmp.h	Contains history records that include a reason, date, and time.

Related Information

The **acctcon1** or **acctcon2** command, **acctmerg** command, **fwtmp** command, **runacct** command, **wtmpfix** command.

Setting up an accounting subsystem in *Operating system and device management*.

Accounting commands in *Operating system and device management*.

aclconvert Command

Purpose

Converts the access control information of a file system object from one type to another.

Syntax

aclconvert [**-R**] [**-I**] **-t** *ACLType File*

Description

The **aclconvert** command converts the access control information (ACL) of the file system object specified by the *File* parameter to another type as specified by *ACLType* argument input to command. The conversion could fail if the target ACL type is not supported by the file system where *File* exists. Also note that the ACL conversion will take place with the help of ACL type specific algorithm and invariably the conversion will be approximate. So the conversion could result in potential loss of access control and it is essential that the user of this command be sure that the converted ACL satisfies the necessary access restrictions. The user might manually review the access control information after the conversion for the file system object to ensure that the conversion was successful and fulfills the requirements of the desired access control.

Flags

-I	Does not display any warning messages.
-R	Recursive option allows the user to convert ACL types for all the file system objects under a directory structure to the desired ACL type.
-t *ACLType*	Specifies the target ACL type to which the File's ACL type will be converted. The conversion will succeed only if the file system in question supports the ACL type requested. If the conversion is lossy, a warning message will be issued. This kind of warning messages can be suppressed using **-I** option.

Exit Status

This command returns the following exit values:

0	The command executed successfully and all requested changes were made.
>0	An error occurred.

Security

Access Control

This command should be a standard user program and have the trusted computing base attribute.

Auditing Events

If the auditing subsystem is properly configured and is enabled, the **aclconvert** command generates the following audit record or event every time the command is run:

Event	Information
FILE_Acl	Lists access controls.

RBAC Disclaimer

Examples

1. To convert the access control information for the `status` file to AIXC ACL type, type:

   ```
   aclconvert -t AIXC status
   ```

   Conversion takes place and any warning or error message is displayed.
2. To convert the access control information for the all file system objects under directory `dir1` file to AIXC ACL type and ignore any warning messages, type:

   ```
   aclconvert -RI -t AIXC dir1
   ```

   This converts all file system objects under `dir1` to the ACL type AIXC..

Location

/usr/bin/aclconvert

Files

/usr/bin/aclconvert	Contains the **aclconvert** command.

Related Information

The **acledit** command, **aclput** command, **chmod** command.

Access control lists in *Operating system and device management*.

The Auditing Overview in *Security* explains more about audits and audit events.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to Securing the network in *Security*.

---

acledit Command

Purpose

Edits the access control information of a file.

Syntax

acledit [**-t** *ACL_type*] [**-v**] *FileObject*

Description

The **acledit** command lets you change the access control information of the file specified by the *FileObject* parameter. The command displays the current access control information and lets the file owner change it with the editor specified by the **EDITOR** environment variable. Before making any changes permanent, the command asks if you want to proceed.

Note: The **EDITOR** environment variable must be specified with a complete path name; otherwise, the **acledit** command will fail. The maximum size of the ACL data is dependent on the ACL type.

The access control information displayed depends on the ACL type associated with the file system object. Information typically includes access control entries displayed for owner and others. Also, file mode bits associated with the object could be displayed.

The following is an example of the access control information of a file:

```
attributes: SUID
base permissions:
    owner  (frank): rw-
    group (system): r-x
    others        : ---
extended permissions:
    enabled
        permit    rw-    u:dhs
        deny      r--    u:chas,    g:system
        specify   r--    u:john,    g:gateway, g:mail
        permit    rw-    g:account, g:finance
```

Note: If the **acledit** command is operating in a trusted path, the editor must have the **trusted process** attribute set.

Flags

-t This optional input specifies the ACL type in which the ACL data will be stored at the end of the ACL editing process. If no option is specified, then the ACL currently associated with the file system object will be edited in its ACL type format. If an ACL type is specified with this flag, then it is assumed that user is trying to modify the current ACL type and store the ACL in a new ACL type format. When this flag is specified and the ACL type does not match the type that exists currently, it is expected that user will modify the contents of the ACL data to format into the new ACL type specific format before saving.

-v Displays the ACL information in Verbose mode. Comment lines will be added to explain more details about the ACL associated with the FS object. These comment lines are generated when the command is executed and do not reside anywhere persistently. Hence, any modifications to the same will be lost when **acledit** is exited.

Security

Access Control

This command should be a standard user command and have the **trusted computing base** attribute.

Auditing Events

If the auditing subsystem is properly configured and is enabled, the **acledit** command generates the following audit record or event every time the command is run:

Event	Information
FILE_Acl	Lists access controls.

For more details about how to properly select and group audit events, and how to configure audit event data collection, see Setting up Auditing in *Security*.

Files Accessed

Mode	File
x	**/usr/bin/aclget**
x	**/usr/bin/aclput**

RBAC Disclaimer

Examples

To edit the access control information of the `plans` file, enter:

```
acledit plans
```

Files

/usr/bin/acledit	Contains the **acledit** command.

Related Information

The **aclget** command, **aclput** command, **auditpr** command, **chmod** command.

Access control lists in *Operating system and device management*.

The Auditing Overview in *Security* explains more about audits and audit events.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to Securing the network in *Security*.

---

aclget Command

Purpose

Displays the access control information of a file.

Syntax

aclget [**-o** *OutAclFile*] [**-v**]*FileObject*

Description

The **aclget** command writes the access control information of the file specified by the *FileObject* parameter to standard output or to the file specified by the *OutAclFile* parameter.

The information that you view depends on the ACL type and typically includes the Access Control Entries (ACEs) depicting the access rights of the users in the system, including the owner of the file object.

Flags

-o *OutAclFile*	Specifies that the access control information be written to the file specified by the *OutFile* parameter.
-v	Displays the ACL information in Verbose mode. Comment lines will be added to explain more details about the ACL associated with the FS object. These comment lines are generated when the command is executed and do not reside anywhere persistently.

Security

Access Control

This command should be a standard user program and have the **trusted computing base** attribute.

RBAC Disclaimer

Access Control Lists

Access Control Lists form the core of protection of file system objects. Each file system object is uniquely associated with one piece of data, called ACL, that defines the access rights to the object. ACL could consist of multiple Access Control Entries (ACEs), each defining one particular set of access rights for a user. Typically ACE consists of information such as identification (to whom this ACE applies) and access rights (allow-read, deny-write). Note that ACE might also capture information such as inheritance flags and alarm and audit flags. The format and enforcement of ACL data is entirely dependent on the ACL type in

which they are defined. AIX provides for the existence of multiple ACL types on the operating systems. The list of ACLs supported by a file system instance is dependent on the physical file system implementation for that file system instance. For more information, see Access Control Lists in *Operating system and device management* for complete list of ACL types supported and also the details of individual ACL type details.

Examples

1. To display the access control information for the `status` file, enter:

   ```
   aclget  status
   ```

   An access control list appears, similar to the example in Access Control Lists.
2. To copy the access control information of the `plans` file to the `status` file, enter:

   ```
   aclget  plans  |  aclput  status
   ```

   This copies the access control information. In most cases, the ACL type associated with `plans` will be the ACL type of ACL associated with the target status. However, it is possible that the target file system does not support the ACL type associated with file system object plans. In this case, the operation will fail and an error message is displayed. The target will retain its original associated ACL.
3. To save the access control information of the `plans` file in the `acl1` file to edit and use later, enter:

   ```
   aclget   -o  acl1  plans
   ```

Files

/usr/bin/aclget	Contains the **aclget** command.

Related Information

The **acledit** command, **aclput** command, **chmod** command.

Access control lists in *Operating system and device management*.

The Auditing Overview in *Security* explains more about audits and audit events.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to Securing the network in *Security*.

---

aclgettypes Command

Purpose

Gets ACL types supported by a file system path.

Syntax

aclgettypes *FileSystemPath*

Description

The **aclgettypes** command retrieves the list of ACL types supported for a given file system path and displays the same. The default ACL type for the file system instance concerned will be displayed as the first entry. See Security Guide more information regarding supported ACL types.

Exit Status

This command returns the following exit values:

0	The command executed successfully and all requested changes were made.
>0	An error occurred.

Security

Access Control

This command should be a standard user program and have the **trusted computing base** attribute.

RBAC Disclaimer

Examples

1. To display ACL types supported by a file system instance that contains path **/home/plan1**, type:

```
aclgettypes /home/plan1
```

Location

/usr/bin/aclgettypes

Files

/usr/bin/aclgettypes	Contains the **aclgettypes** command.

Related Information

The **acledit** command, **aclput** command, **chmod** command.

Access control lists in *Operating system and device management*.

The Auditing Overview in *Security* explains more about audits and audit events.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to Securing the network in *Security*.

---

aclput Command

Purpose

Sets the access control information of a file.

Syntax

aclput [**-i** *inAclFile*] [**-R**] [**-t** *acl_type*] [**-v**]*FileObject*

Description

The **aclput** command sets the access control information of the file object specified by the *FileObject* parameter. The command reads standard input for the access control information, unless you specify the **-i** flag.

Note: If you are reading from standard input your entries must match the expected format of the access control information or you will get an error message. Use the Ctrl-D key sequence to complete the session.

Access Control List

Access Control Lists form the core of protection for file system objects. Each file system object is uniquely associated with one piece of data, called ACL, that defines the access rights to the object. ACL could consist of multiple Access Control Entries (ACEs), each defining one particular set of access rights for an user. Typically, ACE consists of information such as identification (to whom this ACE applies) and access rights (allow-read, deny-write). ACE might also capture information such as inheritance flags and alarm and audit flags. The format and enforcement of ACL data is entirely dependent on the ACL type in which they are defined. AIX provides for existence of multiple ACL types on the operating system. The list of ACLs supported by a file system instance is dependent on the physical file system implementation for that file system instance. See Access Control Lists in *Operating system and device management* for complete list of supported ACL types.

Flags

-i *inAclFile* Specifies the input file for access control information. If the access control information in the file specified by the *InFile* parameter is not correct, when you try to apply it to a file, an error message preceded by an asterisk is added to the input file.

Note: The size of the ACL information depends on the ACL type.

-R Applys ACL to this directory and its children file system objects recursively.

-t *ACL_type* Specifies the ACL type of the ACL information being displayed. If this option is not provided the actual ACL data in its original ACL type will be displayed.

-v Verbose option. This option displays many comment lines as part of the ACL data display. This could help in understanding the details of complex ACL types.

Security

Access Control

This command should be a standard user program and have the **trusted computing base** attribute.

Auditing Events

If the auditing subsystem is properly configured and is enabled, the **aclput** command generates the following audit record or event every time the command is run:

Event	Information
FILE_Acl	Lists file access controls.

For more details about how to properly select and group audit events, and how to configure audit event data collection, see Setting up Auditing in *Security*.

RBAC Disclaimer

Examples

1. To set the access control information for the status file with information from standard input, enter:

   ```
   aclput status
   attributes: SUID
   ```

   and then press the Ctrl-D sequence to exit the session.
2. To set the access control information for the `status` file with information stored in the `acldefs` file, enter:

   ```
   aclput    -i   acldefs   status
   ```

3. To set the access control information for the `status` file with the same information used for the `plans` file, enter:

   ```
   aclget   plans   |   aclput   status
   ```

4. To set the access control information for the `status` file with an edited version of the access control information for the `plans` file, you must enter two commands. First, enter:

   ```
   aclget    -o acl   plans
   ```

   This stores the access control information for the `plans` file in the `acl` file. Edit the information in the `acl` file, using your favorite editor. Then, enter:

   ```
   aclput    -i   acl   status
   ```

   This second command takes the access control information in the `acl` file and puts it on the `status` file.

Files

/usr/bin/aclput	Contains the **aclput** command.

Related Information

The **acledit** command, **aclget** command, **auditpr** command, **chmod** command.

Access control lists in *Operating system and device management*.

The Auditing Overview in *Security* explains more about audits and audit events.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to Securing the network in *Security*.

---

adb Command

Purpose

Provides a general purpose debug program.

Syntax

adb [**-k**] [**-I** *Directory*] [**-w**] [*ObjectFile* [*CoreFile*]]

Description

The **adb** command provides a debug program for programs. With this debug program, you can examine object and core files and provide a controlled environment for running a program.

Normally, the *ObjectFile* parameter is an executable program file that contains a symbol table. If the *ObjectFile* parameter does not contain a symbol table, the symbolic features of the **adb** command cannot be used, although the file can still be examined. The default for the *ObjectFile* parameter is **a.out**.

The *CoreFile* parameter is a core image file produced by running the *ObjectFile* parameter. The default for the *CoreFile* parameter is **core**.

While the **adb** command is running, it takes standard input and writes to standard output. The **adb** command does not recognize the Quit or Interrupt keys. If these keys are used, the **adb** command waits for a new command.

In general, requests to the **adb** command are in the following form:

[*Address*] [,*Count*] [*Command*] [;]

where *Address* and *Count* are expressions. The default for the *Count* expression is a value of 1. If the *Address* expression is specified, the **.** (period) variable is set to *Address*.

The interpretation of an address depends on the context in which it is used. If a subprocess is being debugged, addresses are interpreted in the usual way in the address space of the subprocess.

Enter more than one command at a time by separating the commands with a **;** (semicolon).

The **adb** debug program allows the use of various:

- expressions
- operators
- subcommands
- variables
- addresses

See the adb Debug Program Overview in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* for detailed information.

Note: If the object file does not contain the symbol table, the **adb** command will not be able to show the value of static, automatic, and external variables of a program.

Flags

-k	Causes kernel mapping.
-I *Directory*	Specifies a directory where files to be read with $< or $<< are sought. The default is the **/usr/ccs/bin/adb** file.
-w	Opens the *ObjectFile* and the *Corefile* parameters for reading and writing. If either file does not exist, this flag creates the file.

Return Values

The **adb** debug program is printed when there is no current command or format. The **adb** command indicates such things as inaccessible files, syntax errors, and abnormal termination of commands. Exit status is a value of 0, unless the last command was unsuccessful or returned non-zero status.

Files

/dev/mem	Provides privileged virtual memory read and write access.
a.out	Provides common assembler and link editor output.
core	Contains an image of a process at the time of an error.

Related Information

The **cc** command, **dbx** command.

adb Debug Program Overview in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

addbib Command

Purpose

Creates or extends a bibliographic database.

Syntax

addbib [**-a**] [**-p** *PromptFile*] *Database*

Description

The **addbib** command uses a series of prompts to guide the user through creating or extending a bibliographic database. The user can define responses to these prompts. All default prompts and instructions are contained in the **refer** message catalog.

The first prompt is `Instructions?`. If the answer is affirmative, you can receive directions.

If the answer is negative or if you press the Enter key, you cannot receive directions. The **addbib** command then prompts for various bibliographic fields, reads responses from the terminal, and sends output records to the database specified by the *Database* parameter.

Pressing the Enter key (a null response) means to omit a particular field. Typing a - (minus sign) means to return to the previous field. A trailing backslash allows a field to be continued on the next line. The repeating `Continue?` prompt allows you to resume, to quit the current session, or to edit the database. To resume, type the defined affirmative answer or press the Enter key. To quit the current session, type the defined negative answer.

To edit the database, enter any system text editor (vi, ex, edit, ed).

Flags

-a Suppresses prompting for an abstract. Prompting for an abstract is the default. Abstracts are ended by pressing a Ctrl-D key sequence.

-p*PromptFile* Causes the **addbib** command to use a new prompting skeleton, which is defined in the file specified by the *PromptFile* parameter. This file contains prompt strings, a tab, and the key letters written to the specified database.

The following are the most common key letters and their meanings. The **addbib** command insulates you from these key letters, since it gives you prompts in English. If you edit the bibliography file later, you need to know this information.

%A Author's name

%B Book containing article referenced

%C City (place of publication)

%D Date of publication

%E Editor of book containing article referenced

%F Footnote number or label (supplied by the **refer** command)

%G Government order number

%H Header commentary, printed before reference

%I Issuer (publisher)

%J Journal containing article

%K Keywords to use in locating reference

%L Label field used by **-k** flag of the **refer** command

%M Bell Labs memorandum (undefined)

%N Number within volume

%O Other commentary, printed at end of reference

%P Page numbers

%Q Corporate or foreign author (unreversed)

%R Report, paper, or thesis (unpublished)

%S Series title

%T Title of article or book

%V Volume number

%X Abstract used by the **roffbib** command, not by the **refer** command

%Y,Z Ignored by the **refer** command.

Note: Except for the **%A** key letter, each field should be given just once. Only relevant fields should be supplied.

Examples

The following is an example of a bibliography file:

```
%A Bill Tuthill
%T Refer - A Bibliography System
%I Computing Services
%C Berkeley
%D 1982
%O UNIX 4.3.5.
```

Related Information

The **indxbib** command, **lookbib** command, **refer** command, **roffbib** command, **sortbib** command.

addrpnode Command

Purpose

Adds one or more nodes to a peer domain definition.

Syntax

addrpnode [**–c**] [**–h**] [**–TV**] *node_name1* [*node_name2* ...]

addrpnode [**–c**] { **–f** | **–F** { *file_name* | *"–"* } } [**–h**] [**–TV**]

Description

Before running addrpnode:
To set up the proper security environment, run the **preprpnode** command on each node that is to be added to the peer domain.

The **addrpnode** command adds the specified nodes to the online peer domain in which it (**addrpnode**) is run. This command must be run on a node that is online to the peer domain in which the new nodes are to be added. Though a node can be defined in multiple peer domains, it can only be online in one peer domain. To add one or more nodes to the peer domain, more than half of the nodes must be online.

To enable **addrpnode** to continue when there is an error on one of the nodes, use the **-c** flag.

The **addrpnode** command does *not* bring the added nodes online in the peer domain. To do this, use the **startrpnode** command.

Flags

–c Continues processing the command as long as at least one node can be added to the peer domain.

By default, if the **addrpnode** command fails on any node, it will fail on all nodes. The **-c** flag overrides this behavior, so that the **addrpnode** command will run on the other nodes, even if it fails on one node.

–f | –F { *file_name* **|** *"–"* **}**
Reads a list of node names from *file_name*. Each line of the file is scanned for one node name. The pound sign (**#**) indicates that the remainder of the line (or the entire line if the **#** is in column 1) is a comment.

Use **-f** *"-"* or **-F** *"-"* to specify **STDIN** as the input file.

-h Writes the command's usage statement to standard output.

-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) to be added to the 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 **addrpnode** command needs write permission for the **IBM.PeerDomain** resource class and the **IBM.PeerNode** resource class on each node that is to be added to the peer domain. This is set up by running the **preprpnode** command on each node to be added. Specify the names of all the nodes online in the peer domain with the **preprpnode** command. This gives the online nodes the necessary authority to perform operations on the nodes to be added.

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 in which the new nodes are to be added.

Implementation Specifics

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

Standard Input

When the **-f** ″**-**″ or **-F** ″**-**″ flagd is specified, this command reads one or more node names from standard input.

Standard Output

When the **-h** flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

To add the nodes **nodeB** and **nodeC** to the peer domain **ApplDomain** where **nodeA** is already defined and online to **ApplDomain**, run this command on **nodeA**:

```
addrpnode nodeB nodeC
```

Location

/usr/sbin/rsct/bin/addrpnode

Related Information

Books: *RSCT: Administration Guide*, for information about peer domain operations

Commands: **lsrpnode**, **mkrpdomain**, **preprpnode**, **rmrpnode**, **startrpdomain**, **startrpnode**

Information Files: **rmccli**, for general information about RMC-related commands

---

addX11input Command

Purpose

Adds an X11 input extension record into the ODM (Object Data Manager) database.

Syntax

addX11input

Description

The **addX11input** command is used to add an X11 input extension record into the ODM database. When you enter **addX11input** on the command line, the **addX11input** command requests *DeviceName, GenericName,* and *ModuleName* values in turn. The entire record is then added to the ODM database.

The command is a root/system user command. Its action fails with a permissions error if an unauthorized user attempts to add a record.

Error Codes

ODM could not open class	Returned if the X11 Input extension records in the ODM database are not found in the **/usr/lib/objrepos** directory.

Related Information

The **deleteX11input** command, **listX11input** command.

---

adfutil Command

Purpose

Provides the capability to merge Micro Channel information for PS/2 adapters with the Configuration Database in AIX 5.1 and earlier.

Syntax

adfutil [**-d** *Device* | **-f** *File*] [**-a** *AdapterName* | **-c** *PS/2CardID*] [**-m** *FileName ...*] [**-q**]

Description

The **adfutil** command provides the capability to field merge Micro Channel resource information for existing PS/2 adapters with predefined information in the Configuration database for AIX 5.1 and earlier. This is accomplished by processing information found on DOS formatted diskettes provided with the PS/2 adapter hardware. Included on these diskettes are adapter description files that are ASCII representations of adapter hardware attributes.

The naming convention for an adapter description file found on the DOS formatted diskette is @XXXX.ADF where XXXX is the *PS/2CardID*. If the command is invoked without arguments, the search centers around the home directory of the default device. If no files are found in the form of @XXXX.ADF, an error message is sent to standard output and the **adfutil** command ends. If a single adapter description file is found, execution is continued on that file. If multiple adapter description files are found, an error message is written to standard output and processing ends. If the **-c** flag is specified, a string is built that represents the corresponding DOS file name representation of the desired adapter description file. If this file does not exist on the specified device and path name or default, an error message is sent to standard output and processing ends.

When the adapter description file is found, the contents are written in the **/tmp/adf***nnn* file where *nnn* is the current process ID. This ID is removed after successful completion of the command.

Microcode files can be loaded independently of any adapter description file processing, and without disturbing the adapters database representation. Use the **-m** flag to load microcode files into the **/usr/lib/microcode** directory.

> **Attention:** Micro Channel adapters require bus attribute processing beyond what is supported by the bus configuration program, and should not be added to the system due to the possibility of adversely effecting the configuration of other devices on the system.

Flags

-a *AdapterName*	Searches the ODM database for candid information to form correct DOS filename for the adapter description file. The *AdapterName* parameter is a valid device name.
-c *PS/2CardID*	Identifies the *PS/2CardID* for the adapter. The card identifier is a four character alpha-numeric string that is found in the root of the DOS filename of the adapter description file. There is no default.
-d *Device*	Identifies the *Device* where the adapter description file resides. The default is **/dev/fd0**.
-f *File*	Identifies the file system path name for source adapter description file. If the **-f** flag is specified, any microcode keyword found in the adapter description file must specify a file system path name of the microcode source file.
-m *FileName*	Loads only microcode files found on diskette. If the *FileName* parameter is specified, files are loaded into the **/usr/lib/microcode** file. This is a microcode only flag.
-q	Toggles off the message to insert the adapter description file diskette.

Examples

1. To search the diskette drive **/dev/fd0** in the home directory for an adapter description file, enter:

   ```
   adfutil
   ```

2. To read **/home/owner/adf.file** as an adapter description file, enter:

   ```
   adfutil  -f /home/owner/adf.file
   ```

3. To search the default device **/dev/fd0** for the adapter description file labeled @0FFE.ADF without interrupting execution for the `insert diskette` prompt, enter:

   ```
   adfutil  -c 0FFE  -q
   ```

4. To load adapter microcode without processing adapter description file information, enter:

```
adfutil  -m FileName
```

Related Information

The **dosdir** command.

---

admin Command (SCCS)

Purpose

Creates and controls Source Code Control System (SCCS) files.

Syntax

To Create New SCCS Files

admin { **-n -i**[*FileName*] } [**-a** { *User* | *GroupID* }] ... [**-f** *HeaderFlag*[*Value*] ...] [**-r** *SID*] [**-t** *FileName*] [**-m** *ModificationRequestList*] [**-y**[*Comment*]] *File* ...

Note: Do not put a space between a flag and an optional (bracketed) variable.

To Modify Existing SCCS Files

admin [**-a** { *User* | *GroupID* }] ... [**-e** { *User* | *GroupID* }] ... [{ **-d** *HeaderFlag* | **-f** *HeaderFlag*[*Value*] ... }] [**-m** *ModificationRequestList*] [**-t**[*FileName*]] [**-y**[*Comment*]] *File* ...

Note: Do not put a space between a flag and an optional (bracketed) variable.

To Check Damaged SCCS Files

admin -h *File* ...

To Correct Damaged SCCS Files

admin -z *File* ...

Description

The **admin** command creates new Source Code Control System (SCCS) files or changes specified parameters in existing SCCS files.

The **admin** command can change the parameters controlling how the **get** command builds the files that you can edit. The parameters can also set conditions about who can access the file and which releases of the files may be edited.

If the file specified by the *File* parameter exists, the **admin** command modifies the file as specified by the flags. If the file does not exist and you supply the **-i** or **-n** flag, the **admin** command creates a new file and provides default values for unspecified flags.

If you specify a directory name for the *File* parameter, the **admin** command performs the requested actions on all SCCS files in that directory. All SCCS files contain the **s.** prefix before the file name. If you use a **-** (minus sign) for the *File* parameter, the **admin** command reads standard input and interprets each line as the name of an SCCS file. An end-of-file character ends input.

You must have write permission in the directory to create a file. All SCCS file names must have the form **s.***Name.* New SCCS files are created with read-only permission. The **admin** command writes to a temporary x-file, which it calls **x.***Name*. If it already exists, the x-file has the same permissions as the original SCCS file. The x-file is read-only if the **admin** command must create a new file. After successful completion of the **admin** command, the x-file is moved to the name of the SCCS file. This ensures that changes are made to the SCCS file only if the **admin** command does not detect any errors while running.

Directories containing SCCS files should be created with permission code 755 (read, write, and execute permissions for owner, read and execute permissions for group members and others). The SCCS files themselves should be created as read-only files (444). With these permissions, only the owner can use non-SCCS commands to modify SCCS files. If a group can access and modify the SCCS files, the directories should include group write permission.

The **admin** command also uses a temporary lock file (called **z.***Name*), to prevent simultaneous updates to the SCCS file by different users.

You can enter flags and input file names in any order. All flags apply to all the files. Do not put a space between a flag and an optional variable (variable enclosed in bracket). Header flags can be set with the **-f** flag and unset with the **-d** flag. Header flags control the format of the g-file created with the **get** command.

Flags

-a *User* or **-a** *GroupID*
Adds the specified user to the list of users that can make sets of changes (deltas) to the SCCS file. The *User* value can be either a user name or a group ID. Specifying a group ID is the same as specifying the names of all users in that group. You can specify more than one **-a** flag on a single **admin** command line. If an SCCS file contains an empty user list, anyone can add deltas. If a file has a user list, the creator of the file must be included in the list in order for the creator to make deltas to the file. If the *User* or *GroupID* parameter is preceded by an ! (exclamation point), specified users are denied permission to make deltas. For example, enter `-a !User`.

-d *HeaderFlag*
Deactivates the effects of the specified header flag within the SCCS file. You can specify this flag only with existing SCCS files. You can also specify more than one **-d** flag in a single **admin** command. Refer to the list of header flags that follows to learn more about the supported values.

-e *User* or **-e** *GroupID*
Removes the specified user from the list of users allowed to make deltas to the SCCS file. Specifying a group ID is equivalent to specifying all *User* names common to that group. You can specify several **-e** flags on a single **admin** command line.

-f *HeaderFlag*[*Value*]
Activates the specified header flag and value in the SCCS file. You can specify more than one header flag in a single **admin** command. There are 12 header flags. Refer to the list of header flags that follows to learn more about the supported values. Do not put a space between the *HeaderFlag* and *Value* variables.

-h
Checks the structure of the SCCS file and compares a newly computed checksum with the checksum that is stored in the first line of the SCCS file. When the checksum value is not correct, the file has been improperly modified or damaged. This flag helps you detect damage caused by the improper use of non-SCCS commands to modify SCCS files, as well as accidental damage. The **-h** flag prevents writing to the file, so it cancels the effect of any other flags supplied. If an error message is returned indicating the file is damaged, use the **-z** flag to re-compute the checksum. Then test to see if the file is corrected by using the **-h** flag again.

-i[*FileName*]
Gets the text for a new SCCS file from the *FileName* variable. This text is the first delta of the file. If you specify the **-i** flag but omit the file name, the **admin** command reads the text from standard input until it reaches an end-of-file character. If you do not specify the **-i** flag, but you do specify the **-n** flag, the command creates an empty SCCS file. The **admin** command can only create one file containing text at a time. If you are creating two or more SCCS files with one call to the **admin** command, you must use the **-n** flag, and the SCCS files created will be empty. Each line of the file specified by the *FileName* variable cannot contain more than 512 characters. The file name can include MBCS (multibyte character set) characters. Do not put a space between the flag and the *FileName* variable.

-m *ModificationRequestList*
Specifies a list of Modification Request (MR) numbers to be inserted into the SCCS file as the reason for creating the initial delta. A null or empty list can be considered valid, depending on the validation program used. The **v** header flag must be set. The MR numbers are validated if the **v** header flag has a value (the name of an MR number validation program). The **admin** command reports an error if the **v** header flag is not set or if MR validation fails.

-n
Creates a new, empty SCCS file. When the **-n** flag is used without the **-i** flag, the SCCS file is created with control information but without any file data.

-r *SID*
Specifies the SCCS identification string (SID) file version to be created. The *SID* variable accepts a delta with four levels: release, level, branch, and sequence, for example 3.2.5.1. If only release is specified, the **admin** command automatically assumes level 1. If you do not specify the **-r** flag, the initial delta becomes release 1, level 1 (that is, 1.1). For more details on specifying the SID, refer to the SID Determination table described in the **get** command.

You can specify the **-r** flag only if you also specify the **-i** or **-n** flag. Use this flag only when creating an SCCS file.

-t [*FileName*]
Takes descriptive text for the SCCS file from the file specified by the *FileName* variable. If you use the **-t** flag when creating a new SCCS file, you must supply a file name. In the case of existing SCCS files:

- Without a file name, the **-t** flag removes any descriptive text currently in the SCCS file.
- With a file name, the **-t** flag replaces any descriptive text currently in the SCCS file with text in the named file.
- The file name can include MBCS (multibyte character set) characters.

Do not put a space between the flag and the *FileName* variable.

-y [*Comment*]
Inserts the specified comment into the initial delta in a manner identical to that of the **delta** command. Use this flag only when you create an SCCS file. If you do not specify a comment, the **admin** command inserts a line of the following form:

```
date and time created YY/MM/DD HH:MM:SS by Login
```

The comments can include MBCS (multibyte character set) characters. Do not put a space between the flag and the *FileName* variable.

-z Re-computes the SCCS file checksum and stores it in the first line of the SCCS file (see the **-h** flag).

> **Attention:** Using the **admin** command with the **-z** flag on a damaged file can prevent future detection of the damage. This flag should only be used if the SCCS file is changed using non-SCCS commands because of a serious error.

File Specifies the name of the file created or altered by the **admin** command. If a **-** (minus sign) is specified, the **admin** command reads from standard input. An end-of-file character ends standard input.

Header Flags

The following list contains the header flags that can be set with the **-f** flag and unset with the **-d** flag. Header flags control the format of the g-file created with the **get** command.

b Lets you use the **-b** flag of a **get** command to create branch deltas.

c *Number* Makes the *Number* variable the highest release number that a **get -e** command can use. The value of the *Number* variable must be greater than 0 and less than or equal to 9999. (The default value is 9999.)

d *SID* Makes the *SID* variable the default delta supplied to a **get** command.

f *Number* Makes the *Number* variable the lowest release number that a **get -e** command can retrieve. The *Number* variable must be greater than 0 and less than 9999. (The default value is 1.)

i [*String*] Treats the following informational message, issued by the **get** or **delta** command, as an error:

```
There are no SCCS identification keywords in the file. (cm7)
```

In the absence of this flag, the message is only a warning. The message is issued if no SCCS identification keywords are found in the text retrieved or stored in the SCCS file (refer to the **get** command). If a string is supplied, the keywords must match exactly the given string. The string must contain a keyword and have no embedded newlines.

j Permits concurrent **get** commands for editing the same SID of an SCCS file. Use of the **j** header flag allows multiple concurrent updates to the same version of the SCCS file.

l*List* (lowercase L) Locks the releases specified by the *List* variable against editing, so that a **get -e** command against one of these releases fails. The list has the following syntax:

```
<List> : : = <Range> | <List> , <Range>
<Range> : : = SID | a
```

where character `a` in the list is equivalent to specifying all releases for the named SCCS file.

m *Module* Substitutes the *Module* variable for all occurrences of the **59** keyword in an SCCS text file retrieved by a **get** command. The default *Module* variable is the name of the SCCS file without the **s.** prefix. The module name can include MBCS (multibyte character set) characters.

n Causes the **delta** command to create a null delta in any releases that are skipped when a delta is made in a new release. For example, if you make delta 5.1 after delta 2.7, releases 3 and 4 will be null. Releases 3 and 4 will be created as null delta entries in the delta table of the **s.** file. The resulting null deltas can serve as points from which to build branch deltas. Without this flag, skipped releases do not appear in the SCCS file.

q *Text* Substitutes the specified text for all occurrences of the keyword in an SCCS text file retrieved by a **get** command.

t *Type* Substitutes specified type for all keywords in a g-file retrieved by a **get** command.

v [*Program*] Makes the **delta** command prompt for Modification Request (MR) numbers as the reason for creating a delta. The *Program* variable specifies the name of an MR-number validity-checking program. If the **v** flag is set in the SCCS file, the **-m** flag must also be used, even if its value is null. The program name can include MBCS (multibyte character set) characters.

Locating Damaged SCCS Files

Although SCCS provides some error protection, you may need to recover a file that was accidentally damaged. This damage may result from a system malfunction, operator error, or changing an SCCS file without using SCCS commands.

SCCS commands use the checksum to determine whether a file was changed since it was last used. The only SCCS command that processes a damaged file is the **admin** command when used with the **-h** or **-z** flags. The **-h** flag tells the **admin** command to compare the checksum stored in the SCCS file header against the computed checksum. The **-z** flag tells the command to re-compute the checksum and store it in the file header.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Examples

These examples use an imaginary text file called `test.c` and an editor such as **ed** to edit files.

1. First, create an ordinary SCCS file. To create an empty SCCS file named `s.test.c`, enter:

   ```
   $ admin -n s.test.c
   ```

   Using the **admin** command with the **-n** flag creates an empty SCCS file.

2. To convert an existing text file into an SCCS file, enter:

   ```
   $ admin -itest.c s.test.c
   There are no SCCS identification keywords in the file (cm7)
   $ ls
   s.test.c test.c
   ```

   If you use the **-i** flag, the **admin** command creates delta 1.1 from the specified file. Once delta 1.1 is created, rename the original text file so it does not interfere with SCCS commands:

   ```
   $ mv test.c back.c
   ```

   The message `There are no SCCS identification keywords in the file (cm7)` does not indicate an error. SCCS writes this message when there are no identification keywords in the file. Identification keywords are variables that can be placed in an SCCS file. The values of these variables provide information such as date, time, SID, or file name. See the **get** command for an explanation of identification keywords. If no identification keywords exist, SCCS writes the message. However, if the **i** header flag is set in the **s.** file, this message causes an error condition. This flag is set by the user.

   Give the SCCS file any name, beginning with **s.**. In the preceding example, the original file and the SCCS file have the same name, but that is not necessary.

   Because you did not specify a release number, the **admin** command gave the SCCS file an SID of 1.1. SCCS does not use the number 0 to identify deltas. Therefore, a file cannot have an SID of 1.0 or 2.1.1.0, for example. All new releases start with level 1.

3. To start the `test.c` file with a release number of 3.1, use the **-r** flag with the **admin** command, as shown below, and enter:

   ```
   $ admin -itest.c -r3 s.test.c
   ```

   To restrict permission to change SCCS files to a specific set of user IDs, list user IDs or group ID numbers in the user list of the SCCS file by using the **-a** flag of the **admin** command. This flag may appear multiple times on the command line. These IDs then appear in the SCCS file header. Without the **-a** flag to restrict access, all user IDs can change the SCCS files.

4. To restrict edit permission to the user ID `dan`, enter:

```
$ admin -adan s.test.c
```

5. Check SCCS files on a regular basis for possible damage. The easiest way to do this is to run the **admin** command with the **-h** flag on all SCCS files or SCCS directories, as follows:

   ```
   $ admin -h s.file1 s.file2 ...
   $ admin -h directory1 directory2 ...
   ```

   If the **admin** command finds a file where the computed checksum is not equal to the checksum listed in the SCCS file header, it displays this message:

   ```
   ERROR [s. filename]:
   1255-057 The file is damaged. (co6)
   ```

   If a file was damaged, try to edit the file again or read a backup copy. After fixing the file, run the **admin** command with the **-z** flag and the repaired file name:

   ```
   $ admin -z s.file1
   ```

   This operation replaces the old checksum in the SCCS file header with a new checksum based on the current file contents. Other SCCS commands can now process the file.

Files

/usr/bin/admin	Contains the SCCS **admin** command.

Related Information

The **delta** command, **ed** command, **get** command, **prs** command, **sccshelp** command, **what** command.

The **sccsfile** file format.

List of SCCS Commands in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

Source Code Control System (SCCS) Overview in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

---

aixmibd Daemon

Purpose

Provides the AIX Enterprise Management Information Base (MIB) extension subagent, for use with the Simple Network Management Protocol (SNMP) version 3 agent, that collects data from system for variables defined in the AIX Enterprise Specific MIB.

Syntax

aixmibd [**-f** *FileName*] [**-d** *Level*] [**-a** *Host*] [**-c** *Community*]

Description

The AIX Enterprise MIB extension subagent is a daemon, aixmibd, that collects data from system for variables defined in the AIX Enterprise Specific MIB. The subagent receives SNMP requests and sends data via the SNMP-DPI API for communication with the mainAIX snmpd daemon. An Enterprise Management application or other simple application (example snmpinfo command) uses SNMP protocol to get or set AIX MIB objects.

One focus of the subagent is on the data related to the file systems, volume groups, logical volumes, physical volumes, paging space, processes, print queues, print jobs, system users, system groups, users currently logged in, subsystems, subservers, system environment, and various devices.

Another focus of the subagent is on important system traps. Traps, which are also called indications, or notifications, are event reports and are used to decrease the length of time between when the event happens and when it is noticed by a manager so that the event can be handled timely. Traps are generated periodically to report the status change and operating status of the system. From analyzing the data, a manager can determine if a device and the whole system are functioning properly and securely, and make appropriate adjustment. For example, when the **/home** file system reaches the threshold 95% (percent used size), a trap can be generated to report the event to a manager. The manager can respond by sending an email, paging, and so on. To indicate system critical events instantly, a series of traps will be generated by the subagent.

Note: TheAIX enterprise subagent should be started by the System Resource Controller (SRC). Entering `aixmibd` at the command line is not recommended.

Flags

-a *Host*	Causes the request to be sent to the specified host. The host can be an IPv4 address, an IPv6 address, or a host name.
-c *Community*	Specifies the community name.
-d *Level*	Specifies the tracing/debug level. The default level is 56. The debug levels are defined as follows: • 8 = DPI® level 1 • 16 = DPI level 2 • 32 = Internal level 1 • 64 = Internal level 2 • 128 = Internal level 3 Add the numbers to specify multiple trace levels.
-f *File*	Specifies a non-default configuration file.

Examples

1. In order to cause the **aixmibd** subagent to connect to the SNMP agent on the host 'host1' with the community name 'instrum', enter the following:
   ```
   startsrc -s aixmibd -a "-a host1 -c instrum"
   ```
2. Because the **aixmibd** subagent is controlled by SRC, it can be activated by **startsrc**. After the **aixmibd** subagent is activated by **startsrc** in this example, the subagent will connect to the SNMP agent on the host nmsu over TCP with default community name 'public':
   ```
   startsrc -s aixmibd -a "-a nmsu"
   ```

Files

/etc/aixmibd.conf	Contains the configuration file for the **aixmibd** subagent.
/usr/samples/snmpd/aixmibd_security_readme	**/usr/samples/snmpd/aixmibd_security_readme** contains the example configurations for different views and information about related security issues. Also contains information describing how to set the variables in **/etc/aixmibd.conf**.
/usr/samples/snmpd/aixmibd.my	Contains the MIB definitions for the **aixmibd** subagent.

Related Information

The **clsnmp** command, **snmpinfo** command, SNMP version 3 daemon **snmpdv3**, the **snmptrap** command.

---

aixpert Command

Purpose

Aids the system administrator in setting the security configuration.

Syntax

aixpert

aixpert -l h|high | m|medium | l|low | d|default | s|sox-cobit [**-n -o** *filename*] [**-a -o** *filename*] [**-p**]

aixpert -c [**-p**]

aixpert -u [**-p**]

aixpert -d

aixpert [**-f** *filename*] [**-a -o** *filename*] [**-p**]

Description

The **aixpert** command sets a variety of system configuration settings to enable the desired security level. For more information on which setting can be used in a typical environment, see AIX Security Expert.

Running **aixpert** with the only the **-l** flag set implements the security settings promptly without letting the user configure the settings. For example, running **aixpert -l high** applies all the high-level security settings to the system automatically. However, running **aixpert -l** with the **-n -o** *filename* option saves the security settings to a file specified by the *filename* parameter. The **-f** flag then applies the new configurations.

After the initial selection, a menu is displayed itemizing all security configuration options associated with the selected security level. These options can be accepted in whole or individually toggled off or on. After any secondary changes, **aixpert** continues to apply the security settings to the computer system.

Note: It is recommended that **aixpert** be rerun after any major systems changes, such as the installation or updates of software. If a particular security configuration item is deselected when **aixpert** is rerun, that configuration item is skipped.

Flags

-a	The settings with the associated level security options are written in abbreviated file format to the file specified by the **-o** flag. You must specify the **-o** option when you specify the **-a** option.
-p	Specifies that the output of the security rules will be displayed using verbose output. The **-p** option logs the rules processed into the audit subsystem if the auditing is turned on. This option can be used with any of the **-l**, **-u**, **-c** and **-f** options.
-c	Checks the security settings against the previously applied set of rules. If the check against a rule fails, the previous versions of the rule are also checked. This process continues until the check passes, or until all of the instances of the failed rule in the **/etc/security/aixpert/core/appliedaixpert.xml** file are checked.

-f Applies the security settings in the provided *filename*.

For example, the following command writes all of the high-level security options to the **/etc/security/aixpert/core/hls.xml** file:

```
aixpert -l h -n /etc/security/aixpert/core/hls.xml
```

After removing any unwanted options, you can apply these security settings with the following command:

```
aixpert -f /etc/security/aixpert/core/hls.xml
```

When you specify the **-f** option, security settings are consistently applied from system to system by securely transferring and applying an **appliedaixpert.xml** file from system to system.

All the successfully applied rules are written to the **/etc/security/aixpert/core/appliedaixpert.xml** file and the corresponding ″undo″ action rules are written to the **/etc/security/aixpert/core/undo.xml** file.

-l Sets the system security settings to the level specified with this option. This flag has the following options:

h|high Specifies high-level security options.

m|medium
Specifies medium-level security options.

l|low Specifies low-level security options.

d|default
Specifies AIX standards-level security options.

s|sox-cobit
Specifies SOX-COBIT best practices-level security options.

If you specify both the **-l** and **-n** flags, the security settings are not implemented on the system; however, they are only written to the file that you specified in the **-o** flag.

All the successfully applied rules are written to the **/etc/security/aixpert/core/appliedaixpert.xml** file and the corresponding undo action rules are written to the **/etc/security/aixpert/core/undo.xml** file.

Attention: When you use the **d|default** option, the option can overwrite the configured security settings that you previously set through the **aixpert** command or independently, and restores the system to its traditional open configuration.

-n The settings with the associated level security options are written to the file specified by the **-o** flag. You must specify the **-o** option when you use the **-n** option.

-o Stores security output to the file pointed to by *filename*. The output file has its read and write permissions set to root as a security precaution. This file should be protected against unwanted access.

-u Undoes the security settings that have been applied.

-d Displays the document type definition (DTD).

Parameters

filename The output file that stores the security settings. Root permission is required to access this file.

Security

The **aixpert** command is executable only by root.

Examples

1. To write all of the high-level security options to an output file, use the following command:

   ```
   aixpert -l high -n -o /etc/security/aixpert/plugin/myPreferredSettings.xml
   ```

   After completing this command, the output file can be edited, and specific security roles can be commented out by enclosing them in the standard xml comment string (<-- begins the comment and -\> closes the comment).
2. To apply the security settings from a configuration file, use the following command:

   ```
   aixpert -f /etc/security/aixpert/plugin/myPreferredSettings.xml
   ```
3. To check the security settings that have been applied to the system, and to log the rules that failed into the audit subsystem, use the following command:

   ```
   aixpert –c -p
   ```

Location

/usr/sbin/aixpert/	Contains the **aixpert** command.

Files

/etc/security/aixpert/core/aixpertall.xml	Contains an xml listing of all possible security settings. Has -r-------- permissions, and requires root security.
/etc/security/aixpert/core/ appliedaixpert.xml	Contains an xml listing of applied security.
/etc/security/aixpert/log/aixpert.log	Contains a trace log of applied security settings. This does not use syslog. The **aixpert** command writes directly to the file. Has -rw------- permissions, and requires root security.
/etc/security/aixpert/log/firstboot.log	Contains a trace log of the security settings that were applied during the first boot of a Secure by Default (SbD) installation.
/etc/security/aixpert/core/undo.xml	Contains an xml listing of security settings, which can be undone.

Related Information

AIX Security Expert in *Security*.

---

aixpertldap Command

Purpose

Uploads or downloads AIX Security Expert XML configuration files to or from a centralized location on a Light Directory Access Protocol (LDAP) server.

Syntax

aixpertldap -u -D *bindDN* **-w** *bindPwd* [**-b** *baseDN*] [**-f** *filename*] [**-l** *label*]

aixpertldap -d -D *bindDN* **-w** *bindPwd* [**-b** *basedn*]

aixpertldap [**-?**]

Description

The **aixpertldap** command allows a system administrator to store AIX Security Expert XML configuration files in a centralized location on an LDAP server. By sharing these configuration files, similar systems operating in similar environments can easily download these security policies (XML configuration files), and apply the policies with the **aixpert** command. In this way, systems with similar security requirements are configured the same.

When this command downloads the AIX Security Expert security policy configuration files from the LDAP server, these files are placed in the local **/etc/security/aixpert/ldap** directory. The system administrator can scan these files, choose a relevant file, and apply the security settings specified in the file using the **-f** option of the **aixpert** command. Additionally, if you use the Web-based System Manager (websm) to access the AIX Security Expert, the LDAP server is automatically queried for all AIX Security Expert security policy configuration files, after reading the binding distinguished name (specified by the *bindDN* parameter) and the binding password (specified by the *bindPwd* parameter) from the user. These files are presented as options through the websm graphical user interface (GUI) for selection and implementation on the local system.

Tip: With the existing LDAP setup, this command uses the binding distinguished name and the binding password of the running LDAP client to store or retrieve XML configuration files on or from an LDAP server.

Flags

-D *bindDN* Specifies the binding distinguished name to connect to an LDAP server.

-w *bindPwd* Specifies the binding password to read or write XML configuration files from or to an LDAP server.

-b *basedn* Specifies the centralized location where the XML configuration files are stored.

- If you specify the *basedn* parameter while XML files are being uploaded, the XML files are stored under the location specified by the *basedn* parameter; otherwise the files are stored under the location specified by the default *basedn* value: `cn=aixdata`.

  For example, if the *basedn* parameter is specified as `"ou=Austin,o=ibm,c=US"`, the **aixpertldap** command stores the XML configuration files under the `"ou=aixpert,ou=Austin,o=ibm,c=US"` distinguished name (DN).
- If you specify the *basedn* parameter while XML files are being downloaded, the **aixpertldap** command searches under the specific DN for the XML files; otherwise the default *basedn* value (`cn=aixdata`) is used to search the XML files.

  For example, if the *basedn* parameter is not specified, the **aixpertldap** command searches for XML files under the default *basedn* value: `ou=aixpert, ou=aixdata`.

-d Downloads the XML configuration files from an LDAP server to the local **/etc/security/aixpert/ldap** directory.

-f *filename* Specifies the full path of the XML configuration file to be uploaded to an LDAP server.

If you do not specify the option, the **/etc/security/aixpert/core/appliedaixpert.xml** file is uploaded to the LDAP server by default.

Restriction: The **f** and **d** options are mutually exclusive.

-l *label* Specifies the short description of the content in the XML configuration file that is being uploaded. If you do not this option, the XML file has the host name as the label.

For example, if the XML file contains security settings of Accounts department, the label is named AccountsDept.

Restriction: The **l** and **d** options are mutually exclusive.

-u Uploads the XML configuration files to an LDAP server.

-? Displays the usage statement of the command.

Exit Status

0	Success.
1	Failure or partial failure.

Security

Only root users can run the **aixpertldap** command.

Examples

1. To upload the **/home/hussain/netwsec.xml** file under the `ou=aixpert, ou=Bangalore,o=ibm,c=IN` DN with the `NetworkSecurity` label, use the following command:

   ```
   aixpertldap –u –D binddn -w secret –b ou=Bangalore,o=ibm,c=IN
   –f /home/hussain/netwsec.xml –l NetworkSecuriry
   ```

2. To download all XML files from the `ou=aixpert, ou=Bangalore,o=ibm,c=IN` DN to the **/etc/security/aixpert/ldap** directory, use the following command:

   ```
   aixpertldap –d –D binddn -w secret –b ou=Bangalore,o=ibm,c=IN
   ```

3. To download the XML files from the `ou=aixpert, cn=aixdata` DN, use the following command:

   ```
   aixpertldap -d –D binddn -w secret
   ```

Files

/etc/security/aixpert/ldap	Stores the downloaded XML configuration files.

Related Information

The **aixpert** command.

The AIX Security Expert in *Security*.

The Light Directory Access Protocol in *Security*.

---

aixterm Command

Purpose

Initializes an Enhanced X-Windows terminal emulator.

Syntax

aixterm [**-ah**] [**-ar**] [**-autopush**] [**-b** *NumberPixels*] [**-bd** *Color*] [**-bg** *Color*] [**-bw** *NumberPixels*] [**-cc** *CharRange*:*Value* [,...]] [**-cr** *Color*] [**-csd** *CharShape*] [**-cu**] [**-C**] [**-display** *Name*:*Number*] [**-dw**] [**-f0** *Font*] [**-f1** *Font*] [**-f2** *Font*] [**-f3** *Font*] [**-f4** *Font*] [**-f5** *Font*] [**-f6** *Font*] [**-f7** *Font*] [**-f0** *FontSet*] [**-f1** *FontSet*] [**-f2** *FontSet*] [**-f3** *FontSet*] [**-f4** *FontSet*] [**-f5** *FontSet*] [**-f6** *FontSet*] [**-f7** *FontSet*] [**-fb** *Font*] [**-fg** *Color*] [**-fi** *FontSet*] [**-fn** *Font*] [**-fs** *Font*] [**-fullcursor**] [**-geometry** *Geometry*] [**#geometry** *Geometry*] [**-help**] [**-i**] [**-ib** *File*] [**-im** *InputMethod*] [**-j**] [**-keywords**] [**-lang** *Language*] [**-l**] [**-leftscroll**] [**-lf** *File*] [**-ls**] [**-mb**] [**-mc** *Number*] [**-ms** *Color*] [**-mn**] [**-n** *IconName*] [**-name** *Application*] [**-nb** *Number*] [**-nobidi**] [**-nonulls**] [**-nss** *NumShape*] [**-orient** *Orientation*] [**-outline** *Color*] [**-po** *Number*] [**-ps**] [**-pt** *Preedit*] [**-reduced**] [**-rfb** *Font*] [**-rfi** *Font*] [**-rfn** *Font*] [**-rfs** *Font*] [**-rf0** *Font*] [**-rf1** Font] [**-rf2** Font] [**-rf3** Font] [**-rf4** Font] [**-rf5** Font] [**-rf6**Font] [**-rf7** Font] [**-rf0** *FontSet*] [**-rf1** *FontSet*] [**-rf2** *FontSet*] [**-rf3** *FontSet*] [**-rf4** *FontSet*] [**-rf5** *FontSet*] [**-rf6** *FontSet*] [**-rf7** *FontSet*] [**-rv**] [**-rw**] [**-s**] [**-sb**] [**-sf**] [**-si**] [**-sk**] [**-sl** *NumberLines*] [**-sn**] [

-st] [**-suppress**] [**-symmetric**] [**-T** *Title*] [**-text** *TextType*] [**-ti**] [**-tm** *String*] [**-tn** *TerminalName*] [**-ut**] [**-v**] [**-vb**] [**-W**] [**-xrm** *String*] [**-132**] [**-e** *Command*]

Description

The **aixterm** command provides a standard terminal type for programs that do not interact directly with Enhanced X-Windows. This command provides an emulation for a VT102 terminal or a high function terminal (HFT). The VT102 mode is activated by the **-v** flag.

The **aixterm** command supports the display for up to 16 colors at a time.

The **aixterm** terminal supports escape sequences that perform terminal functions such as cursor control, moving and deleting lines, and **aixterm** private functions.

Many of the special **aixterm** terminal features (like the scroll bar) can be modified under program control through a set of private **aixterm** command escape sequences. You can also use escape sequences to change the title in the title bar.

There are three different areas in the **aixterm** window:

- Scroll bar
- Status line
- Terminal window.

By default, only the terminal window is initially displayed.

The terminal window is the area provided for terminal emulation. When you create a window, a pseudo terminal is allocated and a command (usually a shell) is started.

The **aixterm** command automatically highlights the window border and the text cursor when the mouse cursor enters the window (selected) and unhighlights them when the mouse cursor leaves the window (unselected). If the window is the focus window, the window is highlighted regardless of the location of the mouse cursor. Any window manager, as in the case of the AIXwindows Window Manager (MWM), can cover the **aixterm** border, and the highlight and border color do not show.

The **WINDOWID** environment variable is set to the resource ID number of the **aixterm** window.

When running in an **aixterm** window, the **TERM** environment variable should be **TERM=aixterm**.

The **TERM** environment variable on your home machine determines what the **TERM** environment variable should be on the remote machine (unless it is overridden by your **.profile**).

When you use the **rlogin**, **tn**, or **rsh** commands to login to a different machine, the **TERM** environment variable should be set to **aixterm**. If this operation does not occur, you can perform the following two command line operations:

1. **TERM=aixterm**
2. **export TERM**

If commands (for example, the **vi** command) do not recognize the term type **aixterm** when you login to another system, perform the following one-time operation on the remote system:

1. **su**
2. **cd/tmp**
3. **mkdir** *Xxxxx*
4. **cd** *Xxxxx*
5. **ftp** *LocalSystemName*

6. **cd /usr/share/lib/terminfo**
7. **get ibm.ti**
8. **quit**
9. **TERMINFO=/tmp/***Xxxxx*
10. **export TERMINFO**
11. **tic ibm.ti**
12. **ls**
13. **ls a**
14. **mkdir /usr/share/lib/terminfo/a**
15. **cp a/aixterm* /usr/share/lib/terminfo/a**
16. **cd /tmp**
17. **rm -r /tmp/***Xxxxx*
18. **exit**
19. On the remote machine, enter the following:
    a. **TERM=aixterm**
    b. **export TERM**

Arabic/Hebrew Support

The **aixterm** command supports bidirectional languages such as Arabic and Hebrew. This command can open a window to be used with Arabic/Hebrew applications. You can create an Arabic/Hebrew window by specifying an Arabic or Hebrew locale (**ar_AA**, **Ar_AA**, **iw_IL**, or **lw_IL**) with the **-lang** flag or by predefining an Arabic or Hebrew locale from SMIT for the system. You can also use the Web-based System Manager **wsm system** fast path and selecting the **Cultural Environment** icon.

The Arabic/Hebrew window supports bidirectional text display. Thus, English and Arabic or Hebrew text can be displayed on the same line. There are different aspects in the Arabic/Hebrew window:

- Screen Orientation
- Text mode
- Character shaping
- Numeric representation
- Status line

Screen Orientation: The screen orientation in an Arabic/Hebrew window can be either left-to-right or right-to-left. The default orientation is left-to-right unless otherwise specified with a flag or in the **.Xdefaults** file. While the window is active, you can reverse the screen orientation using special key combinations. You can reverse the screen orientation according to your needs.

Text Mode: An Arabic/Hebrew window supports two text modes and their corresponding manipulation:

- Implicit
- Visual

In the implicit text mode, characters are stored in same order that they are entered. The text is transformed into its visual form only when it is displayed. In the visual text mode, characters are stored in the same way that they are displayed on the window.

Character Shaping: The Arabic/Hebrew window represents Arabic and Hebrew texts differently, according to its context. Text is represented in one of the following forms:

- Automatic
- Isolated
- Initial

- Middle
- Final

Arabic/Hebrew can also be shaped according to the passthru mode. For more information on character shaping, see ″Character Shaping″ in *AIX 5L Version 5.3 National Language Support Guide and Reference*.

Numeric Representation: Numerics can be represented in Arabic numerals, Hindi numerals, or in passthru mode. In implicit text mode, numerals can also be represented according to their contextual form. Thus, Arabic numbers can be displayed in English text or Hindi numbers can be displayed in Arabic text.

Status Line: The Arabic/Hebrew window can display an optional status line that shows the current status of the window. The status line contains the following values:

Value	Current Setting
E	English language
N	National language
SCR->	Left-to-right screen orientation
<-SCR	Right-to-left screen orientation
alef	Auto shape mode
blank	Passthru shaping mode
ghain	Displayed in the currently used shaping mode
I	Implicit text mode
V	Visual text mode
U	Context numbers
A	Arabic numbers
H	Hindi numbers
P	Passthru for numbers

Note: Use the implicit text mode (the default text mode) for more efficient data sorting.

Use the following key combinations in an Arabic/Hebrew window to change certain settings.

Key Combination	Purpose
Alt + Enter	Reverses screen direction.
Alt + Right Shift	Enables Arabic/Hebrew keyboard layer.
Alt + Left Shift	Enables English keyboard layer.

For Implicit Mode only:

Alt + Kpd*	Adjusts the column heading.

For Visual Mode only:

Alt + Kpd 1	Shapes characters in their initial form.
Alt + Kpd 2	Shapes characters in their isolated form.
Alt + Kpd 3	Shapes characters in their passthru form.
Alt + Kpd 4	Shapes characters automatically (Valid also for Implicit).
Alt + Kpd 7	Shapes characters in their middle form.
Alt + Kpd 8	Shapes characters in their final form.
Shift + Kpd /	Toggles the Push Mode (Push/End Push).
Alt + Kpd /	Toggles the Autopush function.

For more information on the Autopush function, the Push/End Push function, or other Arabic/Hebrew functions, see the **telnet,tn or tn3270** command.

Using the aixterm Command Data-Stream Support

The following is a list of the escape sequences supported by the **aixterm** command.

Some escape sequences activate and deactivate an alternate screen buffer that is the same size as the display area of the window. This capability allows the contents of the screen to be saved and restored. When the alternate screen is activated, the current screen is saved and replaced with the alternate screen. Saving lines scrolled off of the window is disabled until the original screen is restored.

The following table uses these abbreviations in the right hand column:

Xv Supported by the **aixterm** command running in VT100 mode.

Xh Supported by the **aixterm** command running in HFT mode.

H Found in the HFT data stream.

V Found in the VT100 data stream.

Name	
BEL	**Function (single-byte control)** Bell **Data Stream** 0x07 **Support** Xv, Xh, H, V
BS	**Function (single-byte control)** Backspace **Data Stream** 0x08 **Support** Xv, Xh, H, V
HT	**Function (single-byte control)** Horizontal tab **Data Stream** 0x09 **Support** Xv, Xh, H, V
LF	**Function (single-byte control)** Linefeed **Data Stream** 0x0A **Support** Xv, Xh, H, V
VT	**Function (single-byte control)** Vertical tab **Data Stream** 0x0B **Support** Xv, Xh, H, V

FF	**Function (single-byte control)** Form feed **Data Stream** 0x0C **Support** Xv, Xh, H, V
CR	**Function (single-byte control)** Carriage return **Data Stream** 0x0D **Support** Xv, Xh, H, V
SO	**Function (single-byte control)** Shift out **Data Stream** 0x0E **Support** Xv, Xh, H, V
SI	**Function (single-byte control)** Shift in **Data Stream** 0x0F **Support** Xv, Xh, H, V
DCI	**Function (single-byte control)** Device control 1 **Data Stream** 0x11 **Support** H, V
DC3	**Function (single-byte control)** Device control 3 **Data Stream** 0x13 **Support** H, V
CAN	**Function (single-byte control)** Cancel **Data Stream** 0x18 **Support** H, V

SUB	**Function (single-byte control)** Substitute (also cancels) **Data Stream** 0x1A **Support** H, V
ESC	**Function (single-byte control)** Escape **Data Stream** 0x1B **Support** Xv, Xh, H, V
SS4	**Function (single-byte control)** Single Shift 4 **Data Stream** 0x1C **Support** H
SS3	**Function (single-byte control)** Single Shift 3 **Data Stream** 0x1D **Support** H
SS2	**Function (single-byte control)** Single Shift 2 **Data Stream** 0x1E **Support** H
SS1	**Function (single-byte control)** Single Shift 1 **Data Stream** 0x1F **Support** H
cbt	**Function (single-byte control)** cursor back tab **Data Stream** ESC [Pn Z **Support** Xv, Xh, H

cha	**Function (single-byte control)** cursor horizontal absolute **Data Stream** ESC [Pn G **Support** Xv, Xh, H
cht	**Function (single-byte control)** cursor horizontal tab **Data Stream** ESC [Pn I **Support** H
ctc	**Function (single-byte control)** cursor tab stop control **Data Stream** ESC [Pn W **Support** H
cnl	**Function (single-byte control)** cursor next line **Data Stream** ESC [Pn E **Support** H
cpl	**Function (single-byte control)** cursor preceding line **Data Stream** ESC [Pn F **Support** Xv, Xh, H
cpr	**Function (single-byte control)** cursor position report **Data Stream** ESC [Pl; Pc R **Support** Xv, Xh, H, V
cub	**Function (single-byte control)** cursor backward **Data Stream** ESC [Pn D **Support** Xv, Xh, H, V

cud	**Function (single-byte control)** cursor down **Data Stream** ESC [Pn B **Support** Xv, Xh, H, V
cuf	**Function (single-byte control)** cursor forward **Data Stream** ESC [Pn C **Support** Xv, Xh, H, V
cup	**Function (single-byte control)** cursor position **Data Stream** ESC [Pl; PC H **Support** Xv, Xh, H, V
cuu	**Function (single-byte control)** cursor up **Data Stream** ESC [Pn A **Support** Xv, Xh, H, V
cvt	**Function (single-byte control)** cursor vertical tab **Data Stream** ESC [Pn Y **Support** H
da1	**Function** Device attributes • request (host to vt100) • response (vt100 to host) **Data Stream** • For a request, ESC [c • For a request, ESC [0 c • For a response, ESC [? 1 ; 2 c **Support** Xv, Xh, V

dch	**Function (single-byte control)** delete character **Data Stream** ESC [Pn P **Support** Xv, Xh, H
decaln	**Function (single-byte control)** screen alignment display **Data Stream** ESC # 8 **Support** Xv, Xh, V
deckpam	**Function (single-byte control)** keypad application mode **Data Stream** ESC = **Support** Xv, V
deckpnm	**Function (single-byte control)** keypad numeric mode **Data Stream** ESC > **Support** Xv, V
decrc	**Function (single-byte control)** restore cursor & attributes **Data Stream** ESC 8 **Support** Xv, Xh, V
decsc	**Function (single-byte control)** save cursor & attributes **Data Stream** ESC 7 **Support** Xv, Xh, V
decstbm	**Function (single-byte control)** set top & bottom margins **Data Stream** ESC [Pt; Pb r **Support** Xv, Xh, V

dl	**Function (single-byte control)** delete line **Data Stream** ESC [Pn M **Support** Xv, Xh, H
dsr	**Function (single-byte control)** device status report **Data Stream** ESC [Ps n **Support** • 0 response from vt100: ready—Xv, Xh, V • 5 command from host: please report status—Xv, Xh, V • 6 command from host: report active position—Xv, Xh, H, V • 13 error report sent from virtual terminal to host—H
dmi	**Function (single-byte control)** disable manual input **Data Stream** ESC ` (back quote) **Support** H
emi	**Function (single-byte control)** enable manual input **Data Stream** ESC b **Support** H
ea	**Function (single-byte control)** erase area **Data Stream** ESC [Ps O **Support** • 0 erase to end of area—Xv, Xh, H • 1 erase from area start—Xv, Xh, H • 2 erase all of area—Xv, Xh, H

ed	**Function (single-byte control)** erase display **Data Stream** ESC [Ps J **Support** • 0 erase to end of display—Xv, Xh, H, V • 1 erase from display start—Xv, Xh, H, V • 2 erase all of display—Xv, Xh, H, V
ef	**Function (single-byte control)** erase field-e,s,all **Data Stream** ESC [Ps N **Support** • 0 erase to end of field—Xv, Xh, H • 1 erase from field start—Xv, Xh, H • 2 erase all of field—Xv, Xh, H
el	**Function (single-byte control)** erase line **Data Stream** ESC [Ps K **Support** • 0 erase to end of line—Xv, Xh, H, V • 1 erase from line start—Xv, Xh, H, V • 2 erase all of line—Xv, Xh, H, V
ech	**Function (single-byte control)** erase character **Data Stream** ESC [Pn X **Support** Xv, Xh, H
hts	**Function (single-byte control)** horizontal tab stop **Data Stream** ESC H **Support** Xv, Xh, H, V

hvp	**Function (single-byte control)** horizontal and vertical position **Data Stream** ESC [Pl; Pc f **Support** Xv, Xh, H, V
ich	**Function (single-byte control)** insert character **Data Stream** ESC [Pn @ **Support** Xv, Xh, H
il	**Function (single-byte control)** insert line **Data Stream** ESC [Pn L **Support** Xv, Xh, H
ind	**Function (single-byte control)** index **Data Stream** ESC D **Support** Xv, Xh, H, V
ls2	**Function (single-byte control)** lock shift G2 **Data Stream** ESC n **Support** Xv
ls3	**Function (single-byte control)** lock shift G2 **Data Stream** ESC o **Support** Xv
nel	**Function (single-byte control)** next line **Data Stream** ESC E **Support** Xv, Xh, H, V

ksi	**Function (single-byte control)** keyboard status information **Data Stream** ESC [Ps p **Support** H
pfk	**Function (single-byte control)** PF key report **Data Stream** ESC [Pn q **Support** Xh, H
rcp	**Function (single-byte control)** restore cursor position **Data Stream** ESC [u **Support** Xv, Xh, H
ri	**Function (single-byte control)** reverse index **Data Stream** ESC M **Support** Xv, Xh, H, V
ris	**Function (single-byte control)** reset to initial state **Data Stream** ESC c **Support** Xv, Xh, H, V
rm	**Function (single-byte control)** reset mode, restore mode, save mode **Data Stream** • reset mode, ANSI specified modes (see sm)—ESC [Ps;...;Ps • reset mode, other private modes and XTERM private modes (see sm)—ESC [? Ps;...;Ps l • restore mode, other private modes and XTERM private modes (see sm)—ESC [? P;...;Ps r • save mode, other private modes and XTERM private modes (see sm)—ESC [? Ps;...;Ps s

sapv	**Function** select alternate presentation variant • 0 set default values for BIDI • 1 set Arabic numeric shapes • 2 set Hindi numeric shapes • 3 set symmetric swapping mode for directional characters • 5 the following graphic character is presented in its isolated form (Arabic only) • 6 the following graphic character is presented in its initial form (Arabic only) • 7 the following graphic character is presented in its middle form (Arabic only) • 8 the following graphic character is presented in its final form (Arabic only) • 13 set Special shaping mode • 14 set standard shaping mode • 15 reset symmetric mode • 18 Passthru (everything) • 19 Passthru (everything except numbers) • 20 Contextual numbers (device dependent) • 21 lock 5, 6, 7, 8 • 22 unlock • 23 set the nonull mode • 24 reset the nonull mode • Values 5-8 affect only the following character unless used with values 21 or 22 **Data Stream** ESC [Psl;...Psn] **Support** Xh
scp	**Function (single-byte control)** save cursor position **Data Stream** ESC [s **Support** Xv, Xh, H

scs	**Function (single-byte control)** select character set • United Kingdom Set • ASCII Set (USASCII) • special graphics **Data Stream** United Kingdom Set: • ESC (A (GO) • ESC) A (G1) • ESC * A (G2) • ESC + A (G3) ASCII Set (USASCII): • ESC (B (GO) • ESC) B (G1) • ESC * B (G2) • ESC + B (G3) special graphics: • ESC (0 (GO) • ESC) 0 (G1) • ESC * 0 (G2) • ESC + 0 (G3) **Support** Xv, V
sd	**Function (single-byte control)** scroll down **Data Stream** ESC [Pn T **Support** H
sl	**Function (single-byte control)** scroll left **Data Stream** ESC [Pn Sp @ **Support** H
spd	**Function (single-byte control)** select screen direction • 0 turn screen to left-to-right, set to Latin keyboard • 1 turn screen direction to right-to-left set to National keyboard **Data Stream** ESC [Ps1;1 S **Support** Xh

sr	**Function (single-byte control)** scroll right **Data Stream** ESC [Pn Sp A **Support** H
srs	**Function (single-byte control)** select reversed string • 0 end push • 1 start push **Data Stream** ESC [Ps[**Support** Xh
ss2	**Function (single-byte control)** single shift G2 **Data Stream** ESC N **Support** Xv
ss3	**Function (single-byte control)** single shift G3 **Data Stream** ESC O **Support** Xv
su	**Function (single-byte control)** scroll up **Data Stream** ESC [Pn S **Support** Xv, Xh, H

sgr	**Function (single-byte control)** set graphic rendition **Data Stream** ESC [Ps m **Support** • 0 normal—Xv, Xh, H, V • 1 bold—Xv, Xh, H, V • 4 underscore—Xv, Xh, H, V • 5 blink (appears as bold)—Xv, Xh, H, V • 7 reverse—Xv, Xh, H, V • 8 invisible—Xh, H • 10..17 fonts—Xh, H • 30..37 foreground colors—Xh, H • 40..47 background colors—Xh, H • 90..97 foreground colors—Xh, H • 100..107 background colors—Xh, H
sg0a	**Function (single-byte control)** set GO character set **Data Stream** ESC (< **Support** Xh, H
sg1a	**Function (single-byte control)** set G1 character set **Data Stream** ESC) < **Support** Xh, H

sm	**Function (single-byte control)** set mode • ANSI specified modes • Other private modes **Data Stream** • ANSI specified modes—ESC [Ps;...;Ps h • Other private modes—ESC [? Ps;...;Ps h **Support** • (ANSI) 4 IRM insert mode—Xv, Xh, H • (ANSI) 12 SRM send/rec mode—H • (ANSI) 18 TSM tab stop mode—H • (ANSI) 20 LNM linefeed/newline—Xv, Xh, H, V • 1 normal/application cursor—Xv, V • 3 80/132 columns—Xv, Xh, V • 4 smooth/jump scroll—Xv, Xh, V • 5 reverse/normal video—Xv, Xh, V • 6 origin/normal—Xv, Xh, V • 7 on/off autowrap—Xv, Xh, H, V • 8 on/off autorept—Xv, Xh, V • 21 CNM CR-NL—H • (XTERM) 40 132/80 column mode—Xv, Xh • (XTERM) 41 curses(5) fix—Xv, Xh • (XTERM) 42 hide/show scroll bar—Xv, Xh • (XTERM) 43 on/off save scroll text—Xv, Xh • (XTERM) 44 on/off margin bell—Xv, Xh • (XTERM) 45 on/off reverse wraparound—Xv, Xh • (XTERM) 47 alternate/normal screen buffer—Xv, Xh • (XTERM) 48 reverse/normal status line—Xv, Xh • (XTERM) 49 page/normal scroll mode—Xv, Xh
tbc	**Function (single-byte control)** tabulation clear **Data Stream** ESC [Ps g (default Ps =0) **Support** • 0 clear horizontal tab stop at active position—Xv, Xh, H, V • 1 vertical tab at line indicated by cursor—H • 2 horizontal tabs on line—H • 3 all horizontal tabs—Xv, Xh, H, V • 4 all vertical tabs—H

VTD	**Function (single-byte control)** virtual terminal data **Data Stream** ESC [x **Support** Xv, Xh, H
VTL	**Function (single-byte control)** virtual terminal locator report **Data Stream** ESC [y **Support** Xh, H
VTR	**Function (single-byte control)** vt raw keyboard input **Data Stream** ESC [w **Support** Xh, H
vts	**Function (single-byte control)** vertical tab stop **Data Stream** ESC I **Support** H
xes	**Function (single-byte control)** erase status line **Data Stream** ESC [? E **Support** Xv, Xh
xrs	**Function (single-byte control)** return from status line **Data Stream** ESC [? F **Support** Xv, Xh
xhs	**Function (single-byte control)** hide status line **Data Stream** ESC [? H **Support** Xv, Xh

xss	**Function (single-byte control)** show status line **Data Stream** ESC [? S **Support** Xv, Xh
xgs	**Function (single-byte control)** go to column of status line **Data Stream** ESC [? Ps T **Support** Xv, Xh
xst	**Function (single-byte control)** set text parameters • 0 change window name and title to Pt • 1 sets only the icon name • 2 sets only the title name • Everything between ESC-P and ESC\ is ignored. aixterm will work as usual after the ESC\. **Data Stream** ESC] Ps ; Pt \007 **Support** Xv, Xh

Copy, Paste, and Re-execute Functions

When you create a terminal window, the **aixterm** command allows you to select text and copy it within the same window or other windows by using copy, paste, and re-execute button functions. These text functions are available in HFT and VT102 emulations. The selected text is highlighted while the button is pressed.

The copy, paste, and re-execute button functions perform as follows:

Copy

The left button is used to save text into the cut buffer. The **aixterm** command does a text cut, not a box cut. Move the cursor to beginning of the text, hold the button down while moving the cursor to the end of the region, and release the button. The selected text is highlighted and saved in the global cut buffer and made the **PRIMARY** selection when the button is released.

- Double clicking selects by words.
- Triple clicking selects by lines.
- Quadruple clicking goes back to characters, and so on.

Multiple clicking is determined from the time the button is released to the time the button is pressed again, so you can change the selection unit in the middle of a selection.

The right button extends the current selection. If you press this button while moving closer to the right edge of the selection than the left, it extends or contracts the right edge of the selection. If you contract the selection past the left edge of the selection, the **aixterm** command assumes you really meant the left edge, restores the original selection, and extends or contracts the left edge of the selection. Extension starts in the selection unit mode that the last selection or extension was performed in; you can multiple click to cycle through them.

Paste Pressing both buttons at once (or the middle button on a three-button mouse) displays (pastes) the text from the **PRIMARY** selection or from the cut buffer into the terminal window that contains the mouse cursor, inserting it as keyboard input.

Re-execute Pressing the Shift key and the left mouse button takes the text from the cursor (at button release) through the end of the line (including the new line), saves it in the global cut buffer and immediately retypes the line, inserting it as keyboard input. The selected text is highlighted. Moving the mouse cursor off of the initial line cancels the selection. If there is no text beyond the initial cursor point, the **aixterm** command sounds the bell, indicating an error.

By cutting and pasting pieces of text without trailing new lines, you can take text from several places in different windows and form a command to the shell. For example, you can take output from a program and insert it into your favorite editor. Since the cut buffer is globally shared among different applications, you should regard it as a file whose contents you know. The terminal emulator and other text programs should treat it as if it were a text file, that is, the text is delimited by new lines.

Menu Usage

The **aixterm** command has two different menus:

- Options
- Modes

Each menu pops up under the correct combinations of key and button presses. Most menus are divided into two sections that are separated by a horizontal line. The top portion contains various modes that can be altered. A check mark is displayed next to a mode that is currently active. Selecting one of these modes toggles its state. The bottom portion of the menu provides the command entries; selecting one of these performs the indicated function.

The Options menu pops up when the Ctrl key and the left mouse button are pressed simultaneously while the mouse cursor is in a window. The menu contains items that apply to all emulation modes.

The Modes menu sets various modes for each emulation mode. The menu is activated by pressing the Ctrl key and the middle mouse button at the same time, while the mouse cursor is in the window. In the command section of this menu, the soft reset entry resets the scroll regions. This is convenient when a program leaves the scroll regions set incorrectly. The full reset entry clears the screen, resets tabs to every eight columns, and resets the terminal modes (such as wrap and smooth scroll) to their initial states after the **aixterm** command finishes processing the command-line options. When the Auto Linefeed option is turned on, a carriage return is added when a carriage return, vertical tab, or form feed is received. The shells generally do this for the linefeed, but not for the vertical tab or form feed.

Scroll Bar

The **aixterm** command supports an optional scroll bar composed of a scroll button that displays at the top of the scroll bar and a scroll region that displays at the bottom. The scroll bar is hidden until you request it to display.

The scroll region displays the position and amount of text currently showing in the window (highlighted) relative to the amount of text actually saved in the scrolling buffer. As more text is saved in the scrolling buffer (up to the maximum), the size of the highlighted area decreases.

The scroll button causes the window to scroll up and down within the saved text. Clicking the right button moves the window position up (the text scrolls downward); clicking the left button moves the window position down (the text scrolls upward). The amount of scrolling is modified by the Shift and Ctrl keys. If neither key is pressed, the window scrolls a single line at a time. Pressing the Shift key causes the text to scroll a full window at a time, minus one line. Pressing the Ctrl key causes the text to be positioned at the extreme top or bottom of the file.

Character Classes

Clicking the left mouse button (the copy function) twice in rapid succession causes all characters of the same class (that is, letters, white space, punctuation, and so on) to be selected. Because people have different preferences for what should be selected (for example, if file names be selected as a whole or only the separate subnames), you can override the default mapping by using the **charClass** (class **CharClass**) resource.

The **charClass** resource is a list of *CharRange*:*Value* pairs where the range is either a single number or a low-to-high number in the range of 0 to 127, corresponding to the ASCII code for the character or characters to be set. The value is arbitrary, although the default table uses the character number of the first character occurring in the set.

The default table is as follows:

```
static int charClass[128] = {
/* NUL  SOH  STX  ETX  EOT  ENQ  ACK  BEL */
   32,   1,   1,   1,   1,   1,   1,   1,
/*  BS   HT   NL   VT   NP   CR   SO   SI */
    1,  32,   1,   1,   1,   1,   1,   1,
/* DLE  DC1  DC2  DC3  DC4  NAK  SYN  ETB */
    1,   1,   1,   1,   1,   1,   1,   1,
/* CAN   EM  SUB  ESC   FS   GS   RS   US */
    1,   1,   1,   1,   1,   1,   1,   1,
/*  SP   !    "    #    $    %    &    ' */
   32,  33,  34,  35,  36,  37,  38,  39,
/*  (    )    *    +    ,    -    .    / */
   40,  41,  42,  43,  44,  45,  46,  47,
/*  0    1    2    3    4    5    6    7 */
   48,  48,  48,  48,  48,  48,  48,  48,
/*  8    9    :    ;    <    =    >    ? */
   48,  48,  58,  59,  60,  61,  62,  63,
/*  @    A    B    C    D    E    F    G */
   64,  48,  48,  48,  48,  48,  48,  48,
/*  H    I    J    K    L    M    N    O */
   48,  48,  48,  48,  48,  48,  48,  48,
/*  P    Q    R    S    T    U    V    W */
   48,  48,  48,  48,  48,  48,  48,  48,
/*  X    Y    Z    [    \    ]    ^    _ */
   48,  48,  48,  91,  92,  93,  94,  48,
/*  `    a    b    c    d    e    f    g */
   96,  48,  48,  48,  48,  48,  48,  48,
/*  h    i    j    k    l    m    n    o */
   48,  48,  48,  48,  48,  48,  48,  48,
/*  p    q    r    s    t    u    v    w */
   48,  48,  48,  48,  48,  48,  48,  48,
/*  x    y    z    {    |    }    ~   DEL */
   48,  48,  48, 123, 124, 125, 126,   1};
```

For example, the string "33:48,37:48,45-47:48,64:48" indicates that the ! (exclamation mark), % (percent sign), - (dash), . (period), / (slash), and & (ampersand) characters should be treated the same way as characters and numbers. This is very useful for cutting and pasting electronic mailing addresses and UNIX file names.

Key Translations

It is possible to rebind keys (or sequences of keys) to arbitrary strings for input. Changing the translations for events other than key and button events is not expected, and causes unpredictable behavior.

The actions available for key translations are as follows:

insert()
Processes the key in the normal way (that is, inserts the ASCII character code corresponding to the keysym found in the keyboard mapping table into the input stream).

string(*String***)**
Rebinds the key or key sequence to the string value; that is, inserts the string argument into the input stream. Quotation marks are necessary if the string contains white space or non-alphanumeric characters. If the string argument begins with the characters ``0x,'' it is interpreted as a hex character constant and the corresponding character is sent in the normal way.

keymap(*Name***)**
Takes a single string argument naming a resource to be used to dynamically define a new translation table; the name of the resource is obtained by appending the string Keymap to *Name*. The keymap name **None** restores the original translation table (the very first one; a stack is not maintained). Uppercase and lowercase is significant.

insert-selection(*Name*[,*Name*]...**)**
Retrieves the value of the first (leftmost) named selection that exists and inserts the value into the input stream. The *Name* parameter is the name of any selection, for example, **PRIMARY** or **SECONDARY**. Uppercase and lowercase is significant.

For example, a debugging session might benefit from the following bindings:

```
*aixterm.Translations: #override <Key>F13: keymap(dbx)
*aixterm.dbxKeymap.translations:\
<Key>F14: keymap(None) \n\
<Key>F17: string("next") string(0x0d) \n\
<Key>F18: string("step") string(0x0d) \n\
<Key>F19: string("continue") string(0x0d) \n\
<Key>F20: string("print") insert-selection(PRIMARY)
```

Key and Button Bindings

The key and button bindings for selecting text, pasting text, and activating the menus are controlled by the translation bindings. In addition to the actions listed in the Key Translations section, the following actions are available:

mode-menu()
Posts one of the two mode menus, depending on which button is pressed.

select-start()
Deselects any previously selected text and begins selecting new text.

select-extend()
Continues selecting text from the previous starting position.

start-extend()
Begins extending the selection from the farthest (left or right) edge.

select-end(*Name*[**,***Name*]...**)**

	Ends the text selection. The *Name* parameter is the name of a selection into which the text is to be copied. The **aixterm** command asserts ownership of all the selections named. Uppercase and lowercase is significant.
ignore()	Quietly discards the key or button event.
bell([*Volume*]**)**	Rings the bell at the specified volume increment above or below the base volume.

The default bindings are:

```
static char defaultTranslations =
"                    <KeyPress>:  insert() \n\
~Shift Ctrl   ~Meta  <Btn1Down>:   mode-menu(options) \n\
~Shift Ctrl   ~Meta  <Btn2Down>:   mode-menu() \n\
~Shift Ctrl   ~Meta  <Btn3Down>:   mode-menu(modes) \n\
~Shift ~Ctrl  ~Meta  <Btn1Down>:   select-start() \n\
~Shift ~Ctrl  ~Meta  <Btn1Motion>: select-extend() \n\
~Shift ~Ctrl  ~Meta  <Btn1Up>:     select-end(PRIMARY)\n\
~Shift ~Ctrl  ~Meta  <Btn2Down>:   ignore() \n\
~Shift ~Ctrl  ~Meta  <Btn2Up>:     insert-selection(PRIMARY)\n\
~Shift ~Ctrl  ~Meta  <Btn3Down>:   start-extend() \n\
~Shift ~Ctrl  ~Meta  <Btn3Motion>: select-extend() \n\
~Shift ~Ctrl  ~Meta  <Btn3Up>:     select-end(PRIMARY)\n\
Shift ~Ctrl   ~Meta  <Btn1Down>:   reexecute() \n\
Shift ~Ctrl   ~Meta  <Btn1Motion>: select-extend() \n\
Shift ~Ctrl   ~Meta  <Btn1Up>:     select-end(PRIMARY)\n\
Shift ~Ctrl   ~Meta  <Btn2Down>:   select-start() \n\
Shift ~Ctrl   ~Meta  <Btn2Motion>: select-extend() \n\
Shift ~Ctrl   ~Meta  <Btn2Up>:     select-end(PRIMARY)\n\
Shift ~Ctrl   ~Meta  <Btn3Down>:   ignore() \n\
Shift ~Ctrl   ~Meta  <Btn3Up>:     insert-selection(PRIMARY)\n\
Shift Ctrl    ~Meta  <BtnDown>:    size(toggle) \n\
Shift Ctrl    ~Meta  <BtnUp>:      ignore() \n\
                     <BtnDown>:    bell(0) \n\
                     <BtnUp>:      bell(0) \n\
";
```

aixterm Command Internationalization (I18N)

To run an aixterm with a different keyboard layout than the X server's (such as a French keyboard layout on a Swiss German X server), run the following commands:

1. Change the X server to a French keyboard:
   ```
   xmodmap /usr/lpp/X11/defaults/xmodmap/Fr_FR/keyboard
   ```
2. Set the locale environment variable to Fr_FR using one of the following:
   - For Korn shells: `export LANG=Fr_FR`
   - For C shells: `setenv LANG Fr_FR`
   - For Bourne shells: `LANG=Fr_FR; export LANG`
3. Start an aixterm terminal emulator:
   ```
   aixterm &
   ```
4. Reset the X server's keyboard file to its original language:
   ```
   xmodmap /usr/lpp/X11/defaults/xmodmap/Gr_SW/keyboard
   ```

The **aixterm** command continues to use the keyboard layout that the X server was using when the aixterm started. It ignores **KeymapNotify** by default.

The **aixterm** command uses the Input Method to convert the X server's keysyms into either printable characters or nonprintable escape strings such as function keys. The Input Method uses its own keymap files, in **/usr/lib/nls/loc**, to convert X keysyms into code points for the printable characters, and escape strings for nonprintable characters. There is a keymap file for each language and one keymap file for

escape sequences. The escape sequences are in **C@outbound.imkeymap**; the source is **C@outbound.imkeymap.src**. The other keymap files begin with the locale name and look like: **locale.imkeymap** and **locale.codeset.imkeymap**. For example:

US English in codeset IBM-850	En_US.IBM-850.imkeymap
US English in codeset ISO8859-1	en_US.ISO8859-1.imkeymap
Turkish in codeset ISO8859-9	tr_TR.ISO8859-9.imkeymap
Japanese in codeset IBM-932	Ja_JP.IBM-932.imkeymap
Japanese in codeset IBM-943	Ja_JP.IBM-943.imkeymap
Japanese in codeset EUC(JP)	ja_JP.IBM-eucJP.imkeymap

The following dependencies apply:

- You can change the locale by entering the following SMIT fast path: `smit mle_sel_menu`, or by using the Web-based System Manager **wsm system** fast path and selecting the **Cultural Environment** icon. You can also change the locale temporarily by modifying the LANG environment variable.
- You can change the system keyboard definition by selecting the following SMIT menu items: System Environments, Manage Language Environment, and Change the Keyboard Map for the Next System Restart, or by using the Web-based System Manager **wsm system** fast path and selecting the **Cultural Environment** icon.
- Codeset depends on the locale (LC_ALL, LANG environment variables).
- Default fonts and font sets depend on the codeset and locale. Using a font that does not match the codeset may produce incorrect output.
- Input Method depends on the locale. The Input Method for the locale should be installed. The Input Method maps Keysyms to a codeset.
- Compose keys (dead keys) depend on the Input Method and X keyboard mapping. An incorrect input method or X keyboard mapping may produce incorrect input.
- Error messages and menu contents depend on the locale and a correct font or fontset. The message catalogs for the locale should be installed. The default messages are English. An incorrect font or fontset can result in garbled menu text and messages.
- Text display depends on the locale and a correct font or fontset. An incorrect font or fontset can result in garbled text. Changing the locale (LC_ALL, LANG environment variables) in an aixterm does not change the codeset that the aixterm displays. If the codeset of the new locale differs from the codeset of **aixterm**, incorrect output (garbled text) may be displayed.
- The X keyboard mapping depends on the system keyboard definition. Xinit sets the X keyboard mapping to match the system keyboard definition. The mapping is changed with **xmodmap**. The X keyboard mapping maps key presses to Keysyms.

Availability of Characters in aixterm

ASCII characters 32 (0x20) to 126 (0x7e) are available in most of the codesets and fonts. Characters (bytes) 0 (0x00) to 31 (0x1f) are treated as control sequences and unprintable characters. Other characters 127 (0x7f) to 255 (0xff) vary with codeset and fonts. Using a font that does not match the codeset the aixterm is started in leads to unpredictable results. For example, box characters (line drawing) are available in **aixterm** vt100 mode with the default vtsingle font. If you use a different font, other characters may be displayed instead. Another example is using a ISO8859-1 font while running in the IBM-850 codeset. Trying to display box characters (line drawing) generates accented characters. Trying to display accented characters generates different accented characters or blanks.

Key Assignments for Bidirectional Languages

In addition to the above key and button bindings, the following key assignments for bidirectional languages are supported by the **aixterm** command:

scr-rev() Reverses the screen orientation and sets the keyboard layer to the default language of the new orientation.

ltr-lang()	Enables the English keyboard layer.
rtl-lang()	Enables the Arabic/Hebrew keyboard layer.
col-mod()	Enables the column heading adjustment which handles each word as a separate column.
auto-push()	Toggles the Autopush function. This function handles mixed left-to-right and right-to-left text. When you enable the Autopush function, reversed segments are automatically initiated and terminated according to the entered character or the selected language layer. Thus, you are relieved of manually invoking the Push function.
chg-push()	Toggles the Push mode. This mode causes the cursor to remain in its position and pushes the typed characters in the direction opposed to the field direction.
shp-in()	Shapes Arabic characters in their initial forms.
shp-is()	Shapes Arabic characters in their isolated forms.
shp-p()	Shapes Arabic characters in their passthru forms.
shp-asd()	Shapes Arabic characters in their automatic forms.
shp-m()	Shapes Arabic characters in their middle forms.
shp-f()	Shapes Arabic characters in their final forms.

The BIDI bindings (for Arabic/Hebrew) are:

```
~Shift ~Ctrl Mod1   <Key>Return:         scr-rev() \n\
~Shift ~Ctrl Mod2   <Key>Return:         scr-rev() \n\
~Shift ~Ctrl Mod1   <Key>Shift_L: ltr-lang() \n\
~Shift ~Ctrl Mod2   <Key>Shift_L: ltr-lang() \n\
~Shift ~Ctrl Mod1   <Key>Shift_R: rtl-lang() \n\
~Shift ~Ctrl Mod2   <Key>Shift_R: rtl-lang() \n\
~Shift ~Ctrl Mod1   <Key>KP_Multiply:    col-mod() \n\
~Shift ~Ctrl Mod2   <Key>KP_Multiply:    col-mod() \n\
~Shift ~Ctrl Mod1   <Key>KP_Divide:      auto-push() \n\
~Shift ~Ctrl Mod2   <Key>KP_Divide:      auto-push() \n\
~Shift ~Ctrl ~Meta  <Key>KP_Divide:      chg-push() \n\
~Shift ~Ctrl Mod1   <Key>KP_1:           shp-in() \n\
~Shift ~Ctrl Mod2   <Key>KP_2:           shp-in() \n\
~Shift ~Ctrl Mod1   <Key>KP_1:           shp-is() \n\
~Shift ~Ctrl Mod1   <Key>KP_2:           shp-is() \n\
~Shift ~Ctrl Mod1   <Key>KP_3:           shp-p() \n\
~Shift ~Ctrl Mod2   <Key>KP_3:           shp-p() \n\
~Shift ~Ctrl Mod1   <Key>KP_4:           shp-asd() \n\
~Shift ~Ctrl Mod2   <Key>KP_4:           shp-asd() \n\
~Shift ~Ctrl Mod1   <Key>KP_7:           shp-m() \n\
~Shift ~Ctrl Mod2   <Key>KP_7:           shp-m() \n\
~Shift ~Ctrl Mod1   <Key>KP_8:           shp-f() \n\
~Shift ~Ctrl Mod2   <Key>KP_8:           shp-f() \n\
```

You can change these values in the **.Xdefaults** file. For example, if you want to use Ctrl+Shift to change language layer, you can add the following line in the **.Xdefaults** file:

```
Translations:   Ctrl<Key>Shift_R: rtl-lang() \n\
                Ctrl<Key>Shift_L: ltr-lang()
```

Flags

A flag takes on the opposite value if the - (minus sign) is changed to a + (plus sign). The following options override those set in the **.Xdefaults** file:

-ah	Highlights the cursor at all times.
-ar	Turns on the autoraise mode of **aixterm**, which automatically raises the window (after a delay determined by the **.Xdefaults** keyword **autoRaiseDelay**) when the mouse cursor enters the window. The default is off. This flag can be turned on and off from the Options menu.
- autopush	Enables the Autopush function for the visual text type.

-b *NumberPixels*
Specifies the width in pixels of an inner border. The inner border is the distance between the outer edge of the characters and the window border. The default is 2.

-bd *Color*
Specifies the color of the highlighted border on color displays. The default is black.

-bg *Color*
Specifies the color of the window background on color displays. The default is white.

-bw *NumberPixels*
Specifies the width of the window border in pixels. The default is 2 pixels. Some window managers can override this option.

-C
Intercepts console messages.

-cc*CharRange***:***Value***,...**
Changes the types of characters that are part of a word. For example, the string `-cc 48-52:3` would make the characters `01234` one word and `56789` a different word. The `:3` defines a word group number 3. By default, numbers are in class 48. The character classes are used by cut and paste.

-cr *Color*
Determines the color of the text cursor on color displays. The default is the foreground color.

-csd *CharShape*
Specifies the default shape of Arabic text. The *CharShape* variable can be one of the following options:

automatic
Shapes the characters automatically.

passthru
Does not shape the characters. The characters are displayed in the same way that they are entered.

isolated
Displays the characters in their isolated form (valid in visual mode only).

initial
Displays the characters in their initial form (valid in visual mode only).

middle
Displays the characters in their middle form (valid in visual mode only).

final
Displays the characters in their final form (valid in visual mode only).

-cu
Causes certain curses applications to display leading tabs correctly. The default is off.

This flag can be turned on and off from the Modes menu.

-display *Name***:***Number*
Identifies the host name and X Server display number where the **aixterm** command is to run. By default, **aixterm** gets the host name and display number from the **DISPLAY** environment variable.

-dw
Causes the mouse cursor to move (warp) automatically to the center of the **aixterm** window when the **aixterm** icon window is deiconified. The default is off.

-e *Command*
Specifies a command to be executed in the window. This flag runs the command; it does not start a shell. If this flag is used, the command and its arguments (if any) must be displayed last on the **aixterm** command line.

When the command exits, the **aixterm** command exits.

-f0 *Font*
Specifies the name of the default font on the command line. Also specifies the name of the font placed in position 0 in the font table. This flag is similar to the **-fn** flag. For example, to specify a default font on the command line, enter the following:

```
aixterm -f0 rom11
```

-f1 *Font*
Specifies the name of the font placed in position 1 in the font table. This flag is similar to the **-fb** flag.

-f2 *Font*
Specifies the name of the font placed in position 2 of the font table. This flag is similar to the **-fi** flag.

-f3 *Font*
Specifies the name of the font placed in position 3 of the font table.

-f4 *Font*
Specifies the name of the font placed in position 4 of the font table.

-f5 *Font*
Specifies the name of the font placed in position 5 of the font table.

-f6 *Font*
Specifies the name of the font placed in position 6 of the font table.

-f7 *Font*
Specifies the name of the font for position 7 in the font table.

—f0 *FontSet*
Specifies the name of the font set for position 0 in the font table. This flag is similar to the **-fn** flag.

—f1 *FontSet*
Specifies the name of the font set for position 1 in the font table. This flag is similar to the **-fb** flag.

—f2 *FontSet*
Specifies the name of the font set for position 2 in the font table. This flag is similar to the **-fi** flag.

—f3 *FontSet*
Specifies the name of the font set for position 3 in the font table.

—f4 *FontSet*
Specifies the name of the font set for position 4 in the font table.

—f5 *FontSet*
Specifies the name of the font set for position 5 in the font table.

—f6 *FontSet*
Specifies the name of the font set for position 6 in the font table.

—f7 *FontSet*
Specifies the name of the font set for position 7 in the font table.

-fb *Font*
Specifies the name of the bold font. This font must be the same height and width as the normal font.

-fi *FontSet*
Specifies the name of the italic font set.

-fg *Color*
Determines the foreground color of the text on color displays. The default is black.

-fn *Font*
Specifies the name of a normal full-text font set. Any fixed-width font set can be used. In HFT emulation, the default is **Rom14.500** for a large display or **Rom10.500** for a small display. In VT102 emulation, the default is **vtsingle**. To specify a font set in the resource file, use **aixterm.Fontset** *FontSet*.

-fs *Font*
Specifies the name of the special graphics font.

-fullcursor
Uses a full block cursor instead of the default underscore cursor.

-geometry *Geometry*
Specifies the location and dimensions of a window. The default is **80x25+0+0**. Some window managers (such as the **mwm** command) can override these defaults.

#geometry*Geometry*
Specifies the location of an icon window. If specified, width and height are ignored. Width and height are taken from the size of the bitmap and the length of the title. The window manager can override the location of the icon.

> **Note:** When you use one of these values as part of an **sh** (shell) command, enclose the value in ″″ (double quotation marks). Normally, #(the pound sign) indicates a comment in a shell script.

-help
Lists the available option flags.

-i
Displays the icon window rather than the normal window when the window is opened. The default is false.

> **Note:** This flag does not work unless the window manager has started.

-ib *File*
Specifies name of the bitmap file to read for use as the icon bitmap file instead of the default bitmap file. You can access a **/usr/include/X11/bitmaps** file from an operating system shell to see a sample bitmap file.

-im *InputMethod*
Specifies a modifier string that identifies the input method to be used by the **aixterm** command.

-j
Causes the **aixterm** command to move multiple lines up at once (jump scroll) if many lines are queued for display. The default is false.

This flag can be turned on and off from the Modes menu.

-keywords Lists the **.Xdefaults** keywords.

-lang *Language* Specifies the language to be used under the **aixterm** command. The language should follow the format for the locale, as used by the **setlocale** function.

-l Causes the **aixterm** command to append output from the window to the end of the **logfile** file. The default is false.

This flag can be turned on and off from the Options menu.

This does not override **LogInhibit** in the **.Xdefaults** file.

-leftscroll Places the scroll bar on the left when it is displayed. The default is on the right side of the text window.

-lf *File* Specifies the file where the output is saved, instead of the default **AixtermLog.***XXXXXX* file, where *XXXXXX* is the process ID of the **aixterm** command. The file is created in the directory where the **aixterm** command is started, or in the home directory for a login **aixterm** command. If the file name begins with a | (pipe symbol), the rest of the string is interpreted as a command to be executed by the shell, and a pipe is opened to the process.

This flag must be used in conjunction with the **-l** flag to work effectively.

-ls Causes the shell run under the **aixterm** command to be a login shell. The user's **.login** or **.profile** file is read, and the initial directory is usually the home directory. The default is false.

-mb Turns on the right margin bell. The default is false.

This flag can be turned on and off from the Modes menu.

-mc *Number* Determines the multiple-click time. This is used by the cut and paste button functions.

-mn Ignores the **XMappingNotify** event. The **-mn** flag is the default.

-ms *Color* Determines the color of the mouse cursor on color displays. The default is the foreground color.

-n *IconName* Specifies the icon name for use by the **aixterm** command.

-name *Application* Specifies the application name to use for the **.Xdefaults** file.

-nb *Number* Specifies the right margin distance at which the margin bell rings. The default is 10 spaces from the right edge of the window.

-nobidi Disables the Arabic/Hebrew functions such as screen reverse, while maintaining an Arabic/Hebrew locale.

- nonulls Enables a Nonulls mode in which nulls within a line are replaced by spaces.

-nss *NumShape* Specifies the default shape of numerals. The *NumShape* variable can be one of the following options:

bilingual
Displays numerals according to the surrounding text. For example, Arabic numerals are displayed within Arabic text and English numerals within English text.

hindi Displays numerals in Hindi.

arabic Displays numerals in Arabic.

passthru
Displays numerals the same way they are entered.

- orient *Orientation* Specifies the default screen orientation. The orientation can be one of the following options:

LTR Left-to-right screen orientation

RTL Right-to-left screen orientation

-outline *Color*
Determines the color of the outline attribute (Keisen) on color displays. The default is the foreground color.

The outline attribute for a character is similar to other character attributes such as bold or reverse video. The outline attribute is displayed as a box drawn to enclose a character or group of characters.

-po *Number*
Specifies the number of lines from the previous screen that display on the screen when the window scrolls one page. The default is 1 line.

-ps
Turns on the page scroll mode.

After a page of lines is displayed, the **aixterm** command stops displaying new lines and the text cursor is no longer displayed. Pressing the Enter key displays one new line. Pressing the Spacebar key or a character key displays a new page. The default is false.

-pt *Preedit*
Specifies the pre-edit type for text composing. The possible pre-edit types are:

over Places the pre-edit window over the spot of character composition.

off Places the pre-edit window off the spot of character composition in the status area.

root Composes character outside of the current window tree.

none Specifies that the input method has no pre-edit area.

-reduced
Causes the **aixterm** command to begin in reduced mode.

-rfb *Font*
Specifies the name of the reduced bold font. This font must be the same width and height as the reduced normal font.

-rfi *Font*
Specifies the name of the reduced italic font. This font must be the same width and height as the reduced normal font.

-rfn *Font*
Specifies the name of the reduced normal font.

-rfs *Font*
Specifies the name of the reduced special graphics font.

-rf0 *Font*
Specifies the name of the reduced font placed in position 0 in the font table. This flag is similar to the **-rfn** flag.

-rf1 *Font*
Specifies the name of the reduced font placed in position 1 in the font table. This flag is similar to the **-rfb** flag.

-rf2 *Font*
Specifies the name of the reduced font placed in position 2 in the font table. This flag is similar to the **-rfi** flag.

-rf3 *Font*
Specifies the name of the reduced font placed in position 3 in the font table.

-rf4 *Font*
Specifies the name of the reduced font placed in position 4 in the font table.

-rf5 *Font*
Specifies the name of the reduced font placed in position 5 in the font table.

-rf6 *Font*
Specifies the name of the reduced font placed in position 6 in the font table.

-rf7 *Font*
Specifies the name of the reduced font placed in position 7 in the font table.

—rf0 *FontSet*
Specifies the name of the reduced fontset placed in position 0 in the font table. This flag is similar to the **-rfn** flag.

—rf1 *FontSet*
Specifies the name of the reduced fontset placed in position 1 in the font table. This flag is similar to the **-rfb** flag.

—rf2 *FontSet*
Specifies the name of the reduced fontset placed in position 2 in the font table. This flag is similar to the **-rfi** flag.

—rf3 *FontSet*
Specifies the name of the reduced fontset placed in position 3 in the font table.

—rf4 *FontSet*
Specifies the name of the reduced fontset placed in position 4 in the font table.

—rf5 *FontSet*
Specifies the name of the reduced fontset placed in position 5 in the font table.

—rf6 *FontSet*
Specifies the name of the reduced fontset placed in position 6 in the font table.

—rf7 *FontSet*
Specifies the name of the reduced fontset placed in position 7 in the font table.

-rv
Reverses the foreground and background colors. This becomes the normal video mode.

This flag can be turned on and off from the Modes menu.

-rw
Turns on the reverse-wraparound mode. The default is false.

This mode allows the cursor to wraparound from the leftmost column to the rightmost column of the previous line. This can be useful in the shell to allow erasing characters backwards across the previous line.

This flag can be turned on and off from the Modes menu.

-s
Turns off synchronous scrolling on the display. The default is true.

When this flag is specified, the **aixterm** command no longer attempts to keep the screen current while scrolling and can run faster when network latencies are very high.

-sb
Causes the scroll bar to display. This flag can be turned on and off from the Modes menu. The default is off.

-sf
Generates the Sun function keycodes for programmed-function (PF) keys in VT102 mode.

-si
Specifies that while using the scroll bar to review previous lines of text, the window is normally repositioned automatically at the bottom of the scroll region before output to the screen is processed. The default is true.

This flag disables window repositioning on output.

-sk
Causes the window to be repositioned automatically in the normal position at the bottom of the scroll region when a key is pressed. The default is false.

This flag is intended for use with the scroll bar to review previous lines of text.

Pressing a key also creates output, which is affected by the **-si** flag.

This flag can be turned on and off from the Scrollbar menu.

-sl *NumberLines*
Specifies the maximum number of lines to save that scroll off of the top of the window. The default is 64.

-sn
Displays the status line to be displayed in normal video (the status line is still enclosed in a box). By default, the status line is displayed in reverse-video relative to the rest of the window. This flag can be turned on and off from the Modes menu.

-st
Displays the status line on startup. The default is false.

-suppress
Specifies that the preediting function in the input method **IMIoctl** call is suppressed.

- symmetric
Enables the Symmetric Swapping mode for handling bidirectional character pairs such as <> and ().

-T *Title*
Sets the title bar name, but not the icon name. If the **-n** option is not specified, or the icon name is not a specified keyword in the **.Xdefaults** file, the title is used as the icon name.

-text *TextType*
Specifies the type of data stream. The *TextType* variable can be one of the following options:

- implicit
Characters are stored in key stroke order.

- visual
Characters are stored the same way that they are displayed. You can use the Autopush mode or Push mode with different shape types.

-ti
Displays the title to the right of the bitmap in the icon window. By default, the title is displayed under the bitmap (if the window manager allows it).

-tm *String*
Specifies a series of terminal setting keywords followed by the characters that should be bound to those functions. Allowable keywords include: **intr**, **quit**, **erase**, **kill**, **eof**, **eol**, **start**, **stop**, **susp**, **dsusp**, **rprnt**, **flush**, **weras**, and **lnext**.

-tn *TerminalName*
Specifies the terminal environment variable. Use the **-tn** flag to change the terminal environment variable only. The terminal environment variable should not be changed to match the terminal in which the X Server is running. The **aixterm** command has no direct access to the terminal where the X Server is running.

-ut
Disables the addition of the login ID to **/etc/utmp**.

-v
Enables VT102 emulation. By default, HFT is emulated.

Note: The keyboard map is needed for this mode.

-vb
Enables the visual bell mode. The visual bell flashes the window on receipt of the Ctrl-G key combination instead of ringing the bell. The default is false.

-W
Causes the mouse cursor to move (warp) to the middle of the **aixterm** window when the window is created. The default is false.

-xrm *String*
Sets the resource string. For example, `aixterm.foreground: blue`

-132
Causes the sm/rm escape sequences to be recognized and the **aixterm** window to be resized as specified. Normally, the sm/rm escape sequences that switch between the 80-column and 132-column modes are ignored. The default is false.

This flag can be turned on and off from the Modes menu.

.Xdefaults Keywords

Use the following keywords to set the defaults for the **aixterm** command.

alwaysHighlight
If true, always highlights the cursor, even when the mouse pointer is outside the window.

autoRaise
If true, raises the **aixterm** window automatically (after a delay of **autoRaiseDelay**) when the mouse cursor enters the window. The default is false. Window managers can override this option.

autoRaiseDelay
If **autoRaise** is true, specifies the number of seconds to delay before automatically raising a window. The default is 2 seconds. Window managers can override this option.

background
Specifies the color of the window background on color displays. The default is a white background.

boldFontSet
Specifies the name of a bold font. This font must have the same height and width as the normal sized font.

borderColor
Specifies the color of the window border. Window managers can override this option.

borderWidth
Specifies the width of the window border in pixels. The default is 2 pixels.

c132	If true, specifies that the sm/rm escape sequences to resize the **aixterm** window between 80 and 132 columns be recognized. The default is false.
charClass	Specifies the character class.
charShape	If set to automatic, the characters are shaped automatically. If set to passthru, the characters do not exert any shaping. If set to isolated, the characters are displayed in isolated shape. If set to initial, the characters are displayed in initial shape. If set to final, the characters are displayed in final shape.
console	If set to true, the **aixterm** command intercepts console messages. The default is false.
curses	If true, causes certain curses applications to display leading tabs correctly. The default is false.
cursorColor	Specifies the color of the text cursor on color displays. The default is the foreground color.
deiconifyWarp	If true, moves or warps the mouse to the center of the window when replacing the **aixterm** icon window with the **aixterm** window. The default is false.
expandTail	The ″seen″, ″sheen″, ″sad″, ″dad″ Arabic characters and their tails are displayed as two characters.
fASD	Enables the automatic shaping function.
fAutoPush	Enables the Autopush function.
fEndPush	Enables the End Push function.
fLTR	Enables the LTR screen orientation.
font0	Specifies the name of the font placed in position 0 in the font table. This flag is similar to the **-fn** flag.
font1	Specifies the name of the font placed in position 1 in the font table. This flag is similar to the **-fb** flag.
font2	Specifies the name of the font placed in position 2 of the font table. This flag is similar to the **-fi** flag.
font3	Specifies the name of the font placed in position 3 of the font table.
font4	Specifies the name of the font placed in position 4 of the font table.
font5	Specifies the name of the font placed in position 5 of the font table.
font6	Specifies the name of the font placed in position 6 of the font table.
font7	Specifies the name of the font for position 7 in the font table.
fontSet	Specifies the name of the normal sized text font used in the body of the **aixterm** window.
fontSet0	Specifies the name of the font set for position 0 in the font table. This flag is similar to the **-fn** flag.
fontSet1	Specifies the name of the font set for position 1 in the font table. This flag is similar to the **-fb** flag.
fontSet2	Specifies the name of the font set for position 2 in the font table. This flag is similar to the **-fi** flag.
fontSet3	Specifies the name of the font set for position 3 in the font table.
fontSet4	Specifies the name of the font set for position 4 in the font table.
fontSet5	Specifies the name of the font set for position 5 in the font table.
fontSet6	Specifies the name of the font set for position 6 in the font table.
fontSet7	Specifies the name of the font set for position 7 in the font table.
foreground	Specifies the color for the text displayed inside the body of the window on color displays. The default is black.
fPush	Enables the Push function.
fRTL	Enables the RTL screen orientation.
fScrev	Enables the Screen Reverse function.
fShapeF	Enables the Final Shape function.
fShapeIN	Enables the Initial Shape function.
fShapeIS	Enables the Isolated Shape function.
fShapeM	Enables the Middle Shape function.
fShapeP	Enables the Passthru shape function.
fullCursor	Displays the full cursor. The default is an underscore cursor.
geometry	Specifies the location or dimensions of the window.

iconBitmap Reads the bitmap file name and uses the resulting bitmap as the icon.

iconGeometry Specifies the location of the icon window.

iconName Specifies the icon name.

iconStartup If true, causes the **aixterm** command to start by displaying an icon window rather than the normal window.

inputMethod Specifies the input method to be used by the **aixterm** command.

internalBorder Specifies the number of pixels between the text characters and the window border. The default is 2 pixels.

italicFontSet Specifies the name of the italic font set.

jumpScroll If true, enables jump scroll. The default is false.

language Specifies the language to be used under the **aixterm** command. The language should follow the format for the locale, as used by the **setlocale** function.

logFile If **logging** is true, specifies the file in which the log is written. The default is **AixtermLog.***XXXXXX*, where *XXXXXX* is a unique ID of the **aixterm** command.

logging If true, appends all input from the pseudo tty to the logfile. The default is false.

logInhibit If true, prevents a user or an application program from enabling logging. This overrides any values set for **logging**.

loginShell If true, indicates that the **aixterm** command should start as a login shell. The default is false.

mappingNotify If set to false, ignores the **XMappingNotify** event. The default is false.

marginBell If true, enables the right margin bell. The default is false.

multiClickTime Specifies the number of milliseconds between button clicks when cutting and pasting. The default is 250 milliseconds.

multiScroll If true, allows asynchronous scrolling.

nMarginBell Specifies the distance from the right edge of the window where the margin bell rings. The default is 10 spaces from the right edge of the window.

noNulls Replaces nulls with spaces within a line.

numShape If set to bilingual, the numbers are shaped according to context. If set to hindi, the numbers are represented in Arabic. If set to arabic, the numbers are represented in English. If set to passthru, the numbers are represented as they are.

orientation If set to LTR, left-to-right is set as the default screen orientation. If set to RTL, right-to-left is set as the default screen orientation.

outline Determines the color of the outline attribute (Keisen) on color displays. The default is the foreground color.

The outline attribute for a character is similar to other character attributes such as bold or reverse video. The outline attribute is displayed as a box drawn to enclose a character or group of characters.

pageOverlap Specifies the number of lines from the previous screen that remain on the screen when the terminal scrolls one page. In page scroll mode, a page is the number of lines in the scrolling region minus the page overlap. The default is 1 line.

pageScroll If true, enables the page scroll mode. The default is false.

After a page of lines displays, **aixterm** stops displaying new lines and the text cursor disappears. Pressing the Enter key displays one new line. Pressing the Spacebar key or a character key displays a new page.

preeditType Specifies the pre-edit type for text composing. The possible pre-edit types are:

over Places the pre-edit window over the spot of character composition.

off Places the pre-edit window off the spot of character composition in the status area.

root Composes character outside of the current window tree.

none Specifies that the input method has no pre-edit area.

pointerColor Specifies the color of the mouse cursor on color displays. The default is the foreground color.

pointerShape Specifies the shape of the mouse cursor to be used in an **aixterm** window. The default is **XC_xterm**. The cursors are listed in the **/usr/include/X11/cursorfont.h** file.

reducedBoldFontSet Specifies the name of the reduced fontset placed in position 1 in the font table.

reducedFont0 Specifies the name of the reduced font placed in position 0 in the font table.

reducedFont1 Specifies the name of the reduced font placed in position 1 in the font table.

reducedFont2 Specifies the name of the reduced font placed in position 2 in the font table.

reducedFont3 Specifies the name of the reduced font placed in position 3 in the font table.

reducedFont4 Specifies the name of the reduced font placed in position 4 in the font table.

reducedFont5 Specifies the name of the reduced font placed in position 5 in the font table.

reducedFont6 Specifies the name of the reduced font placed in position 6 in the font table.

reducedFont7 Specifies the name of the reduced font placed in position 7 in the font table.

reducedFontSet Specifies the name of the reduced fontset placed in position 0 in the font table.

reducedFontSet0 Specifies the name of the reduced fontset placed in position 0 in the font table.

reducedFontSet1 Specifies the name of the reduced fontset placed in position 1 in the font table.

reducedFontSet2 Specifies the name of the reduced fontset placed in position 2 in the font table.

reducedFontSet3 Specifies the name of the reduced fontset placed in position 3 in the font table.

reducedFontSet4 Specifies the name of the reduced fontset placed in position 4 in the font table.

reducedFontSet5 Specifies the name of the reduced fontset placed in position 5 in the font table.

reducedFontSet6 Specifies the name of the reduced fontset placed in position 6 in the font table.

reducedFontSet7 Specifies the name of the reduced fontset placed in position 7 in the font table.

reducedItalicFontSet Specifies the name of the reduced fontset placed in position 2 in the font table.

reducedSpecialFont Specifies the name of the reduced special graphics font.

reducedStartup Causes the **aixterm** command to begin in reduced mode.

reverseVideo If true, reverses the foreground and background color. The default is false.

reverseWrap If true, sets reverse-wraparound mode, which allows the cursor to wrap from the leftmost column to the rightmost column of the previous line. The default is false.

rtArrow The Right Arrow key is handled as a movement key.

saveLines Specifies the maximum number of lines to save when lines scroll off the top of a window. The default is 64 lines.

scrollBar If true, displays the scroll bar during startup.

scrollInput Specifies whether output to the terminal automatically causes the scroll bar to go to the bottom of the scrolling region. The default is true.

scrollKey If true, repositions the window at the bottom of the scroll region (normal position) when a key is pressed while using the scroll bar to review previous lines of text. The default is false.

Pressing a key also creates input, which is affected by the **scrollInput** keyword.

scrollPosition If left, positions the scroll bar to the left side of the screen. The default is right.

signalInhibit If true, specifies that the signals should not be listed. The default is false.

specialFont Specifies the name of the special graphics font.

statusLine If true, displays the status line on startup. The default is false.

statusNormal	If true, displays the status line in normal video (the status line is still enclosed in a box). By default, the status line is in reverse-video relative to the rest of the window.
sunFunctionKeys	If true, the PF keys generate Sun function keycodes when in the VT102 mode. The default is false.
suppress	If true, specifies that the pre-editing function in the input method **IMIoctl** call is suppressed.
symmetric	Enables symmetric character swapping.
termName	Specifies the terminal environment variable, **$TERM**. Use the **termName** keyword to change the terminal environment variable only. The terminal environment variable should not be changed to match the terminal in which the X Server is running. The **aixterm** command has no direct access to the terminal where the X Server is running.
textType	If set to implicit, the data stream type is set to implicit. If set to visual, the data stream type is set to visual.
textUnderIcon	If False, displays the title of the icon window at the right of the bitmap in the icon window. By default, the title is displayed under the bitmap.
title	Specifies the title to show in the title bar. The default is **aixterm**.
ttyModes	Specifies the tty settings.
translations	Specifies the key and button translations to be supplied.
utmpInhibit	If False, adds the login ID to the **/etc/utmp** file. The default is false**.**
visualBell	If true, enables the visual bell mode which flashes the window on receipt of a Ctrl-G key sequence. The default is false.
vt102	If true, enables VT102 mode. The default is emulation.
warp	If true, automatically warps (moves) the mouse cursor to the center of a newly created **aixterm** window. The default is false.

Example

The following example can be used to create an **aixterm**, specifying the size and location of the window, using a font other than the default, and also specifying the foreground color that is used in text. The **aixterm** command then runs a command in that window.

```
aixterm -geometry 20x10+0+175 -fn Bld14.500 -fg DarkTurquoise -e
/tmp/banner_cmd &
```

The **aixterm** command is NOT an X Toolkit based application. Because of this, the **aixterm** command gets resource files as follows:

- **System defaults** from the first of these it finds:

  ```
  $XFILESEARCHPATH %T=app-defaults %N=Xdefaults %L=$LANG
  $XFILESEARCHPATH %T=app-defaults %N=Xdefaults %L=
  /usr/lpp/X11/defaults/$LANG/Xdefaults
  /usr/lpp/X11/defaults/Xdefaults
  /usr/lib/X11/$LANG/app-defaults/Xdefaults
  /usr/lib/X11/app-defaults/Xdefaults
  /usr/lpp/X11/defaults/app-defaults/Xdefaults
  ```

- **Application system defaults** from the first of these it finds:

  ```
  $XFILESEARCHPATH %T=app-defaults %N=Aixterm %L=$LANG
  $XFILESEARCHPATH %T=app-defaults %N=Aixterm %L=
  $XFILESEARCHPATH %T=app-defaults %N=aixterm %L=$LANG
  $XFILESEARCHPATH %T=app-defaults %N=aixterm %L=
  /usr/lpp/X11/defaults/$LANG/Aixterm
  /usr/lpp/X11/defaults/Aixterm
  /usr/lib/X11/$LANG/app-defaults/Aixterm
  /usr/lib/X11/app-defaults/Aixterm
  /usr/lib/X11/defaults/app-defaults/Aixterm
  /usr/lpp/X11/defaults/$LANG/aixterm
  ```

```
/usr/lpp/X11/defaults/aixterm
/usr/lib/X11/$LANG/app-defaults/aixterm
/usr/lib/X11/app-defaults/aixterm
/usr/lib/X11/defaults/app-defaults/aixterm
```

- **User application defaults** from the first of these it finds:

```
$XUSERFILESEARCHPATH %T=app-defaults %N=Aixterm %L=$LANG
$XUSERFILESEARCHPATH %T=app-defaults %N=Aixterm %L=
$XUSERFILESEARCHPATH %T=app-defaults %N=aixterm %L=$LANG
$XUSERFILESEARCHPATH %T=app-defaults %N=aixterm %L=
$XAPPLRESDIR/$LANG/Aixterm
$XAPPLRESDIR/Aixterm
$XAPPLRESDIR/$LANG/aixterm
$XAPPLRESDIR/aixterm
$HOME/$LANG/Aixterm
$HOME/Aixterm
$HOME/$LANG/aixterm
```

- **User defaults** from the first of these it finds:

```
dpy->xdefaults          (A.K.A. "RESOURCE_MANAGER" property)
$HOME/$LANG/.Xdefaults
$HOME/.Xdefaults
```

- **Host defaults** from the first of these it finds:

```
$XENVIRONMENT
$HOME/$LANG/.Xdefaults-hostname
$HOME/.Xdefaults-hostname
```

> **Note:** XFILESEARCHPATH and XUSERFILESEARCHPATH support is limited to the %T, %N and %L substitution strings. Also, $LANG is actually whatever the result of the setlocale(LC_CTYPE,NULL) call is.

Related Information

telnet, tn, or tn3270 command.

Bidirectionality and Character Shaping in *AIX 5L Version 5.3 National Language Support Guide and Reference*.

---

ali Command

Purpose

Lists mail aliases and their addresses.

Syntax

ali [**-alias** *File*] [**-list** | **-nolist**] [**-normalize** | **-nonormalize**] [**-user** *User* | **-nouser**] [*Alias* ...]

Description

The **ali** command lists mail aliases and their addresses. By default, this command searches the **/etc/mh/MailAliases** file and writes to standard output each alias and its address defined in the file. To specify an alternate mail aliases file, use the **-alias** *File* flag.

If you specify the **-user** flag, the **ali** command searches the alias files for the user name and writes to standard output the aliases that contain this user name.

Flags

-alias *File* Specifies the mail alias file to be searched. The default is the **/etc/mh/MailAliases** 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.
-list	Displays each address on a separate line.
-nolist	Displays addresses on as few lines as possible. This flag is the default.
-nonormalize	Prevents conversion of local host nicknames to official host names. This is the default.
-normalize	Converts local host nicknames to their official host names.
-nouser	Lists the address for an alias. This flag is the default.
-user *User*	Lists the aliases that contain the specified user. When the **-user** and **-nonormalize** flags are used together, the result may be a partial list of aliases that contain the specified user.

Examples

1. To display a list of all aliases and their addresses in the **/etc/mh/MailAliases** file, enter:
   ```
   ali
   ```
2. To list the names and addresses of the `mygroup` alias, enter:
   ```
   ali   mygroup
   ```

   A list similar to the following is displayed on your local system:
   ```
   mike@mercury   george@helium   vicky@venus
   ```

Files

$HOME/.mh_profile	Contains the MH user profile.
/etc/group	Contains a list of groups.
/etc/passwd	Contains a list of users.
/etc/mh/MailAliases	Contains the default mail alias file.
/usr/bin/ali	Contains the **ali** command.

Related Information

The **comp** command, **dist** command, **forw** command, **repl** command, **send** command, **whom** command.

Mail applications in *Networks and communication management*.

---

alias Command

Purpose

Defines or displays aliases.

Syntax

alias [**-t**] [**-x**] [*AliasName* [=*String*]] ...

Description

The **alias** command creates or redefines alias definitions or writes existing alias definitions to standard output.

If no flags or parameters are supplied, all existing alias definitions are written to standard output. You can display a specific alias definition by using the *AliasName* parameter.

Create a new alias by using the *AliasName*=*String* parameter pair. When the shell encounters an alias on the command line or in a shell script, it substitutes the definition supplied by the string. The *String* variable

can contain any valid shell text. Enclose the value of the *String* variable in single quotes if the string contains spaces. If the *AliasName* parameter is not a valid name, the **alias** command displays an error message.

If you specify the **-t** flag, the shell displays aliases that are *tracked*. A tracked command uses the full path name of the command. A tracked command can become undefined when the value of the **PATH** environment variable is reset, but aliases created with the **-t** flag remain tracked.

If you specify the **-x** flag, the shell displays aliases that are *exported*. An exported alias is active in all shells.

An alias definition affects the current shell environment and the execution environments of any subshells. The alias definition affects neither the parent process of the current shell nor any utility environment invoked by the shell.

Flags

-t Sets or displays all existing tracked aliases. If this flag is used with the *AliasName* parameter, the new alias is tracked and the alias definition includes the full path name obtained by doing a path search. When the value of the **PATH** environment variable is reset, the alias definition becomes undefined but remains tracked.

-x Displays all existing exported alias definitions. If this flag is used with the *AliasName* parameter, the new alias is exported. Exported alias are not defined across separate invocations of the shell. You must put alias definitions in your environment file to have aliases defined for separate shell invocations.

Exit Status

The following exit values are returned:

0 Successful completion.

>0 One of the specified alias name did not have an alias definition, or an error occurred.

Examples

1. To change the **ls** command so that it displays information in columns and annotates the output, enter:
   ```
   alias ls='ls -CF'
   ```
2. To create a command for repeating previous entries in the command history file, enter:
   ```
   alias r='fc -s'
   ```
3. To use 1KB units for the **du** command, enter:
   ```
   alias du=du\ -k
   ```
4. To create a command to display all active processes for user `Dee`, enter:
   ```
   alias psc='ps -ef | grep Dee'
   ```
5. To see the full path name of the **ls** command, enter:
   ```
   alias -t ls
   ```

   The screen displays `ls=/usr/bin/ls`.

Files

/usr/bin/ksh	Contains the Korn shell **alias** built-in command.
/usr/bin/alias	Contains the **alias** command.

Related Information
The **ksh** command.

---

alog Command

Purpose
Creates and maintains fixed-size log files created from standard input.

Syntax

To Show the Contents of a Log File
alog -f *LogFile* [**-o**]

To Log Data to a Specified Log File
alog -f *LogFile* | [[**-q**] [**-s** *Size*]]

To Display the Verbosity Value of a Specified Log Type
alog -t *LogType* **-V**

To Change the Attributes of a Specified Log Type
alog -C -t *LogType* [**-f** *LogFile*] [**-s** *Size*] [**-w** *Verbosity*]

To Display the Current Attributes of a Specified Log Type
alog -L [**-t** *LogType*]

To Display the Usage of the alog Command
alog -H

Description
The **alog** command reads standard input, writes to standard output, and copies the output into a fixed-size file. This file is treated as a circular log. If the file is full, new entries are written over the oldest existing entries.

The **alog** command works with log files that are specified on the command line or with logs that are defined in the alog configuration database. Logs that are defined in the alog configuration database are identified by *LogType*. The **File**, **Size**, and **Verbosity** attributes for each defined *LogType* are stored in the alog configuration database with the *LogType*. You can add a new *LogType* to the alog configuration database using the **odmadd** command. You can change the attributes of *LogType* defined in the alog configuration database using the **alog** command.

Flags

-C Changes the attributes for a specified *LogType*. Use the **-C** flag with the **-f**, **-s**, and **-w** flags to change the **File**, **Size**, and **Verbosity** attributes for the specified *LogType*. The **-t** *LogType* flag is required.

Note: Using the **-C** flag with **-s***Size* only changes the size value in ODM and does not change the size of the actual log file.

If the **-C** flag is used, the **alog** command does not copy standard input to standard output or to a log file.

When the **-C** flag is used to modify the attributes for the console log type, the console log file is also modified and the console device driver is updated to use the new values. This is a deviation from the normal operation of **alog -C** and is done to accommodate special formatting in the console log file.

Note: You must have root user authority to change **alog** attributes.

-f *LogFile* Specifies the name of a log file. If the specified log file does not exist, one is created. If the **alog** command is unable to write to a log file, it writes to **/dev/null**. Use the **-f** *LogFile* flag with the **-C** and **-t** flags to change the **File** attribute for a *LogType* defined in the alog configuration database.

-H Displays the usage of the **alog** command.

-L Lists the log types currently defined in the alog configuration database. If you use the **-L** flag with the **-t** *LogType* flag, the attributes for a specified *LogType* are listed. The current values of the **File**, **Size**, and **Verbosity** attributes are listed as colon separated values:

```
<File>:<Size>:<Verbosity>
```

If the **-L** flag is used, the **alog** command does not copy standard input to standard output or to **File**.

-o Lists the contents of the log file. Writes the contents of the log file to standard output in sequential order.

-q Copies standard input to a log file but does not write to standard output.

-s *Size* Specifies the size limit of the log file in bytes. The space for the log file is reserved when it is created. If you create a new log file and do not specify the **Size** attribute, the minimum size, 4096 bytes, is used. If the log file already exists, its size will be changed. The size you specify is rounded upward to the next integral multiple of 4096 bytes. The maximum size for a log file is 2 GB. If the specified size is greater than 2 GB, only 2 GB is considered. If you decrease the size of the log file, the oldest entries in the log are deleted if they do not fit within the new size limit. You must have write permission for the log file to change its size.

Use the **-s** *Size* flag with the **-C** and the **-t** flags to change the **Size** attribute for *LogType* defined in the alog configuration database. Only the size value in ODM is changed. The size of the actual log file remains the same. The new **Size** attribute value is used the next time a log file is created.

-t *LogType* Identifies a log defined in the alog configuration database. The **alog** command gets the log's file name and size from the alog configuration database. If *LogFile* does not exist, one is created.

If the **alog** command cannot get the information for the specified *LogType* from the alog configuration database or if the **alog** command is unable to write to *LogFile*, it writes to **/dev/null**.

If you specify *LogType* and *LogFile* using the **-f** flag, *LogFile* is used and *LogType* is ignored.

-V Writes the current value of the **Verbosity** attribute for *LogType* that is defined in the alog configuration database to standard output. If you do not specify *LogType*, or the *LogType* you specify is not defined, nothing is written to standard output.

The value output using the **alog** command with the **-t** *LogType* and the **-V** flags can be used by a command that is piping its output to the **alog** command to control the verbosity of the data it writes to the pipe.

-w *Verbosity*

Changes the **Verbosity** attribute for *LogType* defined in the alog configuration database when used with the **-C** and the **-t** flags.

The **Verbosity** attribute can have a value from 0 to 9. If the value is 0, no information is copied to *LogFile* by the **alog** command. All of the information is still written to standard output. If the value is not 0, all of the information piped to the **alog** command's standard input is copied to *LogFile* and to standard output.

Examples

1. To record the current date and time in a log file named `sample.log`, enter:

   ```
   date | alog -f /tmp/sample.log
   ```

2. To list the contents of `/tmp/sample.log` log file, enter:

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

3. To change the size of the log file named `/tmp/sample.log` to 8192 bytes, enter:

   ```
   echo "resizing log file" | alog -f /tmp/sample.log -s 8192
   ```

4. To add a new log type `sample` to the alog configuration database, create the `alog.add` file in the following format:

   ```
   SWservAt:
       attribute="alog_type"
       deflt="sample"
       value="sample"

   SWservAt:
       attribute="sample_logname"
       deflt="/tmp/sample.log"
       value="/tmp/sample.log"

   SWservAt:
       attribute="sample_logsize"
       deflt="4096"
       value="4096"

   SWservAt:
       attribute="sample_logverb"
       deflt="1"
       value="1"
   ```

   After creating the `alog.add` file, enter:

   ```
   odmadd alog.add
   ```

   This adds the `alog.add` file to the `SWservAt` database.

5. To change the name of the log file for the log type `sample` to `/var/sample.log` in the alog configuration database, enter:

   ```
   alog -C -t sample -f /var/sample.log
   ```

6. To change the size of the boot log to 8192 bytes and reflect the new size in ODM, enter:

   ```
   alog -C -t boot -s 8192
   echo "Changed log size" | alog -t boot -s 8192
   ```

Files

/etc/objrepos/SWservAt	Software Service Aids Attributes Object Class

Related Information

The **odmadd** command.

How to Add Objects to an Object Class in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

alstat Command

Purpose

Shows alignment exception statistics.

Syntax

alstat [**-e** | **-v**] [*Interval*] [*Count*]

Description

The **alstat** command displays alignment exception statistics. Alignment exceptions may occur when the processor cannot perform a memory access due to an unsupported memory alignment offset (such as a floating point double load from an address that is not a multiple of 8). However, some types of unaligned memory references may be corrected by some processors and does not generate an alignment exception.

The alignment exception count since the last time the machine was rebooted and the count in the current interval are displayed. You can optionally display emulation exception statistics or individual processor alignment statistics.

The default output displays statistics every second. The sampling *Interval* and *Count* of iterations can be also specified.

Parameters

Interval	Interval between samples.
Count	Number of iterations.

Flags

-e	Displays emulation exception statistics. This flag cannot be used with the **-v** flag.
-v	Display individual processor statistics. This flag cannot be used with the **-e** flag.

Examples

1. To display alignment exception statistics every second, type:

   ```
   alstat
   ```

   This produces the following output:

   ```
     Alignment  Alignment
     SinceBoot      Delta
       8845591          0
       8845591          0
       8845591          0
       8845591          0
       8845591          0
       8845591          0
     ...
   ```

2. To display emulation and alignment exception statistics every two seconds, a total of 5 times, type:

   ```
   alstat -e 2 5
   ```

   This produces the following output:

   ```
     Emulation  Emulation  Alignment  Alignment
     SinceBoot      Delta  SinceBoot      Delta
      21260604          0   70091846          0
   ```

```
 23423104    2162500   72193861    2102015
 25609796    2186692   74292759    2098898
 27772897    2163101   76392234    2099475
 29958509    2185612   78490284    2098050
```

3. To display alignment exception statistics, every 5 seconds, for each processor, type:

   ```
   alstat -v 5
   ```

   This produces the following output:

```
  Alignment  Alignment  Alignment  Alignment
  SinceBoot      Delta    Delta00    Delta01
   88406295          0          0          0
   93697825    5291530          0    5291530
   98930330    5232505    5232505          0
  102595591    3665261     232697    3432564
  102595591          0          0          0
```

Related Information

The **emstat** command.

---

alt_disk_copy Command

Purpose

Clones (makes a copy of) the currently running system to an alternate disk.

Syntax

To copy rootvg to Alternate Disk:

alt_disk_copy -d *targetdisks...* [-i **image.data**] [-s *script* [-b *bundlename*] [-l *installpflags*] [-l *imageslocation*] [-f *fixbundle*] [-F *fixes*] [-e *excludelist*] [-w *filesets*] [-n] [-P *phaseoption*] [-c *console*] [-x *first boot script*] [-R *resolveconf*] [-DBOVgrTS] [**-u**]

Description

The **alt_disk_copy** command allows users to copy the current rootvg to an alternate disk and to update the operating system to the next maintenance or technology level, without taking the machine down for an extended period of time and mitigating outage risk. This can be done by creating a copy of the current rootvg on an alternate disk and simultaneously applying software updates. If needed, the **bootlist** command can be run after the new disk has been booted, and the bootlist can be changed to boot back to the older maintenance or technology level of the operating system.

Cloning the running rootvg, allows the user to create a backup copy of the root volume group. This copy can be used as a back up in case the rootvg failed, or it can be modified by installing additional updates. One scenario might be to clone a 5300-00 system, and then install updates to bring the cloned rootvg to 5300-01. This would update the system while it was still running. Rebooting from the new rootvg would bring the level of the running system to 5300-01. If there was a problem with this level, changing the bootlist back to the 5300-00 disk and rebooting would bring the system back to 5300-00. Other scenarios would include cloning the rootvg and applying individual fixes, rebooting the system and testing those fixes, and rebooting back to the original rootvg if there was a problem.

At the end of the install, a volume group, altinst_rootvg, is left on the target disks in the varied off state as a place holder. If varied on, it indicates that it owns no logical volumes; however, the volume group does contain logical volumes, but they have been removed from the ODM because their names now conflict with the names of the logical volumes on the running system. Do not vary on the altinst_rootvg volume group; instead, leave the definition there as a placeholder.

After rebooting from the new alternate disk, the former rootvg volume group shows up in a lspv listing as `old_rootvg`, and it includes all disks in the original rootvg. This former rootvg volume group is set to not vary-on at reboot, and it should only be removed with the **alt_rootvg_op -X old_rootvg** or **alt_disk_install -X old_rootvg** commands.

If a return to the original rootvg is necessary, the **bootlist** command is used to change the bootlist to reboot from the original rootvg.

Notes:

1. Alternate disk operations create volume groups, logical volumes, special device files, and file systems using the **alt** prefix. If **alt_disk_copy** is utilized on a system, the administrator should avoid having or creating volume groups, logical volumes, special device files, or file systems with the **alt**, prefix—alternate disk operations might inadvertently remove, alter, or damage these items.
2. NIM alternate disk migration (upgrading version or release levels) is supported with the **nimadm** command in AIX 5.1 and later. Please see the **nimadm** documentation for more details.
3. The current LVM limit for logical volume names is 15 characters. Because the alternate disk installation commands prepend the 4-character **alt**_ prefix, the limit for the original logical volume names in the rootvg to be copied or installed is 11 characters. If an original logical volume name exceeds 11 characters, it can be shortened by using a customized **image.data** (see the **-i** flag).
4. When cloning the rootvg volume group, a new boot image is created with the **bosboot** command. If the **/dev/ipldevice** is removed or altered then the **bosboot** command will fail.
5. Do not use direct LVM commands (such as **exportvg**, **importvg**, **varyoffvg**, or **chlv**) on alternate rootvg volume groups.
6. This function is also available with the Network Installation Management (NIM). See the NIM Guide for more information.
7. The **alt_disk_copy** command only backs up mounted file systems. Mount all file systems that you want to back up. The **mksysb** command backs up mounted journaled file systems (JFS) and enhanced journaled file systems (JFS2) in the rootvg. For more information about backing up file systems, see the **mount** command.

Flags

-b *bundlename*	Path name of optional file with a list of packages or filesets that are installed after a rootvg clone. The **-l** flag must be used with this option.
-B	Would specify not running bootlist after the **mksysb** or clone. If set, then the **-r** flag cannot be used.
-c *Console*	The device name to be used as the alternate rootvg's system console. This option is only valid with the **-O** flag.
-d *targetdisks*	Specifies a space-delimited list of the name or names of the target disks where the alternate rootvg will be created. These disks must not currently contain any volume group definition. The **lspv** command should show these disks as belonging to volume group None.
-D	Turns on debug (sets **-x** output).

-e *excludelist*
Optional *exclude.list* to use when cloning rootvg. The rules for exclusion follow the pattern-matching rules of the grep command. The *excludelist* must be a full path name.
Note: If you want to exclude certain files from the backup, create the **/etc/exclude.rootvg** file with an ASCII editor and enter the patterns of file names that you do not want included in your system backup image. The patterns in this file are input to the pattern-matching conventions of the **grep** command to determine which files will be excluded from the backup. If you want to exclude files listed in the **/etc/exclude.rootvg** file, select the **Exclude Files** field and press the Tab key once to change the default value to yes. For example, to exclude all the contents of the **scratch** directory, edit the exclude file to read as follows:

```
/scratch/
```

For example, to exclude the contents of the **/tmp** directory, and avoid excluding any other directories that have **/tmp** in the path name, edit the exclude file to read as follows:

```
^./tmp/
```

All files are backed up relative to **.** (current working directory). To exclude any file or directory for which it is important to have the search match the string at the beginning of the line, use the caret character (^) as the first character in the search string, followed by the dot character (.), followed by the filename or directory to be excluded. If the filename or directory being excluded is a substring of another filename or directory, use the caret character followed by the dot character (^.) to indicate that the search should start at the beginning of the line, and use the dollar sign character ($) to indicate that the search should end at the end of the line.

-f *fixbundle*
Optional file with a list of APARs to install after a clone of rootvg. The **-l** flag must be used with this option.

-F *fixes*
Optional list of APARs (for example, `IX123456`) to install after a clone of rootvg. The **-l** flag must be used with this option.

-g
Skips disk bootability checks.

-i *image.data*
Optional **image.data** file to use instead of the default **image.data** file created from **rootvg**. The **image.data** file name must be a full path name (such as **/tmp/my_image.data**).

-l *installpflags*
The flags to use when updating or installing new filesets into the cloned **alt_inst_rootvg**. The default flag is **-acgX**. The **-l** flag must be used with this option.

-l *imageslocation*
Location of **installp** images or updates to apply after a clone of rootvg. This can be a directory full path name or device name (such as **/dev/rmt0**).

-n
Remain NIM client. The **/.rhosts** and **/etc/niminfo** files are copied to the file system of the alternate rootvg.

-O
Performs a device reset on the target **altinst_rootvg**. This causes the alternate disk install to not retain any user-defined device configurations. This flag is useful if the target disk or disks become the rootvg of a different system (such as in the case of logical partitioning or system disk swap).

-P *phases*
The phase or phases to execute during this invocation of **alt_disk_copy**. Valid values are: 1, 2, 3, 12, 23, or all (default).

12 Performs phases 1 and 2.

23 Performs phases 2 and 3.

all Performs all three phases.

-r
Specifies to reboot from the alternate disk when the **alt_disk_copy** command finishes.

-R *resolvconf*
The **resolv.conf** file to replace the existing one after the rootvg has been cloned. You must specify a full path name.

-s *script*
Optional customization script to run at the end of the **mksysb** install or the rootvg clone. This file must be executable. This script is called on the running system before the **/alt_inst** file systems are unmounted, so files can be copied from the running system to the **/alt_inst** file systems before the reboot.

-S	Indicates that you want to skip space-checking on target disks before you start performing the cloning or installation operations. **Important:** JFS2 file systems contain more metadata than JFS file systems. When you use the -S flag in conjunction with the -T flag, it skips space-checking. In this situation, it does not verify that there is enough space in the newly created JFS2 file system to store the contents of the file system plus the additional metadata.
-T	Indicates that you want to convert JFS file systems to JFS2 file systems during the process of recreating the rootvg volume group on target disks.
-u	Copies file systems that belong to a workload partition (WPAR) in the defined state in the alternate system. **Note:** To be included in the alternate disk, all file systems that belong to a WPAR in the defined state need to be in the rootvg volume group.
-V	Turn on verbose output. This shows the files that are being backed up for rootvg clones.
-w *filesets*	List of filesets to install after cloning a rootvg. The **-l** flag must be used with this option.
-x *script*	Optional customization script to run during the initial boot of the alternate rootvg, after all file systems are mounted.

Exit Status

0	All **alt_disk_copy** related operations completed successfully.
>0	An error occurred.

Examples

1. To clone the running 5300-00 rootvg to **hdisk3**, then apply updates from **/updates** to bring the cloned rootvg to a 5300-01 level:

   ```
   alt_disk_copy -d hdisk3 -F 5300-01_AIX_ML -l /updates
   ```

   The bootlist would then be set to boot from **hdisk3** at the next reboot.
2. To clone the running rootvg to **hdisk3** and **hdisk4**, and execute **update_all** on all updates from **/updates**:

   ```
   alt_disk_copy -d "hdisk3 hdisk4" -b update_all -l /updates
   ```

   The bootlist would then be set to boot from **hdisk3** at the next reboot.
3. To clone the running rootvg to **hdisk1** and stop after phase 1:

   ```
   alt_disk_copy -d hdisk1 -P1
   ```
4. To execute phases 2 and 3 on an existing alternate rootvg and reboot the system on successful completion:

   ```
   alt_disk_copy -d hdisk1 -P23 -r
   ```
5. To clone the running system to **hdisk1** and **hdisk2**, and to convert the file systems from JFS file systems to JFS2 file systems, run the following command:

   ```
   alt_disk_copy -B -T -d hdisk1 hdisk2
   ```

   Attention: Do not change the bootlist to use the cloned **rootvg**.

Location

/usr/sbin/alt_disk_copy

Files

/usr/sbin/alt_disk_copy	Contains the **alt_disk_copy** command.

Related Information

“alt_disk_mksysb Command” on page 108, “alt_rootvg_op Command” on page 112, “bootlist Command” on page 261, “bosboot Command” on page 269,

The **lspv** in *AIX 5L Version 5.3 Commands Reference, Volume 3*.

The **nim**, **nimadm** command in *AIX 5L Version 5.3 Commands Reference, Volume 4*.

---

alt_disk_install Command

Purpose

Installs an alternate disk with a **mksysb** install image or clones the currently running system to an alternate disk. This command is obsolete in AIX 5.3.

Note: In AIX 5.3, the **alt_disk_install** command is replaced by the **alt_disk_copy**, **alt_disk_mksysb**, and **alt_rootvg_op** commands. The **alt_disk_install** module continues to be shipped as a wrapper to the new commands, but the **alt_disk_install** command does not support any new functions, flags, or features.

Syntax

″ **Create Alternate Disk:** ″

alt_disk_install { **-d** *device* | **-C**} [**-i** *image.data*] [**-s** *script*] [**-R** *resolv_conf*] [**-D**] [**-B**] [**-V**] [**-r**] [**-O**]

[**-p** *platform*] [**-L** *mksysb_level*]

[**-b** *bundle_name*] [**-l** *installp_flags*]

[**-l** *images_location*] [**-f** *fix_bundle*]

[**-F** *fixes*] [**-e** *exclude_list*] [**-w** *filesets*]

[**-n**] [**-P** *phase_option*] *target_disks...*

″**Clean Up Alternate Disk Volume Group:**″

alt_disk_install -X

For alt_disk_install AIX 4.3.2 or later:

″ **Determine Volume Group Boot Disk:**″

alt_disk_install -q *disk*

″**Put-to-sleep Volume Group:**″

alt_disk_install -S

″**Rename Alternate Disk Volume Group:**″

alt_disk_install -v *new_volume_group_name disk*

″**Wake-up Volume Group:**″

alt_disk_install -W *disk*

″**Clean Up Alternate Disk Volume Group:**″

alt_disk_install -X [*volume_group*]

Description

Note: In AIX 5.3 the **alt_disk_install** command has been broken up into three commands: **alt_disk_copy**, **alt_disk_mksysb**, and **alt_rootvg_op**. No new functionality will be added to this command.

The **alt_disk_install** command allows users a way to update the operating system to the next release, maintenance level, or technology level, without taking the machine down for an extended period of time. This can be done in two ways, by installing a mksysb image on a separate disk, or by cloning the current system and then applying updates to get to the next maintenance or technology level.

Attention: alt_disk_install creates volume groups, logical volumes, special device files, and file systems using the ″alt″ prefix. If **alt_disk_install** is utilized on a system, the administrator should avoid having or creating volume groups, logical volumes, special device files, or file systems with the ″alt″ prefix - **alt_disk_install** operations may inadvertently remove, alter, or damage these items.

The first function, installing a mksysb, requires an AIX 4.3 or later mksysb image, an AIX 4.3 or later mksysb tape, or an AIX 4.3.3 or later mksysb CD. The **alt_disk_install** command is called with a disk or disks that are not currently in use, and the mksysb is restored to those disks such that, if the user chooses, the next reboot boots the system on an AIX 4.3 or later system.

Notes:

1. You cannot use **alt_disk_install** to install an earlier version of AIX than the one currently installed on the system. For example, you cannot install an AIX 4.3 mksysb on an AIX 5.1 system.
2. If needed, the **bootlist** command can be run after the new disk has been booted, and the bootlist can be changed to boot back to the older version of the operating system.

The second function, cloning the running rootvg, allows the user to create a backup copy of the root volume group. This copy could be used as a back up in case the rootvg failed, or it could be modified by installing additional updates. One scenario might be to clone a 4.2.0 system, then install updates to bring the cloned rootvg to 4.2.1.0. This would update the system while it was still running, then rebooting from the new rootvg would bring the level of the running system to 4.2.1. If there was a problem with this level, changing the bootlist back to the 4.2.0 disk and rebooting would bring the system back to 4.2.0. Other scenarios would include cloning the rootvg and applying individual fixes, rebooting the system and testing those fixes, and rebooting back to the original rootvg if there was a problem.

Note: NIM alternate disk migration (upgrading version or release levels) is supported with the **nimadm** command in AIX 5.1 and later. Please see the **nimadm** documentation for more details.

Currently, you can run the **alt_disk_install** command on 4.1.4.0 and higher systems for both of these functions. The **bos.alt_disk_install.rte** fileset must be installed on the system to execute the **alt_disk_install** command, and the **bos.alt_disk_install.boot_images** fileset must also be installed to perform a mksysb install to an alternate disk.

The mksysb image that is used must be created ahead of time and have all the necessary device and kernel support required for the system that it’s going to be installed on. No new device or kernel support can be installed before the system is rebooted from the newly installed disk.

Note: The version release maintenance or technology level of mksysb that you are installing must match the level of the **bos.alt_disk_install.boot_images** fileset.

When cloning the rootvg volume group, a new boot image is created with the **bosboot** command. When installing a mksysb image, a boot image for the level of mksysb and platform type is copied to the boot logical volume for the new alternate rootvg. When the system is rebooted, the **bosboot** command is run in the early stage of boot, and the system is rebooted once again. This is to synchronize the boot image with the mksysb that was just restored. The system then boots in normal mode.

At the end of the install, a volume group, altinst_rootvg, is left on the target disks in the varied off state as a place holder. If varied on, it shows as owning no logical volumes, but it does indeed contain logical volumes, but they have been removed from the ODM because their names now conflict with the names of the logical volumes on the running system. It is recommended that you not vary on the altinst_rootvg volume group, but just leave the definition there as a place holder.

After the system reboots from the new alternate disk, the former rootvg volume group does not show up in a **lspv** listing, unless the **alt_disk_install** version is 4.3.2 or higher.

For **alt_disk_install** 4.3.2 or greater:

> After rebooting from the new alternate disk, the former rootvg volume group shows up in a **lspv** listing as ″old_rootvg″, and includes all disk(s) in the original rootvg. This former rootvg volume group is set to NOT varyon at reboot, and should ONLY be removed with the -X flag (i.e. **alt_disk_install** -X old_rootvg).

If a return to the original rootvg is necessary, the bootlist command is used to change the bootlist to reboot from the original rootvg.

For **alt_disk_install** 4.3.2 or greater:

> If it is unclear which disk is the boot disk for a specific volume group, the -q flag can be used to determine the boot disk. This can be useful when a volume group is comprised of multiple disks and a change in the bootlist is necessary.

The alternate root file system is mounted as **/alt_inst**, so other file systems would have that prefix (**/alt_inst/usr**, **/alt_inst/var**). This is how they should be accessed if using a customization script.

> **Attention:** If you have created an alternate rootvg with **alt_disk_install**, but no longer wish to use it, or want to run **alt_disk_install** commands, do not run exportvg on altinst_rootvg.
>
> Simply run the **alt_disk_install -X** command to remove the altinst_rootvg definition from the ODM database. The reason you cannot run the **exportvg** command (or the **reducevg** command) is that the logical volume names and file systems now have the real names, and **exportvg** removes the stanza's for the real file system from **/etc/filesystems** for the real rootvg.
>
> If **exportvg** is run by accident, be sure to recreate the **/etc/filesystems** file before rebooting the system. The system will not reboot without a correct **/etc/filesystems** file.

This function is also available with the Network Installation Management (NIM). See the NIM Guide for more information.

The AIX 4.3.1 and greater version of **alt_disk_install** can be executed in phases. The install is divided into three phases, and the default is to perform all three phases.

Phase 1 Creates the **altinst_rootvg** volume group, the **alt**_ ″logical volumes″, the **/alt_inst** file systems, and restores the mksysb or rootvg data.

Phase 2 Runs any specified customization script, installs updates, new filesets, fixes or bundles (cloning only), copies a **resolv.conf** file if specified, and copies files over to remain a NIM client if specified.

Phase 3 Unmounts the **/alt_inst** file systems, renames the file systems and logical volumes, removes the **alt**_ **logical** volumes, names ODM and varies off the altinst_rootvg. It sets the bootlist and reboots if specified.

You can run each phase separately, run Phases 1 and 2 together, or run Phases 2 and 3 together. Phase 2 can be run multiple times before Phase 3 is run.

You must run Phase 3 to get a volume group that is a usable rootvg. Running Phase 1 and 2 leave the /alt_inst file systems mounted.

If you have run Phase 1 and or Phase 2, and want to start over (remove the altinst_rootvg), run the **alt_disk_install -X** command to clean up.

For **alt_disk_install** 4.3.2 or greater:

If data access is necessary between the original rootvg and the new alternate disk, a volume group ″wake-up″ can be accomplished, using the -W flag, on the non-booted volume group. The ″wake-up″ puts the volume group in a post **alt_disk_install** phase 1 state (i.e. the /alt_inst file systems will be mounted).

Note: The volume group that experiences the ″wake-up″ will be renamed ″altinst_rootvg″.

Limitation

The running system's version of operating system must be greater than or equal to the operating system version of the volume group that undergoes the ″wake-up″. This may mean that it's necessary to boot from the ″altinst_rootvg″ and ″wake-up″ the ″old_rootvg″.

For example: An alternate disk is created from an **alt_disk_install** 4.3.3 mksysb, on a 4.1.5 running system. To access data between the two volume groups, it is necessary to boot from the 4.3.3 alternate disk and ″wake-up″ the 4.1.5 ″old_rootvg″ volume group.

This limitation is caused by a jfs log entry incompatibility. It is possible to ″wake-up″ a volume group that contains a greater operating system version, but the volume group could not have ever been the system rootvg. If so, the volume group would have made jfs log entries that could not be interpreted by an older operating system version rootvg, when the volume group was experiencing a ″wake-up″. JFS log entries are usually present for file systems that were not unmounted before a reboot, for example, /,/usr.

The **alt_disk_install** command will not allow a ″wake-up″ to occur on a volume group with a greater operating system version, unless the FORCE environment variable is set to ″yes″.

Attention: If a FORCE ″wake-up″ is attempted on a volume group that contains a greater operating system version then the running operating system, **AND** the ″waking″ volume group has been a system rootvg, errors will occur.

When data access is no longer needed, the volume group can be put to sleep, using the -S flag.

Note: The volume group that has experienced a ″wake-up″ MUST be ″put-to-sleep″ before it can be booted and used as the rootvg.

Flags

-B Would specify not running bootlist after the mksysb or clone. If set, the **-r** flag cannot be used.
Note: The **-B** and **-X** flags are mutually exclusive.

-C Clone rootvg.

Note: -d and **-C** are mutually exclusive.

-d *device*
The value for *device* can be:

`tape device - for example, /dev/rmt0`

OR

`path name of mksysb image in a file system.`

Note: -d and **-C** are mutually exclusive.

-D
Turns on debug (set **-x** output).

-i *image.data*
Optional image.data file to use instead of default image.data from mksysb image or image.data created from rootvg. The image.data file name must be a full pathname, for example, `/tmp/my_image.data`.

For **alt_disk_install** 4.3.2 or greater:

> If certain logical volumes need to be placed on a specific target disk, this should be annotated in the logical volume LV_SOURCE_DISK_LIST field of the user specified image.data file.

-p *platform*
This is a platform to use to create the name of the disk boot image, which may be supplied by a vendor that wanted to support this function. This flag is only valid for mksysb installs (**-d** flag).

-P*phase*
The *phase* to execute during this invocation of **alt_disk_install**. Valid values are: 1, 2, 3, 12, 23, or all.

- 12 - performs phases 1 and 2.
- 23 - performs phases 2 and 3.
- all - performs all three phases

-r
Would specify to reboot from the new disk when the **alt_disk_install** command is complete.

-R *resolv_conf*
The **resolv.conf** file to replace the existing one after the mksysb has been restored or the rootvg has been cloned. You must use a full pathname for *resolv_conf*.

-s *script*
Optional customization script to run at the end of the mksysb install or the rootvg clone. This file must be executable. This script is called on the running system before the /alt_inst file systems are unmounted, so files can be copied from the running system to the **/alt_inst** file systems before the reboot. This is the only opportunity to copy or modify files in the alternate file system because the logical volume names will be changed to match rootvg's, and they will not be accessible until the system is rebooted with the new alternate rootvg, or a ″wake-up″ is performed on the altinst_rootvg. You must use a full pathname for *script*.

-V
Turn on verbose output. This shows the files that are being backed up for rootvg clones. This flag shows files that are restored for mksysb alt_disk_installs.

-L *mksysb_level*
This level will be combined with the platform type to create the boot image name to use (for example, rspc_4.3.0_boot in AIX 5.1 and earlier). This must be in the form V.R.M. The default will be AIX 4.3. The mksysb image will be checked against this level to verify that they are the same.

-n
Remain NIM client. The **/.rhosts** and **/etc/niminfo** files are copied to the alternate rootvg's file system.

-X
Removes the altinst_rootvg volume group definition from the ODM database. This returns the **lspv** listing for the volume group to ″None″. This will not remove actual data from the volume group. Therefore, you can still reboot from that volume group, if you reset your bootlist.

> For **alt_disk_install** 4.3.2 or greater, the flag allows for specified volume group name ODM database definition removal, for example, `-X old_rootvg`.

Notes:

1. The **-B** and **-X** flags are mutually exclusive.
2. If you specify the **-X** flag, all other flags are ignored.

-O Performs a device reset on the target altinst_rootvg. This will cause **alt_disk_install** to NOT retain any user defined device configurations. This flag is useful if the target disk or disks will become the rootvg of a different system (such as in the case of logical partitioning or system disk swap).

The following flags are only valid for use when cloning the rootvg (**-C**).

-b *bundle_name* Pathname of optional file with a list of packages or filesets that will be installed after a rootvg clone. The **-l** flag must be used with this option.

-e *exclude_list* Optional exclude.list to use when cloning rootvg. The rules for exclusion follow the pattern matching rules of the **grep** command. The *exclude_list* must be a full pathname.

> **Note:** If you want to exclude certain files from the backup, create the **/etc/exclude.rootvg** file, with an ASCII editor, and enter the patterns of file names that you do not want included in your system backup image. The patterns in this file are input to the pattern matching conventions of the **grep** command to determine which files will be excluded from the backup. If you want to exclude files listed in the **/etc/exclude.rootvg** file, select the Exclude Files field and press the Tab key once to change the default value to yes.
>
> For example, to exclude all the contents of the directory called scratch, edit the exclude file to read as follows:
>
> ```
> /scratch/
> ```
>
> For example, to exclude the contents of the directory called **/tmp**, and avoid excluding any other directories that have **/tmp** in the pathname, edit the exclude file to read as follows:
>
> ```
> ^./tmp/
> ```
>
> All files are backed up relative to . (current working directory). To exclude any file or directory for which the it is important to have the search match the string at the beginning of the line, use ^ (caret character) as the first character in the search string, followed by **.** (dot character), followed by the filename or directory to be excluded.
>
> If the filename or directory being excluded is a substring of another filename or directory, use **^.** (caret character followed by dot character) to indicate that the search should begin at the beginning of the line and/or use **$** (dollar sign character) to indicate that the search should end at the end of the line.

-f *fix_bundle* Optional file with a list of APARs to install after a clone of rootvg. The **-l** flag must be used with this option.

-F *fixes* Optional list of APARs (for example, ″IX123456″) to install after a clone of rootvg. The **-l** flag must be used with this option.

-l *installp_flags* The flags to use when updating or installing new filesets into the cloned alt_inst_rootvg. Default flags: ″-acgX″ The **-l** flag must be used with this option.

-l *images_location* Location of installp images or updates to apply after a clone of rootvg. This can be a directory full pathname or device name (like **/dev/rmt0**).

-w *filesets* List of filesets to install after cloning a rootvg. The **-l** flag must be used with this option.

The following flags are available for **alt_disk_install** version 4.3.2 or greater:

-q *disk*
: Used to return the volume group boot disk name. This is especially useful when trying to determine the boot disk from several disks in the ″old_rootvg″ volume group, after rebooting from the alternate disk.

-S
: Will ″put-to-sleep″ the volume group. This is used after a volume group ″wake-up″. (**-W**).

-v *new_volume_group_name disk*
: Used to rename the alternate disk volume group. This is especially useful when creating multiple alternate disks, on multiple volume groups, and name identification is necessary.

-W *disk*
: Used to ″wake-up″ a volume group for data access between the rootvg and the alternate disk rootvg.

  Note: The volume group that experiences the ″wake-up″ will be renamed ″altinst_rootvg″.

Limitation

The running system's version of the operating system must be greater than or equal to the operating system version of the volume group that undergoes the ″wake-up″. This may mean that it's necessary to boot from the ″altinst_rootvg″ and ″wake-up″ the ″old_rootvg″.

Parameters

target_disks
: Specifies the name or names of the target disks where the alternate rootvg will be created. This disk or these disks must not currently contain any volume group definition. The **lspv** command should show these disks as belonging to volume group **None**.

Examples

1. To clone the running 4.2.0 rootvg to hdisk3, then apply updates from /updates to bring the cloned rootvg to a 4.2.1 level:

   ```
   alt_disk_install -C -F 4.2.1.0_AIX_ML -l /updates hdisk3
   ```

   The bootlist would then be set to boot from hdisk3 at the next reboot.
2. To install a 4.3 mksysb image on hdisk3, then run a customized script (/home/myscript) to copy some user files over to the alternate rootvg file systems before reboot:

   ```
   alt_disk_install -d /mksysb_images/4.3_mksysb -s /home/myscript hdisk3
   ```
3. To remove the original rootvg ODM database entry, after booting from the new alternate disk:

   ```
   alt_disk_install -X old_rootvg
   ```

   The **lspv** listing for the original rootvg will be changed to ″None″. Therefore, a new volume group could be created on those disks.
4. To determine the boot disk for a volume group with multiple physical volume:

   ```
   alt_disk_install -q hdisk0
   ```

   Illustrated Example

   ```
   # lspv
   hdisk0          00006091aef8b687     old_rootvg
   hdisk1          00076443210a72ea     rootvg
   hdisk2          0000875f48998649     old_rootvg
   # alt_disk_install -q hdisk0
   hdisk2
   ```

   In this case, the boot disk for ″old_rootvg″ is actually hdisk2. Therefore, you could reset your bootlist to hdisk2 and reboot to the original rootvg volume group.

5. To modify an **alt_disk_install** volume group name:

   ```
   alt_disk_install -v alt_disk_432 hdisk2
   ```

   Illustrated Example

   ```
   # lspv
   hdisk0          00006091aef8b687     rootvg
   hdisk1          00000103000d1a78     rootvg
   hdisk2          000040445043d9f3     altinst_rootvg
   hdisk3          00076443210a72ea     altinst_rootvg
   hdisk4          0000875f48998649     None
   hdisk5          000005317c58000e     None
   # alt_disk_install -v alt_disk_432   hdisk2
   #lspv
   hdisk0          00006091aef8b687     rootvg
   hdisk1          00000103000d1a78     rootvg
   hdisk2          000040445043d9f3     alt_disk_432
   hdisk3          00076443210a72ea     alt_disk_432
   hdisk4          0000875f48998649     None
   hdisk5          000005317c58000e     None
   ```

6. To ″wake_up″ an original rootvg, after booting from the new alternate disk:

   ```
   alt_disk_install -W hdisk0
   ```

   Illustrated Example

   ```
   # lspv
   hdisk0          000040445043d9f3     old_rootvg
   hdisk1          00076443210a72ea     rootvg
   # alt_disk_install -W hdisk0
   # lspv
   hdisk0          000040445043d9f3     altinst_rootvg
   hdisk1          00076443210a72ea     rootvg
   ```

   At this point, the ″altinst_rootvg″ volume group is varied-on and the /alt_inst file systems will be mounted.

7. To ″put-to-sleep″ a volume group that had experienced a ″wake-up″:

   ```
   alt_disk_install -S
   ```

   Illustrated Example

   ```
   # lspv
   hdisk0          000040445043d9f3     altinst_rootvg
   hdisk1          00076443210a72ea     rootvg
   # alt_disk_install -S
   # lspv
   hdisk0          000040445043d9f3     altinst_rootvg
   hdisk1          00076443210a72ea     rootvg
   ```

   The ″altinst_rootvg″ is no longer varied-on and the /alt_inst file systems are no longer mounted. If it's necessary for the ″altinst_rootvg″ volume group name to be changed back to ″old_rootvg″, this can be done with the ″-v″ flag.

Files

/usr/sbin/alt_disk_install	Contains the alt_disk_install command

Related Information

The **bootlist** command, **bosboot** command, **lspv** command, **nim** command, **nimadm** command.

---

alt_disk_mksysb Command

Purpose

Installs an alternate disk with a **mksysb** install base install image.

Syntax

alt_disk_mksysb -m *device* -d *target_disks...* [-i **image.data**] [-s *script*] [-R *resolv_conf*] [-p *platform*] [-L *mksysb_level*] [-n] [-P *phase_option*] [-c *console*] [-K] [-DBOVgkrxyzTS]

Description

The **alt_disk_mksysb** command allows the users to install a **mksysb** system backup to a separate disk without taking the machine down for an extended period, thus mitigating outage risk. Using the **alt_disk_mksysb** command is the only method available to restore a backup containing **multibos** Base Operating System (BOS) instances.

Installing a **mksysb** requires an AIX 4.3 or later **mksysb** image, **mksysb** tape, or **mksysb** CD. The **alt_disk_mksysb** command is called with a disk or set of disks that is not currently in use, and the **mksysb** is restored to those disks such that, if the user chooses, the next reboot boots the system on an AIX 4.3 or later system.

The **bos.alt_disk_install.rte** and **bos.alt_disk_install.boot_images** filesets must be installed on the system to run the **alt_disk_mksysb** command.

The **mksysb** image that is used must have all the necessary device and kernel support required for the system it is installed on. You cannot install a new device or kernel support before the system is rebooted from the newly installed disk.

The alternate root file system is mounted as **/alt_inst** to ensure that the other file systems have a prefix, such as **/alt_inst/usr**, **/alt_inst/var**). This is the method in which the files must be accessed using a customization script.

At the end of the install, a volume group, **altinst_rootvg**, is left on the target disks in the varied-off state as a place holder. If varied on, it indicates that it owns no logical volumes; however, it does contain logical volumes, but they have been removed from the ODM because their names now conflict with the names of the logical volumes on the running system. Do not vary on the **altinst_rootvg** volume group; instead, leave the **altinst_rootvg** volume group as a placeholder.

After the system reboots from the new alternate disk, the former rootvg volume group shows up in the lspv listing as `old_rootvg`. Do not vary on the **old_rootvg** volume group; instead, leave the **old_rootvg** volume group as a placeholder.

If a return to the original **rootvg** is necessary, the **bootlist** command is used to change the bootlist to reboot from the original **rootvg**.

Notes:

1. Alternate disk operations create volume groups, logical volumes, special device files, and file systems using the **alt** prefix. If **alt_disk_copy** is used on a system, the administrator must avoid having or creating volume groups, logical volumes, special device files, or file systems with the **alt**, prefix—alternate disk operations might inadvertently remove, alter, or damage these items.
2. **alt_disk_mksysb** needs to use preexisting boot images during **mksysb** installation. **alt_disk_mksysb** first looks for the boot images in the alternate **rootvg** (that is, the contents of the **mksysb**); if boot images are not found, **alt_disk_mksysb** searches for them in the current **rootvg**.
   - The alternate disk install boot image location for **altinst_rootvg** is: **/alt_inst/usr/lpp/bos.alt_disk_install/boot_images**
   - The alternate disk install boot image location for the current **rootvg** is: **/usr/lpp/bos.alt_disk_install/boot_images**
   - The generic versions of the alternate install boot images are provided by the **bos.alt_disk_install.boot_images** fileset.

3. The version release maintenance or technology level of **mksysb** that you are installing must match the level of the **bos.alt_disk_install.boot_images** fileset. For example, if the `oslevel` on the source system (the system where the **mksysb** was created) returns `5.3.0.0`, the **bos.alt_disk_install.boot_images** fileset must be at 5.3.0.*X*, where *X* is the highest available fix level.
4. If **alt_disk_mksysb** needs to use the generic boot images shipped with the **bos.alt_disk_install.boot_images** fileset, the system performs an additional reboot when booting from the alternate **rootvg** for the first time.
5. You cannot use the **alt_disk_mksysb** command to install an earlier version of AIX than the one currently installed on the system. For example, you cannot install an AIX 4.3 **mksysb** on an AIX 5.1 system.
6. The current LVM limit for logical volume names is 15 characters.As the alternate disk installation commands contain the 4-character **alt**_ prefix, the limit for the original logical volume names in the **rootvg** to be copied or installed is 11 characters. If an original logical volume name exceeds 11 characters, it can be shortened using a customized **image.data** (see the **-i** flag).
7. Do not use direct LVM commands (such as **exportvg**, **importvg**, **varyoffvg**, and **chlv**) on alternate **rootvg** volume groups.
8. This function is also available with the Network Installation Management (NIM). See the NIM Guide for more information.
9. NIM alternate disk migration (upgrading version or release levels) is supported with the **nimadm** command in AIX 5.1 and later. See the **nimadm** command for more details.

Flags

-B Specifies not running bootlist after the operation. If set, then the **-r** flag cannot be used.

-c *console* Specifies the device name to be used as the alternate **rootvg**'s system console. This option is only valid with the **-O** flag.

-D Turns on debug (sets **-x** output).

-d *target_disks* Specifies a space-delimited list of the name or names of the target disks where the alternate **rootvg** is created. This disk or these disks must not currently contain any volume group definition. The **lspv** command must indicate that these disks belong to volume group **None**.

-g Specifies that bootable checks for the **target_disks** are overlooked.

-K Specifies that the 64 - bit kernel must be used, if possible.

-k Specifies that **mksysb** devices be kept (formally the **ALT_KEEP_MDEV** variable).

-i *image_data* Optional **image.data** file to use instead of the default **image.data** file from **mksysb** image. The **image.date** file name must be a full path name (for example, **/tmp/my_image.data**).

-L *mksysb_level* This level is combined with the platform type to create the boot image name (for example, **rspc_5.1.0_boot** in AIX 5.1 and earlier). This must be in the form V.R.M. The **mksysb** image is checked against this level to verify that they are the same.

-m *device* The value for device can be:

- Tape device (for example, **/dev/rmt0**)
- Path name of **mksysb** image in a file system

-n Remain NIM client. The **/.rhosts** and **/etc/niminfo** files are copied to the alternate **rootvg**'s file system.

-P *Phases* The phase or phases to execute during this invocation of the **alt_disk_mksysb** command. Valid values are: 1, 2, 3, 12, 23, or all.

12 Performs phases 1 and 2.

23 Performs phases 2 and 3.

all Performs all three phases.

-p *platform* Platform used to create the name of the disk boot image, which might be supplied by a vendor that wanted to support this function.

-O	Performs a device reset on the target **altinst_rootvg**. This causes **alt_disk_install** to not retain any user-defined device configurations. This flag is useful if the target disk or disks become the **rootvg** of a different system (such as in the case of logical partitioning or system disk swap).
-R *resolv_conf*	The **resolv.conf** file that replaces the existing one after the **mksysb** has been restored. You must use a full path name for *resolv_conf*.
-r	Specifies to reboot from the new disk when the **alt_disk_mksysb** command is complete.
-s *script*	Optional customization script to run at the end of the **mksysb** install. This file must be executable. This script is called on the running system before the **/alt_inst** file systems are unmounted, so files can be copied from the running system to the **/alt_inst** file systems before the reboot. This is the only opportunity to copy or modify files in the alternate file system because the logical volume names will be changed to match **rootvg**'s, and they will not be accessible until the system is rebooted with the new alternate **rootvg**, or a ″wake-up″ is performed on the **altinst_rootvg** using the **alt_rootvg_op** command. You must use a full path name for the script.
-S	Indicates that you want to skip space-checking on target disks before you start performing the cloning or installation operations. **Important:** JFS2 file systems contain more metadata than JFS file systems. When you use the -S flag with the -T flag, it skips space-checking. In this situation, it does not verify that there is enough space in the newly created JFS2 file system to store the contents of the file system plus the additional metadata.
-T	Indicates that you want to convert JFS file systems to JFS2 file systems during the process of recreating the **rootvg** volume group on target disks.
-V	Turn on verbose output. This shows the files that are restored during the **alt_disk_mksysb** operation.
-y	Looks for and imports (if found) **mksysb** volume groups. This flag causes **alt_disk_install** to import the data VGs known to the **mksysb** and to not import the local data VGs known at install time (the default). The imports are performed with the following script: **/usr/lpp/bos.alt_disk_install/bin/alt_import_oldvgs**.
-z	Does not import any type of non-**rootvg** volume groups. This flag overrides the **-y** flag.

Exit Status

0	All **alt_disk_mksysb** related operations completed successfully.
>0	An error occurred.

Examples

1. To install a **mksysb** image on **hdisk3** and **hdisk4** , then run a customized script (**/tmp/script**) to copy some user files over to the alternate **rootvg** file systems before reboot:

   ```
   alt_disk_mksysb -m /mksysb_images/my_mksysb -d "hdisk3 hdisk4" -s /tmp/script
   ```

2. To install a **mksysb** image on **hdisk2** and stop after phase 1:

   ```
   alt_disk_mksysb -m /mksysb_images/my_mksysb -d hdisk2 -P1
   ```

3. To execute phases 2 and 3 on an existing alternate **rootvg** on **hdisk4** and reboot the system upon successful completion:

   ```
   alt_disk_mksysb -d hdisk4 -m /mksysb_images/my_mksysb -P23 -r
   ```

4. To install a **mksysb** image on **hdisk1**, and to convert the file system from a JFS file system to a JFS2 file system, run the following command:

   ```
   alt_disk_mksysb -B -T -m  /mksysb_images/my_mksysb -d hdisk1
   ```

   Attention: Do not change the bootlist to use the cloned **rootvg**.

Location

/usr/sbin/alt_disk_mksysb

Files

/usr/sbin/alt_disk_mksysb Contains the **alt_disk_mksysb** command.

Related Information

"alt_disk_copy Command" on page 97, "alt_rootvg_op Command," "bootlist Command" on page 261.

The **lspv** in *AIX 5L Version 5.3 Commands Reference, Volume 3*.

The **nim**, **nimadm** command in *AIX 5L Version 5.3 Commands Reference, Volume 4*.

---

alt_rootvg_op Command

Purpose

Performs operations on existing alternate **rootvg** volume groups.

Syntax

To determine Volume Group Boot Disk (-q):

alt_rootvg_op -q -d *disk* [-D]

To rename Alternate Disk Volume Group (-v):

alt_rootvg_op -v *new volume group name* -d *disk* [-D]

To wake up Volume Group (-W):

alt_rootvg_op -W -d *disk* [-D]

To put to sleep Volume Group (-S):

alt_rootvg_op -S [-tD]

To clean up Alternate Disk Volume Group (-X):

alt_rootvg_op -X [*volume group*] [-D]

To customize Alternate Disk Volume Group (-C):

alt_rootvg_op -C [-R *resolv_conf*] [-s *script*] [-b *bundle_name*] [-I *installp_flags*] [-I *images_location*] [-f *fix_bundle*] [-F *fixes*] [-w *filesets*] [-DV]

Description

The **alt_rootvg_op** command can be used to determine which disk is the boot disk for a specific volume group. Use the **-q** flag to determine the boot disk. This can be useful when a volume group is comprised of multiple disks and a change in the bootlist is necessary.

This command can also be used to rename the alternate disk volume groups. This is especially useful when creating multiple alternate disks, on multiple volume groups, and name identification is necessary.

If data access is necessary between the current **rootvg** and an alternate disk, use the **alt_rootvg_op** command to perform a volume group ″wake-up″ (using the **-W** flag) on the nonbooted volume group. The

″wake-up″ puts the volume group in a post phase 1 state (that is, the **/alt_inst** file systems will be mounted). The customize operation (**-C** flag) can be executed at this time.

The running system's operating system must be a version greater than or equal to the operating system version of the volume group that undergoes the ″wake-up.″ This might mean that it is necessary to boot from the **altinst_rootvg** and ″wake up″ the **old_rootvg**.

The **alt_rootvg_op** command does not allow a ″wake-up″ to occur on a volume group with a greater operating system version, unless the **FORCE** environment variable is set to Yes.

Notes:

1. The volume group that experiences the ″wake-up″ is renamed **altinst_rootvg**.
2. Do not execute phase 3 on the volume group that experiences the ″wake-up.″
3. Do not reboot the system if there is a volume group in the ″wake″ state. This can cause damage or data loss to the volume group that is in the ″wake″ state. Volume groups in the ″wake″ state can be put to ″sleep″ with the **-S** flag.

When data access is no longer needed, the **alt_rootvg_op** command can be used to put to sleep the volume group in the ″wake″ state, using the **-S** flag. The boot image on the target alternate **rootvg** can be rebuilt if necessary with the **-t** flag. The sleep operations revert the alternate volume group to an inactive state.

When cleaning up the alternate disk volume group, the **alt_rootvg_op** command uses the -X flag to remove the **altinst_rootvg** volume group definition from the ODM database. If the target volume group is varied off at the time this operation is executed, only the ODM definitions associated with the target volume group are removed. The actual volume group data is not removed. If the volume group is bootable, you can still reboot from that volume group, by setting the bootlist to a boot disk in this volume group. The **-X** flag accepts a volume group name as an argument and acts on the **altinst_rootvg** volume group by default.

The customize operation of the **alt_rootvg_op** command (using the **-C** flag) can be used to perform the following functions on an active alternate root volume group:

- Install software and software updates. Apply this operation only to alternate volume groups created with the **rootvg** copy operation.
- Execute customization script.
- Copy **resolve.conf** files.

Flags

-b *bundle_name*	Path name of optional file with a list of packages or filesets that will be installed after a **rootvg** clone. The **-I** flag must be used with this option.
-C	Performs the customization operation on the active **rootvg** volume group.
-d *target_disk*	Specifies a space-delimited list of the name or names of the target disks that will be targets of the given operation.
-D	Turns on debug (sets **-x** output).
-f *fix_bundle*	Optional file with a list of APARs to install after a clone of **rootvg**. The **-I** flag must be used with this option.
-F *fixes*	Optional list of APARs (for example, IY123456) to install after a clone of **rootvg**. The **-I** flag must be used with this option.
-I *installp_flags*	The flags to use when updating or installing new filesets into the cloned **alt_inst_rootvg**. The default flag is **-acgX**. The **-I** flag must be used with this option.
-l *images_location*	Location of **installp** images or updates to apply after a clone of **rootvg**. This can be a directory full path name or device name (like **/dev/rmt0**).
-q	Determines the volume group boot disk.

-R *resolv_conf*	The **resolv.conf** file to replace the existing one in the **rootvg**. You must specify a full path name.
-s *script*	Optional customization script to be executed during the customization phase. This file must be executable. This script is called on the running system before the **/alt_inst** file systems are unmounted, so files can be copied from the running system to the **/alt_inst** file systems before the reboot.
-S	Puts to sleep the alternate root volume group that experienced the previous ″wake″ operation.
-t	Rebuilds the alternate boot image before putting the volume group to ″sleep.″ This flag is only valid for alternate root volume groups created with the clone or copy install operation. The **-t** flag requires the **-S** flag.
-v *Name*	Renames an alternate disk volume group to the name specified with the *Name* parameter.
-V	Turn on verbose output.
-w *filesets*	List of filesets to install after cloning a **rootvg**. The **-l** flag must be used with this option.
-W	Performs a wake-up on the root volume group located on the **target_disk**.
-X	Removes the **altinst_rootvg** volume group definition from the ODM database.

Exit Status

0	All **alt_rootvg_op** related operations completed successfully.
>0	An error occurred.

Examples

1. To remove the original **rootvg** ODM database entry, after booting from the new alternate disk, enter the following command:

   ```
   alt_rootvg_op -X old_rootvg
   ```

2. To cleanup the current alternate disk install operation, enter the following command:

   ```
   alt_rootvg_op -X
   ```

3. To determine the boot disk for a volume group with multiple physical volume, enter the following command:

   ```
   alt_rootvg_op -q -d hdisk0
   ```

   Illustrated Example

   ```
   # lspv

   hdisk0          00006091aef8b687     old_rootvg
   hdisk1          00076443210a72ea     rootvg
   hdisk2          0000875f48998649     old_rootvg

   # alt_rootvg_op -q -d hdisk0

   hdisk2
   ```

4. To modify an **alt_disk_install** volume group name, enter the following command:

   ```
   alt_rootvg_op -v alt_disk_530 -d hdisk2
   ```

   Illustrated Example

   ```
   # lspv

   hdisk0          00006091aef8b687     rootvg
   hdisk1          00000103000d1a78     rootvg
   hdisk2          000040445043d9f3     altinst_rootvg
   hdisk3          00076443210a72ea     altinst_rootvg
   hdisk4          0000875f48998649     None
   hdisk5          000005317c58000e     None
   ```

```
# alt_rootvg_op -v alt_disk_432 -d hdisk2

#lspv

hdisk0          00006091aef8b687     rootvg
hdisk1          00000103000d1a78     rootvg
hdisk2          000040445043d9f3     alt_disk_432
hdisk3          00076443210a72ea     alt_disk_432
hdisk4          0000875f48998649     None
hdisk5          000005317c58000e     None
```

5. To ″wake up″ an original **rootvg** after booting from the new alternate disk, enter the following command:

   ```
   alt_rootvg_op -W -d hdisk0
   ```

6. To ″put to sleep″ a volume group that had experienced a ″wake-up″ and rebuild the boot image, enter the following command:

   ```
   alt_rootvg_op -S -t
   ```

7. To update the active alternate **rootvg** to the latest fileset levels available in **/updates** and install them into the alternate **root** volume group, enter the following command:

   ```
   alt_rootvg_op -C -b update_all -l /updates
   ```

Location

/usr/sbin/alt_rootvg_op

Files

/usr/sbin/alt_rootvg_op	Contains the **alt_rootvg_op** command.

Related Information

“alt_disk_copy Command” on page 97, “alt_disk_mksysb Command” on page 108, “bootlist Command” on page 261, “bosboot Command” on page 269.

The **lspv** in *AIX 5L Version 5.3 Commands Reference, Volume 3*.

The **nim**, **nimadm** command in *AIX 5L Version 5.3 Commands Reference, Volume 4*.

---

amepat Command

Purpose

Active Memory Expansion Planning and Advisory Tool **amepat** reports Active Memory Expansion information and statistics as well as provides advisory report that assists in planning the use of Active Memory Expansion for an existing workload.

Syntax

amepat [{{[**-c** *max_ame_cpuusage%*] | [**-C** *max_ame_cpuusage*]}|[**-e** *startexpfactor [:stopexpfactor [:incexpfactor]*]]}][{[**-t** *tgt_expmem_size*]|[**-a**]}]

[**-n** *num_entries*] [**-m** *min_mem_gain*] [**-u** *minucomp_poolsize*]

[**-v**] [**-N**] [{ [**-P** *recfile*] | [**Duration**] | [**Interval <Samples>**]}]

amepat [**-N**] [**-R** *recfile*] {[**Duration**] | [**Interval <Samples>**]}

Description

Active Memory Expansion Planning and Advisory Tool **amepat** serves two key functions:

1. **Workload Planning -** The **amepat** can be run to determine a workload that would benefit from Active Memory Expansion, and also to provide a list of possible Active Memory Expansion configurations for a workload.
2. **Monitoring -** When Active Memory Expansion is enabled, the **amepat** tool can be used to monitor the workload and Active Memory Expansion performance statistics.

The **amepat** can be invoked in two different modes:

1. In the **Recording** mode **amepat** records systems configuration and various performance statistics into a user specified recording file.
2. In the **Reporting** mode **amepat** analyzes the system configuration and performance statistics, collected in real time or from the user specified recording file, to generate workload utilization and planning reports.

Note: This tool is available from AIX 6.1 with the 6100-04 Technology Level-SP2 release, or later.

Workload Planning

When considering using Active Memory Expansion for an existing workload, **amepat** can be used to provide guidance on possible Active Memory Expansion configurations for the workload. When **amepat** is run concurrently with an existing workload that is not using Active Memory Expansion, **amepat** will monitor the memory usage, memory reference patterns, and data compressibility over a user-configurable time period of the workload. The tool then generate a report with a list of possible Active Memory Expansion configurations for the workload. The tool will include an estimate of the CPU utilization impacts for the different Active Memory Expansion configurations.

amepat can be run on all versions of IBM Power Systems™ supported by AIX 6.1™. It includes systems using POWER4™, POWER5™, POWER6™, or POWER7™ family processors.

There are two key considerations when running **amepat** to do workload planning: the time at which to run the tool and the duration to run the tool. To get the best possible results from the tool, the tool must be run during the period of peak utilization of the workload. It ensures that the tool captures peak of utilization and memory usage information of the workload.

In order to use **amepat** to generate a report for workload planning, a monitoring duration must be specified when starting **amepat**.

In addition to using **amepat** on workload that are not yet using Active Memory Expansion, **amepat** can also be run in LPAR's where Active Memory Expansion is already enabled. When used in this mode, **amepat** it provides a report of other possible Active Memory Expansion configurations for the workload.

Note: **amepat** requires privileged access to do Workload Planning. When a user invokes the tool without the required privilege then the Workload Planning Capability is disabled (**-N** flag will be turned on implicitly)

Monitoring

amepat can also be used to only monitor CPU and memory utilization statistics (Disabling the workload planning capability). With this Monitoring capability, **amepat** just gathers CPU and memory utilization statistics, does not gather the additional data required for generating the report for workload planning. Thus, Active Memory Expansion Modeling and Advisory reports are not generated.

When **amepat** is invoked without a duration or interval, **amepat** defaults to monitoring only capability, and **amepat** will report a snapshot of the LPAR's memory, CPU utilization.

amepat can be invoked with duration and run with Monitoring only capability using the **-N** flag. The **-N** flag disables the workload planning capability of this tool, thus disabling the data gathering & reporting for workload planning.

Note: Both Recording and Reporting modes can be invoked with **-N** flag. The **-N** flag is supported both in Active Memory Expansion Enabled and Disabled Machines.

amepat Report

Following are the six different sections of report displayed by the **amepat** tool:

Command Information Section: The Command Information Section provides details about the arguments passed to the **amepat** tool, time of invocation, the total time the system is monitored and the number of samples collected.

System Configuration Section: The System Configuration Section provides details about the system configuration. The following table provides the complete list of information reported.

Partition Name	Node name from where **amepat** is invoked
Processor Implementation Mode	The processor implementation mode, it can be,POWER4, POWER5, POWER6, and so on.
Number Of Logical CPUs	The total number of logical CPUs configured and active in the partition.
Processor Entitled Capacity	Capacity Entitlement of the partition, represented in the unit of number of physical processors. **Note:** The physical processor units can be in fraction as well, for example, 0.5 physical processor.
Processor Max. Capacity	Maximum Capacity this partition can have, represented in the unit of number of physical processors **Note:** The physical processor units can be in fraction as well, for example, 0.5 physical processor.
True Memory	The true memory represents real physical or logical memory configured for this LPAR.
SMT Threads	Number of SMT threads configured in the partition. This can be **1**, **2** or **4**.
Shared Processor Mode	Indicates whether Shared Processor Mode is configured for this partition. The possible values are: **Disabled** Shared Processor Mode is not configured. **Enabled-Capped** Shared Processor Mode is enabled & running in capped mode. **Enabled-Uncapped** Shared Processor Mode is enabled & running in uncapped mode.
Active Memory Sharing	Indicates whether Active Memory Sharing is **Enabled** or **Diabled**
Active Memory Expansion	Indicates whether Active Memory Expansion is **Enabled** or **Diabled**
Target Expanded Memory Size	Indicates the target expanded memory size in MegaBytes for the LPAR. The Target Expanded Memory Size is the True Memory Size multiplied by the Target Memory Expansion Factor. **Note:** This get displayed only when Active Memory Expansion is enabled
Target Memory Expansion factor	Indicates the target memory expansion factor configured for the LPAR. **Note:** This get displayed only when Active Memory Expansion is enabled

System Resource Statistics: System Resource Statistics provides details about the system resource utilization from CPU/Memory Stand point. The following table shows various statistics related to system resource utilization

CPU Util	The Partition's CPU utilization in the units of number of physical processors. The percentage of utilization against the Maximum Capacity is also reported. **Note:** If Active Memory Expansion is enabled, the CPU utilization due to memory compression / decompression is also included
Virtual Memory Size	The Active Virtual Memory Size in MegaBytes. The percentage against the True Memory Size is also reported.
True Memory In-Use	This is amount of the LPAR's real physical (or logical) memory in MegaBytes. The percentage against the True Memory Size is also reported.
Pinned Memory	This represents the pinned memory size in MegaBytes. The percentage against the True Memory Size is also reported.
File Cache Size	This represents the non-computational file cache size in MegaBytes. The percentage against the True Memory Size is also reported.
Available Memory	This represents the size of the memory available, in MegaBytes, for application execution. The percentage against the True Memory Size is also reported.

Note: For all the utilization metrics Average, Minimum and Maximum values get displayed if **amepat** is run with duration/interval.

Active Memory Expansion Statistics: Active Memory Expansion Statistics provides details about the Active Memory Expansion statistics. This section is only displayed if Active Memory Expansion has been enabled for the LPAR. The following table describes the various statistics that are reported

AME CPU Usage	The CPU utilization for Active Memory Expansion activity in units of physical processors. It indicates the amount of processing capacity used for memory compression activity. The percentage of utilization against the Maximum Capacity is also reported.
Compressed Memory	The total amount of virtual memory that is compressed. This is measured in MegaBytes. The percentage against the Target Expanded Memory Size is also reported.
Compression Ratio	This represents how well the data is compressed in memory. A higher compression ratio indicates that the data compresses to a smaller size. For example, if 4 Kilobyte of data can be compressed down to 1 Kilobyte, then the compression ratio is 4.0.
Deficit Memory Size	The size of the expanded memory, in MegaBytes, deficit for the LPAR. This is only displayed if the LPAR has a memory deficit. The percentage against the Target Expanded Memory Size is also reported.

Note: The Active Memory Expansion Statistics section displays only when the tool is invoked in an Active Memory Expansion enabled machine. It also displays the average, minimum and maximum values of the statistics when the tool invoked with duration/ interval.

Active Memory Expansion Modeled Statistics: Active Memory Expansion Modeled Statistics provides details about the modeled statistics for Active Memory Expansion. The following table provides the information about the modeled statistics.

Modeled Expanded Memory Size	It represents the size of expanded memory that is used to produce the modeled statistics.
Average Compression Ratio	It represents the average compression ratio of the in-memory data of the workload. This compression ratio is used to produce the modeled statistics.
Modeled Expansion Factor	It represents the modeled target memory expansion factor.
Modeled True Memory Size	It represents the modeled true memory size (real physical or logical memory)
Modeled Memory Gain	It represents the amount of memory the partition can gain by enabling Active Memory Expansion for the reported modeled expansion factor

AME CPU Usage Estimate It represents an estimate of the CPU that would be used for Active Memory Expansion activity for the specified configuration. It estimates the amount of processing capacity that would be used for memory compression activity. The CPU usage is reported in units of physical processors. The percentage of utilization against the Maximum Capacity is also reported.
Note: This is just an estimate and should only be used as guidance; the actual usage can be higher or lower depending on the workload.

Note: This section is displayed only when **-N** flag is not used & when run by a privileged user. The generation of Modeled statistics requires Operating System to do certain simulation operation; hence the actual duration of monitoring can be higher than the user specified monitoring time.

Recommendation: Recommendation provides details about the Active Memory Expansion configuration that would provide optimal benefits to the current running workload.

Note: The recommendations are purely done based on the behavior during the monitoring period of the workload and hence the recommendations provided can be used only as guidance. The actual statistics can vary based on the actual behavior in real time of the workload.

Note: Active Memory Expansion Modeled Statistics & Recommendation are used for Workload Planning. When **-N** is specified both these reports will not be displayed. Active Memory Expansion Statistics will be reported only when running in Active Memory Expansion Enabled System.

amepat can be invoked using the System Management Interface Tool (SMIT) **smit amepat** fast path to run this command.

Note: This command is restricted inside WPAR. When **amepat** is invoked without specifying duration or interval then the utilization statistics(System, AME) will not display any Average, Minimum, or Maximum values. It just displays the Current value. The CPU utilization just displays the average from the system boot time.

Flags

-a Specifies to auto-tune the expanded memory size for Active Memory Expansion Modeled Statistics. When this option is selected, the Modeled Expanded Memory Size is estimated based on the current memory usage of the workload (excludes the available memory size).
Note: **-a** **-t** are mutually exclusive.

-c *max_ame_cpuusage%* Specifies the maximum Active Memory Expansion CPU usage in terms of percentage to be used for producing the Modeled statistics & recommendation.
Note: The default maximum used is 15%. The **-C** , **-c** option cannot be specified together. The **-c**, **-e** option are mutually exclusive.

-C *max_ame_cpuusage* Specifies the maximum Active Memory Expansion CPU usage in terms of number of physical processors to be used for producing the Modeled statistics and recommendation.
Note: The **-C**, **-c** option cannot be specified together. The **-C** , **-e** option are mutually exclusive.

-e *startexpfactor:stopexpfactor:incexpfactor*
Specifies the range of expansion factors to be reported in the Active Memory Expansion Modeled Statistics section.

Startexpfactor
Starting expansion factor. This field is mandatory if **-e** is used.

Stopexpfactor
Stop expansion factor. If not specified then the modeled statistics is generated for the start expansion factor alone.

incexpfactor
Incremental expansion factor. Allowed range is 0.01-1.0. Default is 0.5. Stop expansion factor need to be specified to specify incremental expansion factor.

Note: The **-e** option cannot be combined with **-C** or **-c** options.

-m *min_mem_gain*
Specifies the Minimum Memory Gain. This value is specified in Megabytes. This value is used in determining the various possible expansion factors reported in the Modeled Statistics & also influence the produced recommendations.

-n *num_entries*
Specifies the number of entries that need to be displayed in the Modeled Statistics.
Note: When **-e** with **incexpfactor** specified then **-n** value is ignored.

-N
Disable Active Memory Expansion Modeling (Workload Planning Capability)

-P *recfile*
Process the specified recording file and generate report.

-R *recfile*
Record the active memory expansion data in the specified recording file. The recorded data can be post processed later using the **-P** option.
Note: Only **-N** option can be combined with -R.

-t *tgt_expmem_size*
Specifies the Modeled Target Expanded Memory Size. This makes the tool to use the user specified size for modeling instead of the calculated one.
Note: **-t**, **-a** are mutually exclusive.

-u *minuncompressedpoolsize*
Specifies the minimum uncompressed pool size in Megabytes. This value over-rides the tool calculated value for producing Modeled Statistics.
Note: This flag can be used only when Active Memory Expansion is disabled.

-v
Enables Verbose Logging. When specified a verbose log file is generated, named as **amepat_yyyymmddhmm.log**, where **yyyymmddhmm** represents the time of invocation.
Note: The verbose log also contains detailed information on various samples collected and hence the file will be larger than the output generated by the tool.

Duration
Duration represents the amount of total time the tool need to monitor the system before generating any reports.
Note: When duration is specified interval/samples cannot be specified. The interval & samples will be determined by the tool automatically. The actual monitoring time can be higher than the duration specified based on the memory usage and access patterns of the workload.

Interval <Samples>
Interval represents the amount of sampling time, Samples represents the number of samples need to be collected.
Note: When interval, samples are specified, duration is calculated automatically as (interval x Samples). The actual monitoring time can be higher than the duration specified based on the memory usage and access patterns of the workload.

Exit Status

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

ATTENTION: RBAC users and Trusted AIX users:

This command can perform privileged operations. Only privileged users can 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 Active Memory Expansion Monitoring only report, enter:r:

   ```
   amepat
   ```

2. To monitor the workload, for the duration of 16 minutes with 8 minute sampling interval and 2 samples, generate report for Workload Planning, enter:

   ```
   amepat 8 2
   ```

3. To monitor the workload for a duration of 16 minutes and generate Active Memory Expansion report for Workload Planning with modeled memory expansion factors between 1.5 and 3 at 0.5 incremental factor, enter:

   ```
   amepat –e 1.50:3.00:0.5 16
   ```

4. To monitor the workload for a duration of 16 minutes and generate Active Memory Expansion report for Workload Planning with capping the modeled AME CPU usage to 30%, enter:

   ```
   amepat –c 30 16
   ```

5. To monitor the workload for a duration of 16 minutes and generate Active Memory Expansion report for Workload Planning with starting modeled memory gain of 1000 MB, enter:

   ```
   amepat –m 1000 16
   ```

6. To monitor the workload for a duration of 16 minutes and generate Active Memory Expansion report for Workload Planning by modeling a minimum uncompressed pool size 2000 MB, enter:

   ```
   amepat –u 2000 16
   ```

7. To use the recording mode of **amepat** to generate the recording file and generate reports with various filters, enter:

   Start Recording for a duration of 60 minutes.

   ```
   amepat –R myrecord_amepat 60
   ```

   Note: The recording mode will switch itself into background process.

   Generate Report for Workload Planning

   ```
   amepat -P myrecord_amepat
   ```

   Generate Report for Workload Planning with the modeled memory expansion factors ranging between 2 to 4 with 0.5 delta factor

   ```
   amepat -e 2.0:4.0:0.5 -P myrecord_amepat
   ```

   Generate Monitoring only report

   ```
   amepat -N -P myrecord_amepat
   ```

8. To disable Workload Planning Capability & monitor the system for 30 minutes, enter:

   ```
   amepat -N 30
   ```

---

anno Command

Purpose

Annotates messages.

Syntax

anno [**+***Folder*] [*Messages*] [**-component** *Field*] [**-inplace** | **-noinplace**] [**-text** *"String"*]

Description

The **anno** command annotates messages with text and dates. If you enter the **anno** command without any flags, the system responds with the following prompt:

```
Enter component name:
```

Typing a component name and pressing the Enter key annotates the component name and system date to the top of the message being processed. You cannot annotate an existing field. You can only add lines to the top of a message file. The annotation fields can contain only alphanumeric characters and dashes.

Note: To simply add distribution information to a message, use the **dist**, **forw**, or **repl** commands.

Flags

-component *Field*
: Specifies the field name for the annotation text. The *Field* variable must consist of alphanumeric characters and dashes. If you do not specify this flag, the **anno** command prompts you for the name of the field.

+*Folder*
: Identifies the message folder that contains the message to annotate. The default is the current folder.

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

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

-inplace
: Forces annotation to be done in place in order to preserve links to the annotated messages.

Messages
: Specifies what messages to annotate. This parameter can specify several messages, a range of messages, or a single message. If several messages are specified, the first message annotated becomes the current message. Use the following references to specify messages:

  Number
  : Number of the message. When specifying several messages, separate each number with a comma. When specifying a range, separate the first and last number in the range with a hyphen.

  Sequence
  : A group of messages specified by the user. Recognized values include:

    all All messages in the folder.

    cur or . (period)
    Current message. This is the default.

    first First message in a folder.

    last Last message in a folder.

    next
    Message following the current message.

    prev Message preceding the current message.

-noinplace
: Prevents annotation in place. This flag is the default.

-text *"String"*
: Specifies the text to be annotated to the messages. The text must be enclosed with quotation marks.

Profile Entries

The following entries can be made to the *UserMhDirectory*/**.mh_profile** file:

`Current-Folder:` Sets the default current folder.

`Path:` Specifies the location of a user's MH (Message Handler) directory.

Security

RBAC Disclaimer

Examples

1. To annotate the message being processed with the date and time, enter:

   ```
   anno
   ```

   The following prompt is displayed on your screen:

   ```
   Enter component name: _
   ```

   After responding to this prompt, type:

   ```
   Date
   ```

   Press Enter. The component name you entered becomes the prefix to the date and time on the message. The caption appended to the message is similar to the following:

   ```
   Date: Tues, 28 Mar 89 13:36:32 -0600
   ```

2. To annotate the message being processed with the date, time, and a message, enter:

   ```
   anno  -component NOTE  -text "Meeting canceled."
   ```

   A two-line caption similar to the following is appended to the message:

   ```
   NOTE: Mon, 15 Mar 89 10:19:45 -0600
   NOTE: Meeting canceled.
   ```

3. To annotate message `25` in the `meetings` folder, enter:

   ```
   anno  +meetings 25  -component NOTE  -text "Meeting delayed
   until Friday."
   ```

   The top of message 25 is annotated with a caption similar to the following:

   ```
   NOTE: Wed, 19 Jun 87 15:20:12 -0600
   NOTE: Meeting delayed until Friday.
   ```

   Note: Do not press the Enter key until the entire message has been entered, even though the message may be wider than the screen.

Files

$HOME/.mh_profile	Contains the MH user profile.
/usr/bin/anno	Contains **anno** command.

Related Information

The **dist** command, **forw** command, **repl** command.

The **mh_alias** file format, **mh_profile** file format.

Mail applications in *Networks and communication management*.

ap Command

Purpose

Parses and reformats addresses.

Syntax

ap [**-form** *File* | **-format** *String*] [**-normalize** | **-nonormalize**] [**-width** *Number*] *Address*

Description

The **ap** command parses and reformats addresses. The **ap** command is not started by the user. The **ap** command is called by other programs. The command is typically called by its full path name, **/usr/lib/mh/ap**.

The **ap** command parses each string specified by the address parameter and attempts to reformat it. The default output format for the **ap** command is the ARPA RFC 822 standard. When the default format is used, the **ap** command displays an error message for each string it is unable to parse.

Alternate file and string formats are specified by using the **-form** and **-format** flags.

Flags

-form *File*
Reformats the address string specified by the *Address* parameter into the alternate format described in the *File* variable.

-format *String*
Reformats the address string specified by the *Address* parameter into the alternate format specified by the *String* variable. The default format string follows:

```
%<{error}%{error}:%{Address}%:%(putstr(proper{
   Address}))%>
```

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

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

-nonormalize
Does not attempt to convert local nicknames of hosts to their official host names.

-normalize
Attempts to convert local nicknames of hosts to their official host names. This flag is the default.

-width *Number*
Sets the maximum number of columns the **ap** command uses to display dates and error messages. The default is the width of the display.

Files

/etc/mh/mtstailor
Contains the MH tailor file.

$HOME/.mh_profile
Contains the MH user profile.

Related Information

The **ali** command, **dp** command, **scan** command.

The **.mh_alias** file format.

Mail applications in *Networks and communication management*.

apply Command

Purpose

Applies a command to a set of parameters.

Syntax

apply [**-a***Character*] [**-***Number*] *CommandString Parameter* ...

Description

The **apply** command runs a command string specified by the *CommandString* parameter on each specified value of the *Parameter* parameter in turn. Normally, *Parameter* values are chosen individually; the optional **-***Number* flag specifies the number of *Parameter* values to be passed to the specified command string. If the value of the *Number* variable is 0, the command string is run without parameters once for each *Parameter* value.

If you include character sequences of the form %*n* (where *n* is a digit from 1 to 9) in *CommandString*, they are replaced by the *n*th unused *Parameter* value following the *CommandString* parameter when the command string is executed. If any such sequences occur, the **apply** command ignores the **-***Number* flag, and the number of parameters passed to *CommandString* is the maximum value of *n* in the *CommandString* parameter.

You can specify a character other than % (percent sign) to designate parameter substitution character strings with the **-a** flag; for example, `-a@` would indicate that the sequences @1 and @2 would be replaced by the first and second unused parameters following the *CommandString* parameter.

Notes:

1. Because pattern-matching characters in *CommandString* may have undesirable effects, it is recommended that complicated commands be enclosed in ' ' (single quotation marks).
2. You cannot pass a literal % (percent sign) followed immediately by any number without using the **-a** flag.

Flags

-a*Character*	Specifies a character (other than %) to designate parameter substitution strings.
-*Number*	Specifies the number of parameters to be passed to *CommandString* each time it is run.

Examples

1. To obtain results similar to those of the **ls** command, enter:
   ```
   apply echo *
   ```
2. To compare the file named **a1** to the file named **b1**, and the file named **a2** to the file named **b2**, enter:
   ```
   apply -2 cmp a1 b1 a2 b2
   ```
3. To run the **who** command five times, enter:
   ```
   apply -0 who 1 2 3 4 5
   ```
4. To link all files in the current directory to the directory **/usr/joe**, enter:
   ```
   apply 'ln %1 /usr/joe' *
   ```

Related Information

The **xargs** command.

Input and output redirection overview in *Operating system and device management*.

Shells in *Operating system and device management*.

apropos Command

Purpose

Locates commands by keyword lookup.

Syntax

apropos [**-M** *PathName*] *Keyword ...*

Description

The **apropos** command shows the manual sections that contain any of the keyword*s* specified by the *Keyword* parameter in their title. The **apropos** command considers each word separately and does not take into account if a letter is in uppercase or lowercase. Words that are part of other words are also displayed. For example, when looking for the word `compile`, the **apropos** command also finds all instances of the word `compiler`. The database containing the keywords is **/usr/share/man/whatis**, which must first be generated with the **catman -w** command.

If the output of the **apropos** command begins with a name and section number, you can enter **man** *Section Title*. For example, if the output of the **apropos** command is `printf(3)`, you can enter `man 3 printf` to obtain the manual page on the **printf** subroutine.

The **apropos** command is equivalent to using the **man** command with the **-k** option.

> **Note:** When the **/usr/share/man/whatis** database is built from the HTML library using the **catman -w** command, section 3 is equivalent to section 2 or 3. See the **man** command for further explanation of sections.

Flag

-M *PathName*	Specifies an alternative search path. The search path is specified by the *PathName* parameter, and is a colon-separated list of directories.

Examples

1. To find the manual sections that contain the word `password` in their titles, enter:
   ```
   apropos password
   ```
2. To find the manual sections that contain the word `editor` in their titles, enter:
   ```
   apropos editor
   ```

File

/usr/share/man/whatis	Contains the **whatis** database.

Related Information

The **catman** command, **man** command, **whatis** command.

ar Command

Purpose

Maintains the indexed libraries used by the linkage editor.

Syntax

ar [**-c**] [**-l**] [**-g** | **-o**] [**-s**] [**-v**] [**-C**] [**-T**] [**-z**] { **-h** | **-p** | **-t** | **-x** } [**-X** {**32**|**64**|**32_64**|**d64**| **any**}] *ArchiveFile* [*File* ...]

ar [**-c**] [**-l**] [**-g** | **-o**] [**-s**] [**-v**] [**-C**] [**-T**] [**-z**] { **-m** | **-r** [**-u**] } [{ **-a** | **-b** | **-i** } *PositionName*] [**-X** {**32**|**64**|**32_64**|**d64**|**any**}] *ArchiveFile File* ...

ar [**-c**] [**-l**] [**-g** | **-o**] [**-s**] [**-v**] [**-C**] [**-T**] [**-z**] { **-d** | **-q** } [**-X** {**32**|**64**|**32_64**|**d64**|**any**}] *ArchiveFile File* ...

ar [**-c**] [**-l**] [**-v**] [**-C**] [**-T**] [**-z**] { **-g** | **-o** | **-s** | **-w** } [**-X** {**32**|**64**|**32_64**|**d64**|**any**}] *ArchiveFile*

Description

The **ar** command maintains the indexed libraries used by the linkage editor. The **ar** command combines one or more named files into a single archive file written in **ar** archive format. When the **ar** command creates a library, it creates headers in a transportable format; when it creates or updates a library, it rebuilds the symbol table. See the **ar** file format entry for information on the format and structure of indexed archives and symbol tables.

There are two file formats that the **ar** command recognizes. The Big Archive Format, **ar_big**, is the default file format and supports both 32-bit and 64-bit object files. The Small Archive Format can be used to create archives that are recognized on versions older than AIX 4.3, see the **-g** flag. If a 64-bit object is added to a small format archive, **ar** first converts it to the big format, unless **-g** is specified. By default, **ar** only handles 32-bit object files; any 64-bit object files in an archive are silently ignored. To change this behavior, use the **-X** flag or set the **OBJECT_MODE** environment variable.

Flags

In an **ar** command, you can specify any number of optional flags from the set **cClosTv**. You must specify one flag from the set of flags **dhmopqrstwx**. If you select the **-m** or **-r** flag, you may also specify a positioning flag (**-a**, **-b**, or **-i**); for the **-a**, **-b**, or **-i** flags, you must also specify the name of a file within *ArchiveFile* (*PositionName*), immediately following the flag list and separated from it by a blank.

-a *PositionName*	Positions the named files after the existing file identified by the *PositionName* parameter.
-b *PositionName*	Positions the named files before the existing file identified by the *PositionName* parameter.
-c	Suppresses the normal message that is produced when *library* is created.
-C	Prevents extracted files from replacing like-named files in the file system.
-d	Deletes the named files from the library.
-g	Orders the members of the archive to ensure maximum loader efficiency with a minimum amount of unused space. In almost all cases, the **-g** flag physically positions the archive members in the order in which they are logically linked. The resulting archive is always written in the small format, so this flag can be used to convert a big-format archive to a small-format archive. Archives that contain 64-bit **XCOFF** objects cannot be created in or converted to the small format.
-h	Sets the modification times in the member headers of the named files to the current date and time. If you do not specify any file names, the **ar** command sets the time stamps of all member headers. This flag cannot be used with the **-z** flag.

-i *PositionName* Positions the named files before the existing file identified by the *PositionName* parameter (same as the **-b**).

-l Places temporary files in the current (local) directory instead of the **TMPDIR** directory (by default **/tmp**).

-m Moves the named files to some other position in the library. By default, it moves the named files to the end of the library. Use a positioning flag (**abi**) to specify some other position.

-o Orders the members of the archive to ensure maximum loader efficiency with a minimum amount of unused space. In almost all cases, the **-o** flag physically positions the archive members in the order in which they are logically linked. The resulting archive is always written in the big archive format, so this flag can be used to convert a small-format archive to a big-format archive.

-p Writes to standard output the contents of the named in the *Files* parameter, or all files specified in the *ArchiveFile* parameter if you do not specify any files.

-q Adds the named files to the end of the library. In addition, if you name the same file twice, it may be put in the library twice.

-r Replaces a named file if it already appears in the library. Because the named files occupy the same position in the library as the files they replace, a positioning flag does not have any additional effect. When used with the **-u** flag (update), the **-r** flag replaces only files modified since they were last added to the library file.

If a named file does not already appear in the library, the **ar** command adds it. In this case, positioning flags do affect placement. If you do not specify a position, new files are placed at the end of the library. If you name the same file twice, it may be put in the library twice.

-s Forces the regeneration of the library symbol table whether or not the **ar** command modifies the library contents. Use this flag to restore the library symbol table after using the **strip** command on the library.

-t Writes to the standard output a table of contents for the library. If you specify file names, only those files appear. If you do not specify any files, the **-t** flag lists all files in the library.

-T Allows file name truncation if the archive member name is longer than the file system supports. This option has no effect because the file system supports names equal in length to the maximum archive member name of 255 characters.

-u Copies only files that have been changed since they were last copied (see the **-r** flag discussed previously).

-v Writes to standard output a verbose file-by-file description of the making of the new library. When used with the **-t** flag, it gives a long listing similar to that of the **ls -l** command. When used with the **-x** flag, it precedes each file with a name. When used with the **-h** flag, it lists the member name and the updated modification times.

-w Displays the archive symbol table. Each symbol is listed with the name of the file in which the symbol is defined.

-x Extracts the named files by copying them into the current directory. These copies have the same name as the original files, which remain in the library. If you do not specify any files, the **-x** flag copies all files out of the library. This process does not alter the library.

-X *mode*
: Specifies the type of object file **ar** should examine. The *mode* must be one of the following:

  32 Processes only 32-bit object files

  64 Processes only 64-bit object files

  32_64 Processes both 32-bit and 64-bit object files

  d64 Examines discontinued 64-bit XCOFF files (magic number == U803XTOCMAGIC).

  any Processes all of the supported object files.

  The default is to process 32-bit object files (ignore 64-bit objects). The *mode* can also be set with the **OBJECT_MODE** environment variable. For example, **OBJECT_MODE=64** causes **ar** to process any 64-bit objects and ignore 32-bit objects. The **-X** flag overrides the **OBJECT_MODE** variable.

-z
: Creates a temporary copy of the archive and performs all requested modifications to the copy. When all operations have completed successfully, the working copy of the archive is copied over the original copy. This flag cannot be used with the **-h** flag.

ArchiveFile
: Specifies an archive file name; required.

MemberName ...
: Names of individual archive members.

Exit Status

This command returns the following exit values:

0 Successful completion.

>0 An error occurred.

Examples

1. To create a library, enter:

   ```
   ar -v -q lib.a strlen.o strcpy.o
   ```

   If the `lib.a` library does not exist, this command creates it and enters into it copies of the files `strlen.o` and `strcpy.o`. If the `lib.a` library does exist, then this command adds the new members to the end without checking for duplicate members. The **v** flag sets verbose mode, in which the **ar** command displays progress reports as it proceeds.

2. To list the table of contents of a library, enter:

   ```
   ar -v -t lib.a
   ```

   This command lists the table of contents of the `lib.a` library, displaying a long listing similar to the output of the **ls -l** command. To list only the member file names, omit the **-v** flag.

3. To replace or add new members to a library, enter:

   ```
   ar -v -r lib.a strlen.o strcat.o
   ```

   This command replaces the members `strlen.o` and `strcat.o`. If `lib.a` was created as shown in example 1, then the `strlen.o` member is replaced. A member named `strcat.o` does not already exist, so it is added to the end of the library.

4. To specify where to insert a new member, enter:

   ```
   ar -v -r -b strlen.o lib.a strcmp.o
   ```

   This command adds the `strcmp.o` file, placing the new member before the `strlen.o` member.

5. To update a member if it has been changed, enter:

   ```
   ar -v -r -u lib.a strcpy.o
   ```

This command replaces the existing strcpy.o member, but only if the file strcpy.o has been modified since it was last added to the library.

6. To change the order of the library members, enter:

```
ar -v -m -a strcmp.o lib.a strcat.o strcpy.o
```

   This command moves the members strcat.o and strcpy.o to positions immediately after the strcmp.o member. The relative order of the strcat.o and strcpy.o members is preserved. In other words, if the strcpy.o member preceded the strcat.o member before the move, it still does.

7. To extract library members, enter:

```
ar -v -x lib.a strcat.o strcpy.o
```

   This command copies the members strcat.o and strcpy.o into individual files named strcat.o and strcpy.o, respectively.

8. To extract and rename a member, enter:

```
ar -p lib.a strcpy.o >stringcopy.o
```

   This command copies the member strcpy.o to a file named stringcopy.o.

9. To delete a member, enter:

```
ar -v -d lib.a strlen.o
```

   This command deletes the member strlen.o from the lib.a library.

10. To create an archive library from multiple shared modules created with the **ld** command, enter:

```
ar -r -v libshr.a shrsub.o shrsub2.o shrsub3.o ...
```

    This command creates an archive library named libshr.a from the shared modules named shrsub.o, shrsub2.o, shrsub3.o, and so on. To compile and link the main program using the libshr.a archive library, use the following command:

```
cc -o main main.c -L/u/sharedlib -lshr
```

    The main program is now executable. Any symbols referenced by the main program that are contained by the libshr.a archive library have been marked for deferred resolution. The **-l** flag specifies that the libshr.a library be searched for the symbols.

11. To list the contents of **lib.a**, ignoring any 32-bit object file, enter:

```
ar -X64 -t -v lib.a
```

12. To extract all 32-bit object files from **lib.a**, enter:

```
ar -X32 -x lib.a
```

13. To list all files in **lib.a**, whether 32-bit, 64-bit, or non-objects, enter:

```
ar -X32_64 -t -v lib.a
```

File

/tmp/ar* Contains temporary files.

Related Information

The **ld** command, **lorder** command, **make** command, **nm** command, **strip** command.

---

arithmetic Command

Purpose

Tests arithmetic skills.

Syntax

arithmetic [**+**][**-**][**x**][**/**][*Range*]

Description

The **arithmetic** command displays simple arithmetic problems and waits for you to enter an answer. If your answer is correct, the program displays `Right!` and presents a new problem. If your answer is wrong, it displays `What?` and waits for another answer. After a set of 20 problems, the **arithmetic** command displays the number of correct and incorrect responses and the time required to answer.

The **arithmetic** command does not give the correct answers to the problems it displays. It provides practice rather than instruction in performing arithmetic calculations.

To quit the game, press the Interrupt (Ctrl-C) key sequence; the **arithmetic** command displays the final game statistics and exits.

Flags

The optional flags modify the action of the **arithmetic** command. These flags are:

+	Specifies addition problems.
-	Specifies subtraction problems.
x	Specifies multiplication problems.
/	Specifies division problems.
Range	A decimal number that specifies the permissible range of numbers. This range goes up to and includes 99. For addition and multiplication problems, the range applies to all numbers (except answers). For subtraction and division problems, the range applies only to the answers. At the start of the game, all numbers within this range are equally likely to appear. If you make a mistake, the numbers in the problem you missed become more likely to reappear.

If you do not select any flags, the **arithmetic** command selects addition and subtraction problems and a default range of 10. If you give more than one problem specifier (**+**, **-** ,**x**, **/**), the program mixes the specified types of problems in random order.

Examples

1. To drill on addition and subtraction of integers from 0 to 10:
   ```
   arithmetic
   ```
2. To drill on addition, multiplication, and division of integers from 0 to 50:
   ```
   arithmetic  +x/  50
   ```

File

/usr/games	Location of the system's games.

Related Information

The **back** command, **bj** command, **craps** command, **fish** command, **fortune** command, **hangman** command, **moo** command, **number** command, **quiz** command, **ttt** command, **turnoff** command, **turnon** command, **wump** command.

arp Command

Purpose

Displays and modifies address resolution, including ATM (Asynchronous Transfer Mode) interfaces.

Syntax

To Display ARP Entries

arp { [**-t** *ifType*] *HostName* | **-a** [**n**] [**/dev/kmem**] }

To Display ARP ATM Entries

arp { **-t atm** *HostName* | **-a** [**n**] [**/dev/kmem**] [**pvc** | **svc**] }

To Delete an ARP Entry

arp [**-t** *ifType*] **-d** *HostName*

To Delete a PVC ARP ATM Entry

arp -t atm -d pvc *vpi:vci* **if** *ifName*

To Create an ARP Entry

arp [**-t** *ifType*] **-s** *Type HostName AdapterAddress* [*Route*] [**temp**] [**pub**]

To Create an SVC ARP ATM Entry

arp -t atm -s *Type HostName AdapterAddress* [**temp**]

To Create a PVC ARP ATM Entry

arp -t atm -s *Type* **pvc** *vpi:vci* { *HostName* | **if** *ifName }* [**no-llc**] [**no-arp**] [**temp**]

To Import ARP Entries from Another File

arp [**-t** *ifType*] **-f** *FileName* [*Type*]

Description

The **arp** command displays and modifies the Internet-to-adapter address translation tables used by the Address in *Networks and communication management*. The **arp** command displays the current ARP entry for the host specified by the *HostName* variable. The host can be specified by name or number, using Internet dotted decimal notation.

Flags

-a

Used as { [**-t** *ifType*] *HostName* | **-a** [**n**] [**/dev/kmem**] }

Displays all of the current ARP entries. Specify the **-a /dev/kmem** flag to display ARP information for kernel memory. The '**n**' modifier causes hostname lookups to be suppressed.

Used as { **-t atm** *HostName* | **-a** [**n**] [**/dev/kmem**] [**pvc** | **svc**] }

The **pvc** specification will display only ATM PVC (Permanent Virtual Circuits) types of virtual circuits, **svc** specification will display only ATM SVC (Switched Virtual Circuits) types of virtual circuits. If the **pvc** | **svc** parameter is omitted, all ATM virtual circuits will be displayed.

-d

Used as [**-t** *ifType*] **-d** *HostName*

Deletes an entry for the host specified by the *HostName* variable if the user has root user authority.

Used as **-t atm -d pvc** *vpi:vci* **if** *ifName*

Deletes a PVC ARP entry by specifying *vpi:vci* rather than hostname. The *vpi:vc*i variables specify the virtual circuit that is to be deleted. The *ifname* variable specifies the name of the ATM interface on which the virtual circuit is to be deleted.

-f *FileName* [Type]

Causes the file specified by the *FileName* variable to be read and multiple entries to be set in the ARP tables. Entries in the file should be in the form:

```
[Type] HostName AdapterAddress [Route] [temp] [pub]
```

where

Type Specifies the type of hardware address. If the address type is specified when invoking **arp** from the command line, it should not be specified in the file entries. Otherwise, it should be specified in each file entry. Valid hardware address types are:

- `ether` for an Ethernet interface
- `802.3` for an 802.3 interface
- `fddi` for a Fiber Distributed Data interface
- `802.5` for a Token-Ring interface

HostName
Specifies the remote host.

AdapterAddress
Specifies the hardware address of the adapter for this host as 6 hexadecimal bytes separated by colons. Use the **netstat -v** command to display the local hardware address.

Route Specifies the route for a Token-Ring interface or Fiber Distributed Data Interface (FDDI) as defined in the Token-Ring or FDDI header.

temp Specifies that this ARP table entry is temporary. The table entry is permanent if this argument is omitted.

pub Specifies that this table entry is to be published, and that this system will act as an ARP server responding to requests for `HostName`, even though the host address is not its own.

Note: The **-f** flag is not supported for ATM.

-s

Used as [**-t** *ifType*] **-s** *Type HostName AdapterAddress* [*Route*] [**temp**] [**pub**]

Creates an ARP entry of the type specified by the *Type* variable for the host specified by the *HostName* variable with the adapter address specified by the *AdapterAddress* variable. Only users with root authority can use the **-s** flag. The adapter address is given as 6 hexadecimal bytes separated by colons. The line must be in the following format:

```
Type HostName AdapterAddress [Route] [temp] [pub]
```

where the *Type*, *HostName*, *AdapterAddress*, *Route*, **temp**, and **pub** parameters have the same purpose and definitions as the parameters for the **-f** flag.

Used as **-t atm -s** *Type HostName AdapterAddress* [**temp**]

Creates a SVC type of ARP entry for the remote host, specified by the *HostName* variable, with the ATM address specified by the *ATMAddress* variable. The ATM address is given as 20 hexadecimal bytes separated by colons. Creation of this entry causes this IP station to not use ARP server mechanism to resolve IP addresses.

Used as **-t atm -s** *Type* **pvc** *vpi:vci* { *HostName* | **if** *ifName }* [**no-llc**] [**no-arp**] [**temp**]

Creates a PVC type of ARP entry for the remote host, specified by the *HostName* variable, with the PVC specified by the *vpi:vci*. Either destination *Hostname* or the local *ifname* needs to be specified. The **no-llc** flag is used to indicate that LLC/SNAP encapsulation will not be used on this virtual circuit, in this case, the destination *Hostname* needs to be specified. The **no-arp** flag is used to indicate that ARP protocol will not be used on this virtual circuit, in this case, the destination *Hostname* needs to be specified.

The *temp* parameter specifies that this ARP table entry is temporary, the table entry is permanent if this argument is omitted.

-t *ifType*

The **-t** *iftype* flag is used to indicate the type of Network interface. This flag is only required for the following interfaces:

- at for ATM
- ib for infiniband

Security

RBAC Disclaimer

Examples

1. To add a single entry to the **arp** mapping tables until the next time the system is restarted, type:

   ```
   arp -s 802.3 host2 0:dd:0:a:85:0 temp
   ```

2. To delete a map table entry for the specified host with the **arp** command, type:

   ```
   arp   -d   host1 flag
   ```

3. To display arp entries for atm host `host1` , type:

   ```
   arp -t atm -a host1
   ```

4. To add a PVC arp entry for atm host `host2`, type:

   ```
   arp   -t atm   -s atm pvc 0:20 host2
   ```

5. To add a PVC arp entry for an interface `at0`, type:

   ```
   arp   -t atm   -s atm pvc 0:20 if at0
   ```

Related Information

The **ifconfig** command, **netstat** command.

The **inetd** daemon.

TCP/IP protocols in *Networks and communication management*.

---

artexdiff Command

Purpose

The **artexdiff** command compares the parameters and values between two profiles or between a profile and a system.

Syntax

artexdiff [**-a** | **-u** | **-c**] [**-d** | **-s**] [**-r** | **-n**]
[**-p** | **-f** { *txt* | *csv* | *xml* }]*profileA* [*profile*]

Description

The **artexdiff** command compares the parameters and values between two profiles, or between a profile and a system.

When the comparison is between a profile and a system, the current values of the parameters of the running system are compared. If the current value cannot be retrieved, then it compares with **nextboot** values. If **–n** option is specified, then the comparison uses the **nextboot** values for the systems with the parameters specified in the profile.

This command displays the output in three different formats to `stdout`. This output can be saved into a file using the redirector (>). If none of the output formats are specified, it displays in XML format. If Comma Separated Values (CSV) format (`-f csv`) is specified, then it displays in csv format, which can be used to open in a spreadsheet. If a text format (`-f txt`) is specified, the output will be in a table like readable format. When text format is specified, the output format can be either **diff** command output format (**-d** option) or **sdiff** command output format (**-s** option). So, the **-s** and **-d** flags can only be used in conjunction with the **-f** txt flag. When the **–p** option is specified, this command generates XML output in profile format that includes the parameters and values from the profile that are different from the system. Use the XML output in profile format to set the system by calling the **artexset** command. This ensures that the system is compliant with the input profile. When the **–p** option is specified, the output is always XML in profile format .

Selection criteria, as specified by the **-u** or **-c** flags, indicate how to list the comparison results. When no selection criteria is specified, all comparison results display. If the **–c** option is specified, only parameters that are different in the comparison are displayed. If the **–u** option is specified, only the parameters that have the same values are displayed.

The specified profile can exist on the local file system using a relative or absolute path or on an LDAP server.

Flags

-a	Indicates that **artexdiff** output will be recorded in the AIX audit log.
-u	Indicates to output only the values found by the comparison that are found to be identical. If neither **-u** nor **-c** is specified, all parameter values are noted in the output.
-c	Indicates to output only the values found by the comparison that are found to be different. If neither **-u** nor **-c** is specified, all parameter values are noted in the output.

-d Indicates to output the comparison results into a format like the **diff** command.

-s Indicates to output the comparison results into a format like the **sdiff** command.

-r Indicates to use the system's current values for comparison. This option is only valid when the comparison includes a system.

-n Indicates to use the system's `nextboot` values for comparison. This option is only valid when the comparison includes a system.

-p Generates XML output in profile format that includes the parameters and values from the profile that are different from the system. This option is valid only when the comparison is between a profile and a system.

-f Specifies the output formats. Possible formats include the following:

- The *txt* option indicates to use plain text format. The flags **–d** and **–s** can be used only when this **-f** flag is set.
- The *csv* option indicates to use comma-separated values format.
- The *xml* option indicates to use xml format. This is the default format.

Parameters

profileA Specifies the filename for the profile that lists the tunables by which all other information is gathered for comparison.

profile Specifies the filename for the profile to compare to the profile noted by the *profileA* parameter. If no profile is specified for the *profile* parameter, the comparison is performed against *profileA* and the system.

Limitations

The filtering options **-u** and -**c**, formatting options **-d**, **-s** and **-f** <format> are not supported with custom parameters like blob, binary data that do not use internal data structure format.

Exit Status

0 The command completed successfully and no differences were found.

1 Differences were found.

>1 An error occurred.

Examples

The following example illustrates how to compare the parameters and values between two profiles.

```
artexdiff profile1.xml profile2.xml
```

The following example illustrates how to compare the parameters and values between the `ldap_profile.xml` profile stored on LDAP server and the system.

```
artexdiff ldap://ldap_profile.xml
```

The following example illustrates how to create a new profile with the parameters and values from an input profile that are different from the system.

```
artexdiff -p profile.xml > diff_profile.xml
```

artexget Command

Purpose

The **artexget** command lists the configuration and tuning parameter information from a specified profile or from the system.

Syntax

artexget [**-p** | **-r** | **-n**] [**-l** {*dynamic* | *disruptive* | *reboot* }] [**-f** [*txt* | *csv* | *xml*]] *profile*]

Description

The **artexget** command lists the configuration and tuning parameter information from a profile or from the system. If none of the options **-p**, **-r**, or **-n** are specified, the command outputs the parameter and value pairs from the argument *profile*. If **-r** option is specified, the command outputs the current values of the parameters from the system. If the **-n** option is specified, the command outputs the values of the parameters after the next system restart. If **-p** option is specified, it outputs either current values of the parameters or values of the parameters after the next system restart, based on the `applyType` attribute value in the profile.

This command can also list the subset of the parameters based on selection criteria. If no selection criteria is specified, the command outputs a list of all parameters listed in the profile. If dynamic selection criteria (`-l dynamic`) is specified, then the command outputs a list of the parameters that do not require a reboot or disruptive action, like stopping and restarting a service or unmounting and mounting a file system. If disruptive selection criteria (`-l disruptive`) is specified, then the command outputs a list of parameters that need a disruptive action. If the selection criteria reboot (`-l reboot`) is specified, the command outputs a list of parameters that require a reboot.

This command displays the output in three different formats to `stdout`. This output can be saved into a file using the redirector (>). If none of the output formats are specified, the output displays in XML format. If Comma Separated Values (CSV) format (`-f csv`) is specified, then it displays in csv format, which can be used to open in a spreadsheet. If a text format (`-f txt`) is specified, the output will be in a table like readable format.

The specified profile can exist on the local file system using a relative or absolute path or on an LDAP server.

Flags

-p	Lists either the current values of the parameters or values of the parameters after the next system restart, based on the `applyType` attribute value in the profile.
-r	Lists the current values on the running system.
-n	Lists the values of the parameters after the next system restart. If the **-p**, **-r**, or **-n** option is not specified, then list the tunable values described by the profile.
-l { *dynamic* | *disruptive* | *reboot* }	Indicates what tunable values to list in the output. The **-l** flag has the following options: • The *dynamic* variable indicates to list the tunable values that take effect immediately. • The *disruptive* variable indicates to list the tunable values that require recycling of a resource. • The *reboot* variable indicates to list the tunable values that require a system reboot in order to take effect.
-f { *txt* | *csv* | *xml* }	Specifies the output format. The **-f** flag has the following variables: • The *txt* variable specifies plain text format. • The *csv* variable specifies comma separated values format. • The *xml* format specifies xml format. This is the default format.

Parameters

profile	This is a mandatory file. The file specified includes a list of the tunable parameters.

Exit Status

0	The command completed successfully
>0	An error occurred.

Examples

The following example illustrates how to output the parameter and value pairs from the `profile1.xml` profile that is stored on an LDAP server.

```
artexget ldap://profile1.xml
```

The following example illustrates how to output the values of parameters after the next system restart from the system using the `local_profile.xml` profile.

```
artexget -n local_profile.xml
```

The following example illustrates how to output the current values of the parameters in text format from the system using the `local_profile.xml` profile.

```
artexget -r -f txt local_profile.xml
```

---

artexlist Command

Purpose

Outputs a list of profiles from the local system or LDAP server or outputs a list of catalogs that are installed on the local system.

Syntax

artexlist [**-c** | [**-l**] **path**]

Description

The command **artexlist** finds and lists the AIX Runtime Expert profiles on the local system or on LDAP server.

If the **-c** option is specified, the output returns a list of catalogs that are installed on the local system rather than a list of profiles.

By default, this command outputs a list of the profiles from `/etc/security/artex/samples` directory. To override the default path, set the environment variable ARTEX_PROFILE_PATH to one or more semicolon delimited paths. Otherwise, use the *path* argument. In addition to the local system profiles, use the **–l** option to list the profiles from the LDAP server.

Flags

-c	Indicates to list the catalogs installed on the local system in `/etc/security/artex/catalogs` directory.
-l	Indicates to list the profiles from the LDAP server.
path	Specifies the path on the local system that contains the list of profiles that are to be returned in the output.

Exit Status

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

Examples

The following example illustrates how to list the sample profiles from the default path `/etc/security/artex/samples`.

```
artexlist
```

The following example illustrates how to list the profiles using environment variable ARTEX_PROFILE_PATH.

```
export ARTEX_PROFILE_PATH="/tmp:/$HOME/profiles"
artexlist
```

The following example illustrates how to list the profiles from `/data/profiles` directory.

```
artexlist /data/profiles
```

The following example illustrates how to list the profiles from an LDAP server and from a local system.

```
artexlist –l
```

The following example illustrates how to list the catalogs installed on the system.

```
artexlist –c
```

artexmerge Command

Purpose

The **artexmerge** command merges two or more profiles.

Syntax

artexmerge [**-v** | **-t**][**-f**] *profile . . .*

Description

The command **artexmerge** merges two or more profiles and displays the output to `stdout`. You can also save the output to a file using the redirector (>).

When merging the profiles, the command returns an error if a parameter exists in more than one profile. To override this error condition, use the **-f** option. The **-f** option indicates to use the parameter and value from the last profile listed in the command syntax.

The **artexmerge** command validates the parameters of the profiles specified to be merged. If the **–v** option is specified, the parameters for each profile specified are verified prior to the merge. If the **–t** option is specified, the verification is performed after the profiles are merged. These two options are mutually exclusive.

The specified profiles can exist on the local file system using a relative or absolute path or on an LDAP server.

Flags

-v	Indicates to verify that each profile's parameters are valid prior to the merge. The output will list the profiles, along with its invalid parameters.
-t	Indicates to verify the profile's parameters against the merged profile rather than prior to the merge.
-f	Indicates to force the merge. If two or more profiles contain the same parameter, use the value of the parameter noted in the later profile.

Parameters

profile . . .	Lists the filenames of the profiles to merge, separated by a space. For example, `profileA profileB profileC`.

Exit Status

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

Examples

The following example illustrates how to combine profiles located on a LDAP server and on a local file system.

```
artexmerge /tmp/no_profile1.xml ldap://ldap_raso_profile.xml /data/nfs_profile.xml
```

The following example illustrates how to combine two profiles with duplicate parameters and save as `merged_profile.xml`.

```
artexmerge –f profile1.xml profile2.xml > merged_profile.xml
```

---

artexremset Command

Purpose

artexremset command executes **artexset** command on one or more remote systems.

Syntax

artexremset [[[[[**-q**] [**-c**] [**-r**] [**-R**]] | **-t** | **-p**] [**-l** {*dynamic* | *noreboot* | *reboot* | *all*}]] | **-b** | **-x** | **-u**] [**-L**] [**-D**] *profile* {*clientname* | *nim_mac_group*}

artexremset [**-q**] [**-c**] [**-r**] [**-R**] [**-l** {*dynamic* | *noreboot* | *reboot* | *all*}] [**-L**] [**-D**] *profile* {*clientname* | *nim_mac_group*}

artexremset [**-l** {*dynamic* | *noreboot* | *reboot* | *all*}] **-t** [**-L**] [**-D**] *profile* {*clientname* | *nim_mac_group*}

artexremset [**-l** {*dynamic* | *noreboot* | *reboot* | *all*}] **-p** [**-L**] [**-D**] *profile* {*clientname* | *nim_mac_group*}

artexremset -b [**-L**] [**-D**] *profile* {*clientname* | *nim_mac_group*}

artexremset -x [**-D**] {*clientname* | *nim_mac_group*}

artexremset -u [**-D**] {*clientname* | *nim_mac_group*}

Description

artexremset provides the ability to execute **artexset** commands on each client with a designated profile provided by the server or a profile stored on an LDAP server. Therefore, all the command options

designated for the local **artexset** command must also be provided by the server so these options can be directly conveyed to each client's local **artexset** command.

The **artexremset** command runs only on NIM master. When the profile is on NIM master, the **artexremset** command copies the profile to a remote client machine prior to requesting the client to execute **artexset** command. When the **–L** option is specified, the profile name given is assumed to be the pathname to a profile that exists in LDAP. Thus, no profile is copied to the client from NIM master. Instead, the LDAP pathname is packaged in the custom script file and the local **artexset** command should realize that the `ldap://` prefix represents an LDAP file.

By default, the exit status of the **artexremset** command will be a cumulative "OR" of all the remote **artexset** commands. With the **–D** option, the results of each individual NIM command result is captured and associated with each individual node and listed in a `stdout` listing.

Flags

-q	Indicates to ignore non-fatal warning messages.
-c	Indicates to verify that the **artexset** command set the values and that they were successfully applied to the system.
-r	If the **-c** option indicates that not all parameters were applied successfully, the **-r** option indicates to rollback the parameter values for the specified *profile* to their original state. To do this, the command applies the values stored in the `latest_rollback.xml` file.
-l { *dynamic* | *noreboot* | *reboot* | *all*}	Indicates the level to which to apply the command. The **-l** flag has the following options: • The *dynamic* variable indicates to apply non-disruptive parameters only. • The *noreboot* variable indicates to apply all parameters that do not need a reboot, and recycle the resources as needed. • The *reboot* variable indicates to apply only the parameters that have reboot constraint • The *all* variable indicates to apply all parameters, including the ones that need reboot.
-R	Specifies to not create a rollback profile.
-b	Indicates to enable the master profile, which is also referred to as the boot profile.
-x	Indicates to disable the master profile, which is also referred to as the boot profile. This flag is the opposite of the **-b** option. If the -x option is specified, no profile parameter is required.
-t	Indicates to test if the values listed in the *profile* are valid tunables, as recognized by the runtime system.
-p	Generates XML output that includes the parameters and values from the profile that are different from the system. This option is valid only when the comparison is between a profile and a system.
-u	Indicates to rollback the parameter values of the last applied profile, as they were prior to issuing the last **artexset** command. To do this, the command applies the values stored in the `/etc/security/artex/latest_rollback.xml` file. If the **-u** option is specified, no profile parameter is required.
-L	Instructs each individual AIX Runtime Expert client to download the profile from an LDAP repository designated by the profile string.
-D	Indicates to output the results of the remote **artexset** command associated with each individual node.

Parameters

profile	This is a mandatory file, except when the **-x** or **-u** option is specified. The file specified includes a list of the tunable parameters.

clientname | nim_mac_group The name of the client node or pre-defined NIM machine groups.

Exit Status

Individual error messages from the resulting NIM commands are masked unless the **-D** option is used. A cumulative return value that consists of an “OR” of all the individual nodes or groups is retuned by the **artexremset** command.

0	The command completed successfully
>0	An error occurred.

Examples

The following example illustrates how to execute the **artexset** command on a client machine, using a profile located on NIM master.

```
artexremset nim_profile.xml client1
```

The following example illustrates how to execute the **artexset** command on multiple client machines, using a profile located on LDAP server.

```
artexremset –L ldap://profile1.xml client1 mac_group1 client2
```

The following example illustrates how to output the results of the remote **artexset** command associated with each individual client machine.

```
artexremset –D profile1.xml client1 client2
```

---

artexset Command

Purpose

The **artexset** command applies an AIX Runtime Expert profile to a system. The profile contains values for parameters that are to be set on the system.

Syntax

artexset [[[[[**-q**] [**-c**] [**-r**] [**-R**]] |**-t** | -p]
[[**-l** {*dynamic* | *noreboot* | *reboot* | *all* }]] | **-b** | **-x** | **-u**] *profile*]

artexset [**-q**] [**-c**] [**-r**] [**-R**]
[**-l** {*dynamic* | *noreboot* | *reboot* | *all* }] *profile*

artexset [**-l** {*dynamic* | *noreboot* | *reboot* | *all* }] **-t** *profile*

artexset [**-l** {*dynamic* | *noreboot* | *reboot* | *all* }] **-p** *profile*

artexset -b *profile*

artexset -x

artexset -u

Description

The **artexset** command applies an AIX Runtime Expert profile to a system. The profile contains values for parameters that are to be set on the system. This command also allows you to verify the accuracy of setting the parameters for a profile, preview the parameters that the command changes, enable and disable the ability to set the profile parameters during boot time, and rollback to a previous profile.

When the **–t** option is specified, the command tests the correctness of the profile. The command checks whether the profile has the correct XML format. Also, it checks whether the parameters defined in the profile are valid and supported by AIX Runtime Expert.

When the **–p** option is specified, the parameters for the profile are not set but rather the parameters that would change are identified. Only the parameter values that would change are listed in the output. For example, if the parameter value on the system is same as the parameter value in the profile, the parameter would not be listed in the output since the parameter value is not affected by the command.

By default, this command creates a rollback profile. The rollback profile allows you to undo a profile change if needed. If the **–R** option is specified, the command does not create a rollback profile.

If you want to rollback to a previous state, use the **-u** option. One level of rollback is supported. For example, after a rollback is complete, you cannot perform another subsequent rollback until **artexset** is run again to set the parameters.

When **–b** option specified, the parameters are set during each system boot. This option can be disabled by using the **-x** option.

With the **-l** option, you can set a subset of the parameters that are noted in the profile. If the **-l** option is not specified, all parameters listed in the profile are applied only if none of the parameters require a reboot. If dynamic selection criteria (`-l dynamic`) is specified, all parameters that do not require a reboot, disruptive action, like stopping and restarting a service, or unmounting and mounting a file system are set. If `noreboot` selection criteria (`-l noreboot`) is specified, all parameters that do not need a reboot are set. If the selection criteria reboot (`-l reboot`) is specified, all parameters that require a reboot are set. If the selection criteria all (`-l all`) is specified, then all parameters are set.

The specified profile can be on the local file system using a relative or absolute path or on an LDAP server.

Flags

-q Indicates to ignore non-fatal warning messages.

-c Indicates to verify that the command set the values and that they were successfully applied to the system. If they were not successfully applied, then the **artexset** operation is aborted.

-r Indicates to rollback if a failure occurs.

-l { *dynamic* | *noreboot* | *reboot* | *all*} Specifies the level to which to apply the parameters. The **-l** flag has the following options:

- The *dynamic* variable indicates to apply non-disruptive parameters only.
- The *noreboot* variable indicates to apply all parameters that do not need a reboot.
- The *reboot* variable indicates to apply only the parameters that have a reboot constraint.
- The *all* variable indicates to apply all parameters, including the ones that need a reboot.

-R Specifies to not create a rollback profile.

-b Indicates to enable the master profile, which is also referred to as the boot profile.

-x Indicates to disable the master profile, which is also referred to as the boot profile. This flag is the opposite of the **-b** option. If the -x option is specified, no profile parameter is required.

-t Indicates to test if the values listed in the *profile* are valid tunables, as recognized by the runtime system.

-p Specifies to preview setting the parameters but does not set the parameters for the *profile*. This flag identifies which parameters would change as a result of issuing this command. The output lists what parameters would change, what services would restart, and whether the system would need to restart, if the profile is applied. Only the parameter values that would change are listed in the output. For example, if the parameter value on the system is same as the parameter value in the profile, the parameter would not be listed in the output since the parameter value is not affected by the command.

-u Indicates to rollback the parameter values of the last applied profile, as they were prior to issuing the last **artexset** command. To do this, the command applies the values stored in the `/etc/security/artex/latest_rollback.xml` profile. If the **-u** option is specified, no profile parameter is required.

Parameters

profile This is a mandatory file, except when the **-x** or **-u** option is specified. The file specified includes a list of the tunable parameters.

Exit Status

0 The command completed successfully
>0 An error occurred.

Examples

The following example illustrates how to set all parameters defined in the profile `local_profile.xml`.

```
artexset -l all local_profile.xml
```

The following example illustrates how to check the correctness of the `ldap_profile.xml` profile stored on an LDAP server.

```
artexset -t ldap://ldap_profile.xml
```

The following example illustrates how to enable applying the profile `/tmp/boot_profile.xml` at every system restart.

```
artexset -b /tmp/boot_profile.xml
```

The following example illustrates how to disable applying a profile at every system restart.

```
artexset -x
```

The following example illustrates how to rollback the parameters to the values prior to previous issue of the **artexset** command.

```
artexset -u
```

as Command

Purpose

Reads and assembles a source file.

Syntax

as [**-a** *Mode*] [**-o** *ObjectFile*] [**-n** *Name*] [**-u**] [**-l** [*ListFile*]] [**-W** | **-w**] [**-x** [*XCrossFile*]] [**-s** [*ListFile*]] [**-m** *ModeName*] [**-E***offlon*] [**-p***offlon*] [**-i**] [**-v**] [*File*]

Description

The **as** command reads and assembles the named *File* (by convention, this file ends with a **.s** suffix). If you do not specify a *File*, the **as** command reads and assembles standard input. It stores its output, by default, in a file named **a.out**. The output is stored in the XCOFF file format.

All flags for the **as** command are optional.

Flags

-a *Mode*
Specifies the mode in which the **as** command operates. By default, the **as** command operates in 32-bit mode, but the mode can be explicitly set by using the flag **-a32** for 32-bit mode operation or **-a64** for 64-bit mode operation.

-l[*ListFile*]
Produces an assembler listing. If you do not specify a file name, a default name is produced by replacing the suffix extension of the source file name with a **.lst** extension. By convention, the source file suffix is a **.s**. For example:

```
sourcefile.xyz
```

produces a default name of:

```
sourcefile.lst
```

If the source code is from standard input and the **-l** flag is used without specifying an assembler-listing file name, the listing file name is **a.lst**.

-m *ModeName*
Indicates the assembly mode. This flag has lower priority than the **.machine** pseudo-op.

If this flag is not used and no **.machine** pseudo-op is present in the source program, the default assembly mode is used. The default assembly mode has the POWER family/PowerPC intersection as the target environment, but treats all POWER family/PowerPC incompatibility errors (including instructions outside the POWER family/PowerPC intersection and invalid form errors) as instructional warnings.

If an assembly mode that is not valid is specified and no **.machine** pseudo-op is present in the source program, an error is reported and the default assembly mode is used for instruction validation in pass 1 of the assembler.

If the **-m** flag is used, the *ModeName* variable can specify one of the following values:

""
Explicitly specifies the default assembly mode that has the POWER family/PowerPC intersection as the target environment, but treats instructions outside the POWER family/PowerPC intersection and invalid form errors as instructional warnings. A space is required between **-m** and the null string argument (two double quotation marks).

com
Specifies the POWER family/PowerPC intersection mode. A source program can contain only instructions that are common to both POWER family and PowerPC; any other instruction causes an error. Any instruction with an invalid form causes errors, terminates the assembly process, and results in no object code being generated.
Note: Certain POWER family instructions are supported by the PowerPC 601 RISC Microprocessor in AIX 5.1 and earlier, but do not conform to the PowerPC architecture. These instructions cause errors when using the **com** assembly mode.

any
Specifies the indiscriminate mode. The assembler generates object code for any recognized instruction, regardless of architecture. This mode is used primarily for operating system development and for testing and debugging purposes.
Note: All POWER family and PowerPC incompatibility errors are ignored when using the **any** assembly mode, and no warnings are generated.

ppc Specifies the PowerPC mode. A source program can contain only PowerPC instructions. Any other instruction causes an error.

Notes:

1. The PowerPC optional instructions are implemented in each PowerPC processor and do not belong to the **ppc** mode. These instructions generate an error if they appear in a source program that is assembled using the **ppc** assembly mode.
2. Certain instructions conform to the PowerPC architecture, but are not supported by the PowerPC 601 RISC Microprocessor in AIX 5.1 and earlier.

ppc64 Specifies the POWER-based 64-bit mode. A source program can contain 64-bit POWER-based instructions.

pwr Specifies the POWER® mode. A source program can contain only instructions for the POWER implementation of the POWER architecture.

pwr2(pwrx)
Specifies the POWER2 mode through AIX 5.1 only. A source program can contain only instructions for the POWER2 implementation of the POWER architecture in AIX 5.1 and earlier. **pwr2** is the preferred value. The alternate assembly mode value **pwrx** means the same thing as **pwr2**.
Note: The POWER implementation instruction set is a subset of the POWER2 implementation instruction set for AIX 5.1 and earlier.

pwr5 Specifies the POWER5 mode. A source program can contain only instructions for the POWER5 implementation of the POWER architecture.

601 Specifies the PowerPC 601 RISC Microprocessor mode through AIX 5.1 only. A source program in AIX 5.1 and earlier can contain only instructions for the PowerPC 601 RISC Microprocessor.

The PowerPC 601 RISC Microprocessor design was completed before the POWER-based platform for AIX 5.1 and earlier. Therefore, some PowerPC instructions may not be supported by the PowerPC 601 RISC Microprocessor in AIX 5.1 and earlier.

Attention: It is recommended that the **601** assembly mode not be used for applications that are intended to be portable to future PowerPC systems. The **com** or **ppc** assembly mode should be used for such applications.

The PowerPC 601 RISC Microprocessor in AIX 5.1 and earlier implements the POWER-based platform plus some POWER family instructions are not included in the POWER-based platform. This allows existing POWER applications to run with acceptable performance on POWER-based systems.

603 Specifies the PowerPC 603 RISC Microprocessor mode through AIX 5.1 only. A source program can contain only instructions for the PowerPC 603 RISC Microprocessor in AIX 5.1 and earlier.

604 Specifies the PowerPC 604 RISC Microprocessormode. A source program can contain only instructions for the PowerPC 604 RISC Microprocessor.

970 Specifies the PowerPC 970 mode. A source program can contain instructions valid for PowerPC 970 compatible processors.

A35 Specifies the A35 mode. A source program can contain only instructions for the A35.

pwr5x Specifies the POWER5+ mode. A source program can contain instructions valid for POWER5+ compatible processors.

pwr6 Specifies the POWER6 mode. A source program can contain instructions valid for POWER6 compatible processors.

-n *Name*
Specifies the name that appears in the header of the assembler listing. By default, the header contains the name of the assembler source file.

-o *ObjectFile*
Writes the output of the assembly process to the specified file instead of to the **a.out** file.

-s[*ListFile*]
Indicates whether or not a mnemonics cross-reference for POWER family and PowerPC is included in the assembler listing. If this flag is omitted, no mnemonics cross-reference is produced. If this flag is used, the assembler listing will have POWER family mnemonics if the source contains PowerPC mnemonics, and will have PowerPC mnemonics if the source contains POWER family mnemonics.

The mnemonics cross-reference is restricted to instructions that have different mnemonics in the POWER family and PowerPC, but that have the same op code, function, and input operand format.

Because the **-s** flag is used to change the assembler-listing format, it implies the **-l** flag. If both option flags are used and different assembler-listing file names (specified by the *ListFile* variable) are given, the listing file name specified by the *ListFile* variable used with the **-l** flag is used. If an assembler-listing file name is not specified with either the **-l** or **-s** flag, a default assembler listing file name is produced by replacing the suffix extension of the source file name with a **.lst** extension.

-u
Accepts an undefined symbol as an extern so that an error message is not displayed. Otherwise, undefined symbols are flagged with error messages.

-W
Turns off all warning message reporting, including the instructional warning messages (the POWER family and PowerPC incompatibility warnings).

-w
Turns on warning message reporting, including reporting of instructional warning messages (the POWER family and PowerPC incompatibility warnings).
Note: When neither **-W** nor **-w** is specified, the instructional warnings are reported, but other warnings are suppressed.

-x[*XCrossFile*]
Produces cross reference output. If you do not specify a file name, a default name is produced by replacing the suffix extension of the source file name with an **.xref** extension. Conventionally, the suffix is a **.s**. For example:

```
sourcefile.xyz
```

produces a default name of:

```
sourcefile.xref
```

Note: The assembler does not generate an object file when the **-x** flag is used.

-E
Specifies whether to report errors due to the new v2.00 syntax (`-Eon`), or to ignore them (`-Eoff`). By default, v2.00 errors are ignored.

-p
Specifies whether to use new v2.00 branch prediction (`-pon`), or pre-v2.00 branch prediction (`-poff`). By default, pre-v2.00 branch prediction is used.

-i
Specifies that branch prediction suffixes are to be encoded. By default, this option is not set. This option will be ignored if the **-p** option is specified.

-v
Displays the version number of this command.

File
Specifies the source file. If no file is specified, the source code is taken from standard input.

Environment Variables

OBJECT_MODE

The assembler respects the setting of the OBJECT_MODE environment variable. If neither **-a32** or **-a64** is used, the environment is examined for this variable. If the value of the variable is anything other than the values listed in the following table, an error message is generated and the assembler exits with a non-zero return code. The implied behavior corresponding to the valid settings are as follows:

OBJECT_MODE = **32**
Produce 32-bit object code. The default machine setting is **com**.

OBJECT_MODE = **64**	Produce 64-bit object code (XCOFF64 files). The default machine setting is **ppc64**.
OBJECT_MODE = **32_64**	Invalid.
OBJECT_MODE = *anything else*	Invalid.

Examples

1. To produce a listing file named file.lst and an object file named **file.o**, enter:
   ```
   as -l -o file.o file.s
   ```
2. To produce an object file named **file.o** that will run on the 601 processor for AIX 5.1 and earlier and generate a cross reference for POWER family and PowerPC mnemonics in an assembler listing file named **file.lst**, enter:
   ```
   as  -s  -m  601  -o  file.o  file.s
   ```
3. To produce an object file named **file.o** using the default assembly mode and an assembler listing file named **xxx.lst** with no mnemonics cross reference, enter:
   ```
   as  -lxxx.lst  -o  file.o  file.s
   ```

Files

/usr/ccs/bin/as	Contains the **as** command
a.out	The default output file.

Related Information

The **ld** command, **m4** command.

The Assembler Language Reference manual.

---

asa or fpr Command

Purpose

Prints FORTRAN files to in line-printer conventions.

Syntax

{ **asa** | **fpr** } [*File ...*]

Description

The **asa** and **fpr** commands print FORTRAN files to conform to this operating systems line-printer conventions. Both commands work like a filter to transform files formatted according to FORTRAN carriage control conventions into files formatted according to line-printer conventions.

The *File* variable specifies the name of the input file that the **asa** and **fpr** commands read instead of the standard input. The **asa** and **fpr** commands read the file, replace the carriage control characters with recognizable operating system characters, and print the file to standard output.

Both commands read the first character of each line from the input file, interpret the character, and space the line according to the definition of the first character. If the first character is either a **Blank**, a **0**, a dash (**-**) , a **1**, or a plus sign (**+**), either command does the following:

Blank	Advances the carriage one line and prints the input line.
0	Advances the carriage two lines and prints the input line.

- Advances the carriage three lines and prints the input line.
1 Advances the carriage to the top of the next page.
+ Does not advance the carriage and starts printing the input line in the first space of the output file.

The commands interpret a blank line as if its first character is a blank and delete a blank that appears as a carriage control character. It treats lines that begin with characters other than the defined control characters as if they begin with a blank character. The first character of a line is not printed. If any such lines appear, an appropriate diagnostic appears in the standard error.

Note: Results are undefined for input lines longer than 170 characters.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

1. Use the **fpr** command in the following manner to change the carriage control characters in an `a.out` file produced by a FORTRAN compiler into carriage control characters and print the resulting file:
   ```
   a.out | fpr | qprt
   ```
2. Use the **asa** command in the following manner to run the `f77.output` file through the **asa** command to change carriage control characters from FORTRAN to the operating system and print the resulting file.
   ```
   asa f77.output | qprt
   ```

Files

/usr/ucb/fpr Contains the **fpr** command.
/usr/bin/asa Contains the **asa** command.

Related Information

The **fsplit** command, **qprt** command, **struct** command.

---

at Command

Purpose

Runs commands at a later time.

Syntax

To Schedule Jobs to Run at a Later Time

at [**-c** | **-k** | **-s** | **-q** *Queue*] [**-m**] [**-f** *File*] { **-t** *Date* |*Time* [*Day*] [*Increment*] }

To Report Scheduled Jobs

at -l [**-v**] [**-o**] [*Job ...* | **-q** *Queue*]

at -n [*User*]

To Remove Scheduled Jobs

at -r [**-F**] [**-i**] *Job* ...

at -r [**-F**] [**-i**] **-u** *User*

Description

The **at** command reads from standard input the names of commands to be run at a later time and allows you to specify when the commands should be run.

The **at** command mails you all output from standard output and standard error for the scheduled commands, unless you redirect that output. It also writes the job number and the scheduled time to standard error.

When the **at** command is executed, it retains the current process environment. It does not retain open file descriptors, traps, and priority.

The **/var/adm/cron/at.allow** and **/var/adm/cron/at.deny** files control what users can use the **at** command. A person with root user authority can create, edit, or delete these files. Entries in these files are user login names with one name to a line. The following is an example of an **at.allow** file:

```
root
nick
dee
sarah
```

If the **at.allow** file exists, only users whose login names appear in it can use the **at** command. A system administrator can explicitly stop a user from using the **at** command by listing the user's login name in the **at.deny** file. If only the **at.deny** file exists, any user whose name does not appear in the file can use the **at** command.

A user cannot use the **at** command if one of the following is true:

- The **at.allow** file and the **at.deny** file do not exist (allows root user only).
- The **at.allow** file exists but the user's login name is not listed in it.
- The **at.deny** file exists and the user's login name is listed in it.

If the **at.allow** file does not exist and the **at.deny** file does not exist, only users with root authority can submit a job with the **at** command.

To schedule a job to run at a later time, you must specify a time to start the job. You may specify the time by using either the **-t** *Date* flag or the *Time*, *Day*, and *Increment* parameters. At most, 60 jobs can be scheduled in any given run queue at the granularity of one per second. If more jobs than can be handled are submitted, for every job beyond 60, a file **_at<pid>** is created in **/var/spool/cron/atjobs/** which can be safely deleted by the end user.

The *Date* variable to the **-t** flag is specified using the following format:

[[*CC*]*YY*]*MMDDhhmm*[**.***SS*]

The digits in the *Date* variable are defined as follows:

CC	Specifies the first two digits of the year (the century).
YY	Specifies the second two digits of the year.
MM	Specifies the month of the year (01 through 12).
DD	Specifies the day of the month (01 through 31).
hh	Specifies the hour of the day (00 through 23).
mm	Specifies the minute of the hour (00 through 59).

SS Specifies the second of the minute (00 through 59).

Both the *CC* and *YY* digits are optional. If neither is given, the current year is assumed. If the *YY* digits are specified but the *CC* digits are not, the *CC* digits are defined as follows:

- If the value of the *YY* digits is between 70 and 99, the value of the *CC* digits is assumed to be 19.
- If the value of the *YY* digits is between 00 and 37, the value of the *CC* digits is assumed to be 20.
- The default value of *SS* is 00.

The resulting time is affected by the value of the **TZ** environment variable.

The *Time* parameter may be specified as a number followed by an optional suffix. The **at** command interprets one- and two-digit numbers as hours. It interprets four digits as hours and minutes. The **T_FMT** item in the **LC_TIME** locale category specifies the order of hours and minutes. The default order is the hour followed by the minute. You can also separate hours and minutes with a : (colon). The default order is *Hour*:*Minute*.

In addition, you may specify one of the following suffixes:

- **am**
- **pm**
- **zulu**

If you do not specify **am** or **pm**, the **at** command uses a 24-hour clock. These suffixes can follow the time as a separate argument or separated with spaces. The **am** and **pm** suffixes are defined values from the **AM_STR** and **PM_STR** items in the **LC_TIME** locale category. The suffix **zulu** indicates that the time is **GMT** (Greenwich Mean Time).

The **at** command also recognizes the following keywords as special values for the *Time* parameter:

- **noon**
- **midnight**
- **now**
- **A** for AM
- **P** for PM
- **N** for noon
- **M** for midnight

You may specify the optional *Day* parameter as either a month name and a day number (and possibly a year number preceded by a comma), or a day of the week. The **D_FMT** item in the **LC_TIME** locale category specifies the order of the month and day (by default, month followed by day). The **DAY_1** through **DAY_7** items in the **LC_TIME** locale category specify long day names. The **ABDAY_1** through **ABDAY_7** items in the **LC_TIME** locale category specify short day names. The **MON_1** through **MON_12** items in the **LC_TIME** locale category specify long month names. The **ABMON_1** through **ABMON_12** items in the **LC_TIME** locale category specify short month names. By default, the long name is fully spelled out; the short name is abbreviated to two or more characters for weekdays, and three characters for months.

The **at** command recognizes **today** and **tomorrow** as special default values for the *Day* parameter. The **today** value is the default *Day* if the specified time is later than the current hour; the **tomorrow** value is the default if the time is earlier than the current hour. If the specified month is less than the current month (and a year is not given), next year is the default year.

Flags

-c Requests that the **csh** command be used for executing this job.

-f *File* Uses the specified file as input rather than using standard input.

-F Suppresses delete verification. Use this flag with the **-r** flag.

-i Specifies interactive delete. Use this flag with the **-r** flag.

-k Requests that the **ksh** command be used for executing this job.

-l Reports your scheduled jobs. If you have root user authority, you can get jobs issued by other users.

-m Mails a message to the user about the successful execution of the command.

-n [*User*] Reports the number of files in your queue or user's queue.

-o Lists jobs in schedule order. This flag is useful only with the **-l** flag.

-q *Queue* Specifies the queue in which to schedule a job for submission. When used with the **-l** flag, the report is limited to the queue specified by the *Queue* variable. By default, **at** jobs are scheduled in the **a** queue. The **b, c** and **d** queues are reserved for **batch** jobs, **cron** jobs, and **sync** jobs respectively.

-q a Queues **at** jobs.

-q b Queues **batch** jobs. The **batch** command calls the **at** command with this flag.

Note: When using the **b** queue, commands are read from standard input. Also, the **now** keyword is used for the *Time* parameter, regardless of what you specify on the command line.

-q e Queues **ksh** jobs. Equivalent to the **-k** flag.

-q f Queues **csh** jobs. Equivalent to the **-c** flag.

-q g-z Queues user defined queue jobs.

-r *Job...* Removes *Job*s previously scheduled by the **at** or **batch** commands, where *Job* is the number assigned by the **at** or **batch** commands. If you do not have root user authority (see the **su** command), you can remove only your own jobs. The **atrm** command is available to the root user to remove jobs issued by other users or all jobs issued by a specific user.

-s Requests that the **bsh** command (Bourne shell) be used for executing this job.

-t *Date* Submits the job to be run at the time specified by the *Date* variable.

-u *User* Deletes all jobs for the specified user. If used with the **-r** flag, do not specify a *Job* variable (the correct syntax is **at -r -u** *User*).

-v Used with **-l** flag to show content of listed jobs.

Parameters

Day Specifies the optional *Day* parameter as either a month name and a day number (and possibly a year number preceded by a comma), or a day of the week.

Increment The optional *Increment* parameter can be one of the following:

- A **+** (plus sign) followed by a number and one of the following words:
  - **minute**[**s**]
  - **hour**[**s**]
  - **day**[**s**]
  - **week**[**s**]
  - **month**[**s**]
  - **year**[**s**]
- The special word **next** followed by a one of the following words:
  - **minute**[**s**]
  - **hour**[**s**]
  - **day**[**s**]

- **week**[**s**]
- **month**[**s**]
- **year**[**s**]

Security

Auditing Events

If the auditing subsystem is properly configured and is enabled, the **at** command generates the following audit record or event every time the command is run:

Event	Information
AT_JobAdd	Lists **at** jobs that were run, the time the task was completed, and the user who issued the command.

For more details about how to properly select and group audit events, and how to configure audit event data collection, see Setting Up Auditing in *Security*.

RBAC Disclaimer

Exit Status

This command returns the following exit values:

0 The **at** command successfully submitted, removed, or listed a job or jobs.
>0 An error occurred.

Examples

1. To schedule the command from the terminal, enter a command similar to one of the following:

   If uuclean is in your current directory, enter:

   ```
   at   5   pm   Friday
   uuclean
   <Ctrl-D>

   at   now   next   week
   uuclean
   <Ctrl-D>
   ```

   If uuclean is in **$HOME/bin/uuclean**, enter:

   ```
   at   now   +   2   days
   $HOME/bin/uuclean
   <Ctrl-D>
   ```

   Note: When entering a command name as the last item on the command line, a full path name must be given if the command is not in the current directory, and the **at** command will not accept any arguments.

2. To run the **uuclean** command at 3:00 in the afternoon on the 24th of January, enter any one of the following commands:

   ```
   echo   uuclean   |   at   3:00   pm   January   24

   echo   uuclean   |   at   3 pm   Jan   24

   echo   uuclean   |   at   1500   jan   24
   ```

3. To have a job reschedule itself, invoke the **at** command from within the shell procedure by including code similar to the following within the shell file:

   ```
   echo   "ksh   shellfile"   |   at   now   tomorrow
   ```

4. To list the jobs you have sent to be run later, enter:

   ```
   at    -l
   ```

5. To cancel a job, enter:

   ```
   at    -r   ctw.635677200.a
   ```

   This cancels job `ctw.635677200.a`. Use the **at -l** command to list the job numbers assigned to your jobs.

Files

/var/adm/cron/FIFO	A named pipe that sends messages to the **cron** daemon when new jobs are submitted with the **crontab** or **at** commands.
/usr/bin/at	Contains the **at** command.
/var/adm/cron	Contains the main **cron** directory.
/var/adm/cron/at.allow	Specifies the list of allowed users.
/var/adm/cron/at.deny	Specifies the list of denied users.
/var/spool/cron/atjobs	Contains the spool area directory for **at**.

Related Information

The **atq** command, **atrm** command, **auditpr** command, **batch** command, **bsh** command, **kill** command, **ksh** command, **mail** command, **nice** command, **ps** command, **sh** command, **su** command.

The **cron** daemon.

The **environment** file.

Auditing Overview in *Security* provides more information about audits and audit events.

Input and output redirection overview in *Operating system and device management* describes how the operating system processes input and output.

National Language Support Overview for Programming in *Operating system and device management* explains collating sequences, equivalence classes, and locale.

Shells in *Operating system and device management* describes what shells are, the different types of shells, and how shells affect the way commands are interpreted.

---

ate Command

Purpose

Starts the Asynchronous Terminal Emulation (ATE) program.

Syntax

ate

Description

The **ate** command starts the Asynchronous Terminal Emulation (ATE) program. The ATE program establishes a connection between a workstation and a remote computer. A workstation acts as a terminal connected to the remote computer. Using ATE the user can connect to, and exchange data with, remote databases and other systems.

> **Note:** Users must be a member of the UNIX-to-UNIX Copy Program (uucp) group in order to use ATE. A user with root authority uses System Management Interface Tool (SMIT) to install individual users in groups.

ATE establishes the connection and allows users to record and control the session. After logging in to the remote system, users execute programs, issue commands, and use files on the remote system as a local user. ATE also enables a workstation to emulate a VT100 terminal.

The ATE program uses menus and subcommands. From the menus, users issue subcommands to connect to a remote system, receive and transfer files, and execute commands. The Unconnected Main Menu displays any time users issue the **ate** command. The Connected Main Menu displays when users press the MAINMENU_KEY (usually the Ctrl-V key sequence) while connected to another system. The **connect** subcommand makes the connection.

The ATE program supports three control key sequences: the CAPTURE_KEY (usually Ctrl-B), PREVIOUS_KEY (usually CTRL-R), and MAINMENU_KEY (usually CTRL-V). These control keys do not function until the ATE program is started. The control keys and other ATE defaults can be changed by editing the **ate.def** file format.

Examples

To start the ATE program, enter:

```
ate
```

The ATE Unconnected Main Menu displays.

Subcommands

alter	Temporarily changes data transmission characteristics in the ATE program.
break	Interrupts current activity on a remote system.
connect	Connects to a remote computer.
directory	Displays the ATE dialing directory.
help	Provides help information for the ATE subcommands.
modify	Temporarily modifies local settings used for terminal emulation.
perform	Allows the user to issue workstation operating system commands while using ATE.
quit	Exits the Asynchronous Terminal Emulation (ATE) program.
receive	Receives a file from a remote system.
send	Sends a file to a remote system.
terminate	Terminates an ATE connection to a remote system.

alter Subcommand

a [**l** *CharacterLength*] [**s** *StopBit*] [**p** *Parity*] [**r** *BaudRate*] [**d** *Device*] [**i** *DialPrefix*] [**f** *DialSuffix*] [**w** *Seconds*] [**a** *RedialAttempts*] [**t** *TransferProtocol*] [**c** *PacingType*]

> **Note:** The default values of the **alter** subcommand flags can be permanently changed by editing the **ate.def** file format.

The **alter** subcommand is accessed from the Asynchronous Terminal Emulation (ATE) Connected or Unconnected Main Menu. Issuing the **ate** command from the command line displays the Unconnected Main Menu. The **alter** subcommand temporarily changes these data transmission characteristics:

- Data character length
- Baud rate
- Stop and parity bits
- Port name
- Modem dialing prefixes and suffixes
- Waiting time and retry limits
- File transfer protocol
- Pacing character or delay time

The settings return to the defaults as defined in the **ate.def** file format when the user exits ATE.

When issued without flags from either of the ATE main menus, the **alter** subcommand displays the Alter Menu. To bypass the Alter Menu, enter the **alter** subcommand, followed by the appropriate flags, at the command prompt on either ATE main menu.

The **alter** subcommand can change more than one feature at a time. To change the value of more than one variable, type the first flag followed by the new value, followed by a space, then the second flag and second value, and so on.

To permanently change the settings affected by the **alter** subcommand, customize the **ate.def** file format.

The Alter Menu: The Alter Menu displays the current settings of the changeable characteristics with the **alter** subcommand. Enter the letter **a** after the command prompt on either the ATE Connected or Unconnected Main Menu to view the Alter Menu.

The Alter Menu contains the following columns:

Column Names	**Contents**
COMMAND	Flag that changes the value of a variable
DESCRIPTION	Description of the variable that the flag affects
CURRENT	Current value of the variable
POSSIBLE CHOICES	Possible values of the variable

To change the value of a variable, enter the flag (from the COMMAND column) and new value (from the POSSIBLE CHOICES column) at the command prompt on the Alter Menu.

To return to one of the ATE main menus from the Alter Menu, press the Enter key.

Flags:

a *RedialAttempts*
: Specifies the maximum number of times the ATE program redials for a connection. If the *RedialAttempts* variable is 0, no redial attempt occurs.

  Options: 0 (none) or a positive integer

  Default: 0

c *PacingType*

Specifies the type of **pacing** protocol used.

Default: 0 (no pacing)

> **Note:** The *PacingType* variable has no effect when the **xmodem** protocol is used.

The *PacingType* can be either of the following:

Character
: Signal to transmit a line. The signal can be any ASCII character.

  When the **send** subcommand encounters a line-feed character while transmitting data, it waits to receive the pacing character before sending the next line.

  When the **receive** subcommand is ready to receive data, it sends the pacing character and then waits 30 seconds to receive data. The **receive** subcommand sends a pacing character again whenever it finds a carriage-return character in the data. The **receive** subcommand ends when it receives no data for 30 seconds.

Interval
: Number of seconds the system waits between each line it transmits. The value of the *Interval* variable must be an integer. The default value is 0 indicating a pacing delay of 0 seconds.

d *Device*

Specifies the name of the asynchronous port used to connect to a remote system.

Options: Locally created port names. The first 8 characters of the port name display in the Alter Menu.

Default: tty0

f *DialSuffix*

Specifies the dial suffix that must follow the telephone number when autodialed with a modem. Consult the modem documentation for the proper dial command.

Options: 0 (none) or a valid modem suffix. The first 8 characters display in the Alter Menu.

Default: no default

i *DialPrefix*

Specifies the dial prefix that must precede the telephone number when autodialed with a modem. Consult the modem documentation for the proper dial commands.

Options: ATDT, ATDP, or other values depending on the type of modem used. The first 8 characters display in the Alter Menu.

Default: ATDT

l *CharacterLength*

Specifies the number of bits in a data character. This length must match the length expected by the remote system.

Options: 7 or 8

Default: 8

p *Parity*

Checks whether a character was successfully transmitted to or from a remote system. Must match the parity of the remote system.

For example, if the user selects even parity, when the number of 1 bits in the character is odd, the parity bit is turned on to make an even number of 1 bits.

Options: 0 (none), 1 (odd), or 2 (even)

Default: 0

r *BaudRate*

Specifies the baud rate, or bits transmitted per second (bps). The speed must match the speed of the modem and that of the remote system.

Options: 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, or 19200

Default: 1200

s *StopBit*

Specifies the number of stop bits appended to a character to signal the end of that character during data transmission. This number must match the number of stop bits used by the remote system.

Options: 1 or 2

Default: 1

t *TransferProtocol*

Defines the type of asynchronous protocol that transfers files during a connection.

p File transfer protocol controls the data transmission rate by waiting for either a specified character or a certain number of seconds between line transmissions. This helps prevent loss of data when the transmission blocks are either too large or sent too quickly for the system to process.

x An 8-bit file transfer protocol to detect data transmission errors and retransmit the data.

Options: **p** (**pacing)**, or **x** (**xmodem**)

Default: **p**

w *Seconds*

wait

Specifies the number of seconds between redial attempts. The wait period does not begin until the connection attempt times out or until it is interrupted. If the **attempts** flag is set to 0, no redial attempt occurs.

Options: 0 (none) or a positive integer

Default: 0

Examples:

1. To display the Alter Menu, enter the **alter** subcommand at the command prompt on either ATE main menu:

   ```
   a
   ```

   The Alter Menu is displayed.
2. To alter transmission settings from the Alter Menu, enter the appropriate flags at the command prompt on the Alter Menu:

- To change the value for the **rate** flag, enter:

  ```
  r 9600
  ```

For the current session of ATE, the baud rate is changed to `9600` bps.

- To change the value of the **wait** flag, enter:

  ```
  w 7
  ```

For the current session of ATE, the wait time for redial changes to `7` seconds.

- To bypass the Alter Menu when using the **alter** command, type the command abbreviation **a**, followed by the appropriate flags, at the prompt on one of the ATE main menus. For example, to change the **rate**, **wait**, and **attempt** values, enter the following at the prompt on either ATE main menu:

  ```
  a r 9600 w 5 a 1
  ```

For the current session of ATE, the baud rate changes to 9600 bps, the wait time for redial changes to 5 seconds, and the maximum number of redial attempts changes to 1 attempt.

break Subcommand

b

The **break** subcommand sends a break signal to the remote system connected to the terminal by the Asynchronous Terminal Emulation (ATE) program. The **break** subcommand interrupts current activity on the remote system. Issue the **break** subcommand from the ATE Connected Main Menu.

Attention: The **break** subcommand may disconnect the current session. The system may lose data.

Example: To interrupt the current session, at the remote system login screen, press the MAINMENU_KEY (usually the Ctrl-V key sequence). When the ATE Connected Main Menu displays, enter:

```
b
```

A break signal is sent to the remote system, and the ATE Unconnected Main Menu displays. Now exit the ATE program or issue other ATE subcommands.

connect Subcommand

c [*TelephoneNumber* | *PortName*]

The ATE **connect** subcommand enables users to connect to a remote computer using Asynchronous Terminal Emulation (ATE). Issue the **connect** subcommand from the ATE Unconnected Main Menu. The connection can be made between two machines connected by cable or by telephone line. Users establish connection in one of three ways:

direct	Uses an established cabled link to another system.
manually dialed	Uses a telephone number dialed by the user.
automatically dialed	Uses a modem to dial a specified telephone number (a modem-dialed connection).

If the system login is not disabled, attempts to connect to another computer return an error. To disable the workstation port that handles system login by remote users, a user with root authority must use the **pdisable** command. Once the workstation port is secure from remote logins, the user must then ensure the remote system is ready to receive calls.

No connection is established if the line is busy, if the party does not answer, or if the user specified an unrecognized number. If any of these conditions exist, a message is displayed.

If a busy signal is received while trying to connect to a remote workstation, press the PREVIOUS_KEY (usually the Ctrl-R key sequence), and enter the *TelephoneNumber* parameter again.

Once the connection is established, ATE displays a message indicating the name of the port used for the connection.

Parameters:

PortName	Specifies the name of the port used for a direct connection.
TelephoneNumber	Specifies the telephone number used to establish a modem connection.

Examples:

1. To establish a direct connection, at the command line of the ATE Unconnected Main Menu, enter:

   ```
   c tty0
   ```

This command establishes a direct connection using port `tty0`. After connection is established, a message displays, followed by a login screen. Enter the requested login information and press the MAINMENU_KEY (usually the Ctrl-V key sequence) to display the ATE Connected Main Menu.

2. To establish a manually dialed connection, at the command line of the ATE Unconnected Main Menu, enter:

   ```
   c
   ```

   The ATE program prompts the user for information necessary to establish a manually dialed connection, such as a telephone number or modem to use. After connection is established, ATE displays a message giving the port name used for the connection, followed by a login screen. Enter the requested login information and press the MAINMENU_KEY (usually the Ctrl-V key sequence) to display the ATE Connected Main Menu.

3. To establish an automatically dialed connection, at the command line of the ATE Unconnected Main Menu, enter:

   ```
   c  2229999
   ```

   This example dials the telephone number `222-9999`. After connection is established, a message displays indicating the port used for the connection, followed by a login screen. Enter the requested login information and press the MAINMENU_KEY (usually the Ctrl-V key sequence) to display the ATE Connected Main Menu.

directory Subcommand

d

The ATE **directory** subcommand displays a dialing directory. Users establish a connection to a remote computer by selecting one of the directory entries from the displayed directory. The **directory** subcommand is issued from the ATE Unconnected Main Menu. The **directory** subcommand uses the information contained in the dialing directory to establish an automatically dialed (modem-dialed) connection.

When ATE starts, it checks the current directory for an **ate.def** file format. If an **ate.def** file format does not exist in the current directory, it creates one. The initial location of the dialing directory is **/usr/lib/dir**, but this value can be changed by Editing the ATE default file the **ate.def** file format. If users specify a different dialing directory in the **ate.def** file format, that directory is used.

The dialing directory contains entries for remote systems called with the ATE program in the format:

```
Name Phone Rate Length StopBit Parity Echo Linefeed
```

These fields give the name of the entry (usually the person or company whose computer the phone number reaches), the telephone number, and other information the ATE program uses to establish the connection.

See ″Dialing Directory File Format for ATE″ in *AIX 5L Version 5.3 Files Reference* for more information about dialing directory entries.

When an entry displays on the screen using the **directory** subcommand, the entry is preceded by an entry number. Select the entry to establish a connection to by entering its entry number in response to a prompt.

Example: To display a dialing directory, at the command line of the Unconnected Main Menu, enter:

```
d
```

The dialing directory specified in the **ate.def** file format displays and prompts the user for an entry number. Enter the number of the dialing directory entry to establish a connection with. ATE establishes the connection and displays a message indicating the port name used.

See Setting up an ATE dialing directory in *Networks and communication management*.

help Subcommand

h [**a**] [**b**] [**c**] [**d**] [**m**] [**p**] [**q**] [**r**] [**s**] [**t**]

The ATE **help** subcommand provides help information for the ATE subcommands. Issue the **help** subcommand from either the Unconnected or Connected Main Menu of ATE. Help information is available for all the ATE subcommands, and can be requested for several subcommands at the same time.

When issuing the **help** subcommand, ATE displays a description of each subcommand requested and instructions for using the subcommand. Help information for each subcommand displays individually, in the order requested. After reading each help message, press Enter to view the next page of help text. At the end of the help text, press Enter to return to the main menu.

Issue the **help** subcommand with the first letter of an ATE subcommand for help information. These are the names for the ATE subcommands:

Name	ATE Subcommand
a	**alter** subcommand
b	**break** subcommand
c	**connect** subcommand
d	**directory** subcommand
m	**modify** subcommand
p	**perform** subcommand
q	**quit** subcommand
r	**receive** subcommand
s	**send** subcommand
t	**terminate** subcommand

Examples:

1. To receive help information for a single subcommand, enter the following at one of the ATE main menus:

   ```
   h c
   ```

   Help information displays for the **connect** (c) subcommand. After viewing the help information, press the Enter key, and ATE displays the menu from which the **help** subcommand was issued.
2. To receive help information for multiple subcommands, enter the following at one of the ATE main menus:

   ```
   h r s
   ```

   The help information for the **receive** subcommand (r) displays first. After viewing the help information, press the Enter key. Help information for the **send** subcommand (s) displays. After viewing the help information, press the Enter key, and ATE displays the menu from which the **help** subcommand was issued.

modify Subcommand

m [**n** *CaptureFileName*] [**e**] [**l**] [**v**] [**w**] [**x**]

> **Note:** The default *CaptureFileName* and the initial settings of the other **modify** subcommand flags can be permanently changed in the **ate.def** file format.

The **modify** subcommand is accessed from the Asynchronous Terminal Emulation (ATE) Connected or Unconnected Main Menu. The **modify** subcommand temporarily changes how ATE functions on the local system in the following ways:

- Changes the name of the capture file that receives incoming data.

- Switches (toggles) the following features on or off:
  - Add a line-feed character at the end of each line of incoming data.
  - Use echo mode.
  - Emulate a DEC VT100 terminal at the console.
  - Write incoming data to a capture file as well as to the display.
  - Use an **Xon/Xoff** (transmitter on/off) signal.

The settings return to the default values as defined in the **ate.def** file format when the user exits ATE.

When issued without flags from either of the ATE main menus, the **modify** subcommand displays the Modify Menu. The Modify Menu can be bypassed by entering **m** (the **modify** subcommand abbreviation), followed by the appropriate flags, at the command prompt on either ATE main menu.

The **modify** subcommand can change more than one feature at a time. To change the **name** variable, enter the **n** flag followed by the new file name. All other variables are switches that can be turned on or off by typing the flag. Typing the flag switches, or toggles, the value.

To permanently change the settings affected by the **modify** subcommand, customize the **ate.def** file format in the directory running ATE.

Modify Menu: The Modify Menu displays the current settings of the features changeable with the **modify** subcommand. To display the Modify Menu, enter the letter **m** after the command prompt on either the ATE Connected Main Menu or the ATE Unconnected Main Menu.

The Modify Menu contains the following columns:

Column Names	Contents
COMMAND	Flag to enter to change a value
DESCRIPTION	Description of the variable the flag affects
CURRENT	Current value of the variable
POSSIBLE CHOICES	Possible values of the variable

To change the value of a flag other than the **name** flag, enter the flag (from the COMMAND column) at the command prompt on the Modify Menu. The flag value toggles to the alternate setting. To change the name of the capture file, enter the letter **n** (the **name** flag), followed by the new file name, at the prompt on the Modify Menu.

To return to the ATE Connected or Unconnected Main Menu from the Modify Menu, press the Enter key.

Flags:

e **echo**

Displays the input typed by the user.

With a remote computer that supports echoing, each character sent returns and displays on the screen. When the **echo** flag is on, each character is displayed twice: first when it is entered and again when it returns over a connection. When the **echo** flag is off, each character displays only when it returns over the connection.

Options: On or off

Default: Off

l **linefeed**

Adds a line-feed character after every carriage-return character in the incoming data stream.

Options: On or off

Default: Off

n *CaptureFileName* **name**

Specifies the file name for incoming data when the **write** flag is on, or when the CAPTURE_KEY (usually the Ctrl-B key sequence) is pressed during a connection.

Options: Any valid file name. The first 18 characters display in the Modify Menu.

Default: **capture**

v **VT100**

The local console emulates a DEC VT100 terminal so DEC VT100 codes can be used with the remote system. With the **VT100** flag off, the local console functions like a workstation.

Options: On or off

Default: Off

> **Note:** No keys on the console keyboard are remapped. In addition, some DEC VT100 codes, such as 132 columns, double-height and double-width lines, origin mode, and graphics characters generated from a 10-key keypad, are not supported.

w **write**

Routes incoming data to the capture file (specified by the **name** flag) as well as to the display. The **write** command functions like the CAPTURE_KEY key sequence during a connection. Carriage return and line-feed combinations are converted to line-feed characters before being written to the capture file. In an existing file, data is appended to the end of the file.

Options: On or off

Default: Off

x **Xon/Xoff**

Controls data transmission at a port using the **Xon/Xoff** protocol, as follows:

- When an **Xoff** signal is received, transmission stops.
- When an **Xon** signal is received, transmission resumes.
- An **Xoff** signal is sent when the receive buffer is nearly full.
- An **Xon** signal is sent when the buffer is no longer full.

Options: On or off

Default: On

> **Note:** If you use a variable value with any flag other than the **name** flag, the following error message displays:
>
> ```
> 828-003 not 'command-name' command is not valid.
> Enter the first letter of a command
> from the list on the menu.
> ```
>
> This error message indicates either an incorrect letter was entered or a value that is not valid was included.

Examples:

1. To display the Modify Menu, enter the **modify** subcommand at the command prompt on either ATE main menu:

   ```
   m
   ```

   The Modify Menu displays.
2. To modify settings from the Modify Menu, enter the appropriate flag at the command prompt at the bottom of the Modify Menu:
   - To toggle the values of the **linefeed** flag, at the prompt on the Modify Menu enter:

     ```
     l
     ```

     The value of the **linefeed** flag is switched to the alternate setting.
   - To change the **name** variable to `schedule`, at the prompt on the Modify Menu enter:

     ```
     n schedule
     ```

     Any data saved is now put into the `schedule` file.
3. To bypass the Modify menu when using the **modify** subcommand, type the **m** subcommand (the **modify** subcommand abbreviation), followed by the appropriate flags, at the command prompt on either ATE main menu:
   - To toggle the values of the **linefeed** and **echo** flags, at the prompt on either ATE main menu enter:

     ```
     m l e
     ```

     The values of the **linefeed** and **echo** flags are switched to the alternate settings. Display the Modify Menu to view the current settings of the flags.
   - To change the **name** variable to `schedule` and toggle the values of the **write** and **Xon/Xoff** flags, at the prompt on either ATE main menu enter:

     ```
     m n schedule w X
     ```

     Any data saved is now put into the `schedule` file, and the values of the **write** and **Xon/Xoff** flags are switched to the alternate settings. Display the Modify Menu to view the settings of the flags.

perform Subcommand

p [*Command*]

The ATE **perform** subcommand allows the user to issue workstation operating system commands while using Asynchronous Terminal Emulation (ATE). Issue the **perform** subcommand from the ATE Unconnected or Connected Main Menu. *Command* specifies a valid workstation operating system command.

Examples:

1. To issue a workstation operating system command, at the command line of the ATE Unconnected or Connected Main Menu, enter:

   ```
   p
   ```

   ATE prompts the user to enter a command. ATE executes the specified command. After the command finishes, ATE displays the menu from which the **perform** subcommand was issued.
2. To specify the command to be executed, at the command line of the ATE Unconnected or Connected Main Menu, enter:

   ```
   p cat mystuff
   ```

   ATE executes the **cat** command, which displays the `mystuff` file. After the **cat** command finishes, ATE displays the menu from which the **perform** subcommand was issued.

quit Subcommand

q

The ATE **quit** subcommand exits the Asynchronous Terminal Emulation (ATE) program. Issue the **quit** subcommand from the ATE Unconnected or Connected Main Menu. Issuing the **quit** subcommand ends the ATE program and displays the command prompt.

Example: To exit the ATE program, from the command line of either ATE main menu, enter:

```
q
```

The ATE program ends and the command prompt displays.

receive Subcommand

r *FileName*

The ATE **receive** subcommand enables your system to receive a file from a remote system. The ATE **receive** subcommand is issued from the ATE Connected Main Menu.

The ATE **receive** subcommand uses the **xmodem** file transfer protocol, which enables your system to receive data from a remote system, a block at a time, with error checking. The remote system must be set to send the file before your system can receive. Use the **xmodem** command with the **-s** flag on the remote system to enable the remote system to send the file. Then issue the **receive** subcommand. *FileName* names the file where the received data is stored.

Example: To receive a file sent from the remote system, at the command line of the ATE Connected Main Menu, enter:

```
r myfile
```

The data is received from the remote system and is stored in the `myfile` file.

send Subcommand

s [*FileName*]

The ATE **send** subcommand sends a file to a remote system. Issue the ATE **send** subcommand from the ATE Connected Main Menu once a connection is established. The ATE **connect** subcommand establishes the connection and prepares the remote system to receive files.

The **send** subcommand uses the **xmodem** file transfer protocol, sending data to a remote system, a block at a time, with error checking. Issue the **xmodem** command with the **-r** flag on the remote system to enable the remote system to receive the file. Then issue the **send** subcommand. *FileName* names the file to send to the remote system.

Examples:

1. To send a file to a remote system, at the command line of the ATE Connected Main Menu, enter:

   ```
   s
   ```

   ATE prompts the user for the name of the file to send to the remote system.
2. To specify a file to send to the remote system, at the command line of the ATE Connected Main Menu, enter:

   ```
   s mystuff
   ```

   The `mystuff` file is sent to the remote system.

terminate Subcommand

t

The ATE **terminate** subcommand ends an Asynchronous Terminal Emulation (ATE) connection to a remote system and returns to the ATE Unconnected Main Menu. Issue the **terminate** subcommand from the ATE Connected Main Menu.

Example: To terminate the current session, from the remote system login screen, press the MAINMENU_KEY (usually the Ctrl-V key sequence). When the ATE Connected Main Menu displays, enter:

```
t
```

A terminate signal is sent to the remote system, the session ends, and ATE displays the Unconnected Main Menu. Now issue other ATE subcommands or exit ATE.

File

/usr/lib/dir Contains the default dialing directory.

Related Information

The **ate.def** file format contains ATE default values.

ATE main menus in *Networks and communication management* describes the ATE program, its menus, and its control keys.

Editing the ATE default file in *Networks and communication management* explains how to permanently change ATE defaults.

---

atmstat Command

Purpose

Shows Asynchronous Transfer Mode adapters statistics.

Syntax

atmstat [**-d -r**] *Device_Name*

Description

The **atmstat** command displays Asynchronous Transfer Mode (ATM) adapter statistics. The user can optionally specify that the device-specific statistics be displayed in addition to the device generic statistics. If no flags are specified, only the device generic statistics are displayed. For information on statistic from the **atmstat** command, see ATM adapter statistics in the *Networks and communication management*.

If an invalid *Device_Name* is specified, the **atmstat** command produces an error message stating that it could not connect to the device.

Flags

-d Displays detailed statistics.
-r Resets all the statistics back to their initial values. This flag can only be issued by privileged users.

Parameters

Device_Name The name of the ATM device, for example, **atm0**.

Examples

To display the adapter generic statistics for **atm0**, enter:

```
atmstat atm0
```

This produces the following output on a Micro Channel machine in AIX 5.1 and earlier:

```
ATM STATISTICS (atm0) :
Device Type: Turboways 155 MCA ATM Adapter
Hardware Address: 08:00:5a:99:88:d5
Elapsed Time: 2 days 23 hours 38 minutes 18 seconds

Transmit Statistics:                          Receive Statistics:
--------------------                          -------------------
Packets: 50573                                Packets: 0
Bytes: 2225182                                Bytes: 0
Interrupts: 0                                 Interrupts: 12904
Transmit Errors: 0                            Receive Errors: 0
Packets Dropped: 0                            Packets Dropped: 0
                                              Bad Packets: 0
Max Packets on S/W Transmit Queue: 0
S/W Transmit Queue Overflow: 0
Current S/W+H/W Transmit Queue Length: 0

Cells Transmitted: 50573                      Cells Received: 0
Out of Xmit Buffers: 0                        Out of Rcv Buffers: 0
Current HW Transmit Queue Length: 0           CRC Errors: 0
Current SW Transmit Queue Length: 0           Packets Too Long: 0
                                              Incomplete Packets: 0
                                                Cells Dropped: 0

General Statistics:
-------------------
No mbuf Errors: 0
Adapter Loss of Signals: 0
Adapter Reset Count: 0
Driver Flags: Up Running Simplex
 64BitSupport
Virtual Connections in use: 2
Max Virtual Connections in use: 2
Virtual Connections Overflow: 0
SVC UNI Version: auto_detect

Turboways ATM Adapter Specific Statistics:
------------------------------------------------
Packets Dropped - No small DMA buffer: 0
Packets Dropped - No medium DMA buffer: 0
Packets Dropped - No large DMA buffer: 0
Receive Aborted - No Adapter Receive Buffer: 0
Transmit Attempted - No small DMA buffer: 0
Transmit Attempted - No medium DMA buffer: 0
Transmit Attempted - No large DMA buffer: 0
Transmit Attempted - No MTB DMA buffer: 0
Transmit Attempted - No Adapter Transmit Buffer: 0
Max Hardware transmit queue length: 12
Small Mbuf in Use: 0
Medium Mbuf in Use: 0
Large Mbuf in Use: 64
Huge Mbuf in Use: 0
MTB Mbuf in Use: 0
Max Small Mbuf in Use: 0
Max Medium Mbuf in Use: 0
Max Large Mbuf in Use: 64
Max Huge Mbuf in Use: 0
MTB Mbuf in Use: 0
Small Mbuf overflow: 0
```

```
Medium Mbuf overflow: 0
Large Mbuf overflow: 0
Huge Mbuf overflow: 0
MTB Mbuf overflow: 0
```

This produces the following output on a PCI machine:

```
--------------------                            -------------------
Packets: 299                                    Packets: 294
Bytes: 9727                                     Bytes: 10123
Interrupts: 0                                   Interrupts: 297
Transmit Errors: 0                              Receive Errors: 0
Packets Dropped: 0                              Packets Dropped: 0
                                                Bad Packets: 0
Max Packets on S/W Transmit Queue: 0
S/W Transmit Queue Overflow: 0
Current S/W+H/W Transmit Queue Length: 2

Cells Transmitted: 450                          Cells Received: 457
Out of Xmit Buffers: 0                          Out of Rcv Buffers: 0
Current HW Transmit Queue Length: 2             CRC Errors: 0
Current SW Transmit Queue Length: 0             Packets Too Long: 0
                                                Incomplete Packets: 0
                                                  Cells Dropped: 5

General Statistics:
-------------------
No mbuf Errors: 0
Adapter Loss of Signals: 0
Adapter Reset Count: 0
Driver Flags: Up Running Simplex
        64BitSupport
Virtual Connections in use: 4
Max Virtual Connections in use: 5
Virtual Connections Overflow: 0
SVC UNI Version: uni3.1

IBM PCI 155 Mbps ATM Adapter Specific Statistics:
------------------------------------------------
Total 4K byte Receive Buffers: 96   Using: 64
```

Related Information

The **entstat** command, **fddistat** command, **netstat** command, **tokstat** command.

ATM adapter statistics in the *Networks and communication management*.

---

atq Command

Purpose

Displays the queue of jobs waiting to be run.

Syntax

atq [**c** | **-n**] [*User* ...]

Description

The **atq** command displays the current user's queue of jobs that are waiting to be run at a later date, sorted in the order the jobs will be run. These jobs were created with the **at** command. If the user is root and *User* name is specified, the **atq** command displays only jobs belonging to that user.

Flags

-c Sorts the queue by the time that the **at** command was issued.
-n Displays only the number of jobs currently in the queue.

Security

RBAC Disclaimer

Examples

In order to look at the queue created by the **at** command, enter:

```
atq
```

If there are jobs in the queue, a message similar to the following appears:

```
root.635623200.a          Wed     Feb 21  12:00:00 1990
root.635670000.a          Thu     Feb 22  01:00:00 1990
```

Files

/usr/bin/atq	Contains the **atq** program.
/var/spool/cron/atjobs	Specifies the spool area.

Related Information

The **at** command, **atrm** command.

The **cron** daemon.

Input and output redirection overview in *Operating system and device management* describes how the operating system processes input and output.

Shells in *Operating system and device management* describes what shells are, the different types of shells, and how shells affect the way commands are interpreted.

---

atrm Command

Purpose

Remove jobs spooled by the **at** command.

Syntax

atrm [**-f**] [**-i**] [**-a** | **-**] [*Job* ... | *User* ...]

Description

The **atrm** command removes jobs that were created with the **at** command, but have not executed. If one or more job numbers is specified, the **atrm** command attempts to remove only those jobs.

If one or more user names is specified, all jobs belonging to those users are removed. This form of invoking the **atrm** command is useful only if you have root user authority.

Flags

- Removes all jobs belonging to the user invoking the **atrm** command.

-a Removes all jobs belonging to the user invoking the **atrm** command. This flag is provided for System V compatibility.
-f Suppresses all information about the jobs being removed.
-i Prompts before a job is removed. Enter `y` to remove the job.

Security

RBAC Disclaimer

Examples

To remove job number `root.62169200.a` from the **at** command queue, enter:

```
atrm root.621619200.a
```

Files

/usr/bin/atrm	Contains the **atrm** program file.
/var/spool/cron/atjobs	Specifies the spool area.

Related Information

The **at** command, **atq** command.

The **cron** daemon.

Input and output redirection overview in *Operating system and device management* describes how the operating system processes input and output.

Shells in *Operating system and device management*describes what shells are, the different types of shells, and how shells affect the way commands are interpreted.

---

attachrset Command

Purpose

Attaches an rset to a process.

Syntax

`attachrset [ -P ] [ -F ] [ -S ] rsetname pid`

or

`attachrset [ -P ] [ -F ] [ -c CPUlist ] [ -m MEMlist ] pid`

Description

The **attachrset** command attaches an rset to a process. The command limits the specified process to run only on the processors and/or memory regions contained in the rset. An rset name in the system registry can be attached to the process. Or, an rset containing the specified processors and memory regions can be attached to the process.

Flags

-P Attaches an rset as a partition rset.

-F Forces the rset attachment to occur. This option will remove a bindprocessor bind and all threads' rset in the process before attaching the new rset. If the **-P** option is also specified, it will also detach the effective all threads' rset from the process before attaching the new rset.

-c *CPUlist* List of CPUs to be in the rset. This can be one or more CPUs or CPU ranges.

-m *MEMlist* List of memory regions to be in the rset. This can be one or more memory regions or ranges.

-S A hint that indicates that the process must be scheduled to run in single-threaded mode. Only one of the hardware threads of each physical processor that is included in the specified rset will be used to schedule the job. If all the hardware threads of a physical processor are not included in the specified rset, that processor will be ignored. The specified rset must be an exclusive rset or the command fails. Specifying this flag allows jobs to run with single-thread behavior.

Parameters

rsetname The name of the rset to be attached to the process. The name consists of a *namespace* and an *rsname* separated by a ″/″ (slash). Both the *namespace* and *rsname* may contain up to 255 characters. See the **rs_registername()** service for additional information about character set limits of rset names.

pid Process ID to connect rset.

Security

The user must have `root` authority or have **CAP_NUMA_ATTACH** capability and read access to the specified rset registry name (if **-r** option used) and target process must have the same effective userid as the command issuer. The user must have `root` authority to set the partition rset on a process (the **-P** option).

RBAC Disclaimer

Examples

1. To attach an **rset** containing CPUs 0-7 to process 18838, type:
   ```
   attachrset -c 0-7 18838
   ```
2. To attach **rset** named **test/cpus0to7** to process 20124, type:
   ```
   attachrset test/cpus0to7 20124
   ```

Files

/usr/bin/attachrset Contains the **attachrset** command.

Related Information

The **detachrset**, **execrset**, **lsrset**, **mkrset**, and **rmrset** commands.

---

audit Command

Purpose

Controls system auditing.

Syntax

audit { **on** [**panic** | **fullpath**] | **off** | **query** | **start** | **shutdown** }{**-@** *wparname* ...}

Description

The **audit** command controls system auditing through several keywords. You must include one keyword each time you enter the command. The **start** keyword and the **shutdown** keyword start and stop the auditing system and reset the system configuration. The **off** keyword and the **on** keyword suspend and restart the audit system without affecting the system configuration. The **query** keyword lets you query the current status.

The auditing system follows the instructions established in the following configuration files:

- **/etc/security/audit/config**
- **/etc/security/audit/events**
- **/etc/security/audit/objects**
- **/etc/security/audit/bincmds**
- **/etc/security/audit/streamcmds**

Each of these files is described in the ″Files″ section. For information on configuring the audit system, see ″Setting up Auditing″ in *Operating system and device management*.

The **-@** option is not supported when you run it in a WPAR.

Keywords

start Starts the audit subsystem. This keyword reads the instructions in the configuration files and performs the following tasks:

object auditing
Writes the audit event definitions in the **/etc/security/audit/objects** file into the kernel to define the object auditing events.

event auditing
Writes the audit class definitions in the **/etc/security/audit/config** file into the kernel to define the audit classes.

bin auditing
Starts the **auditbin** daemon according to the configuration information in the bin stanza in the **/etc/security/audit/config** file, if the start stanza contains `binmode=on`.

stream auditing
Invokes the audit stream commands as defined in the stream stanza in the **/etc/security/audit/config** file, if the start stanza contains `streammode=on`.

Attention: Avoid invocation of stream auditing from **/etc/inittab**.

user auditing
Audits all users currently logged into the system, if they are set up in the users stanza of the **/etc/security/audit/config** file.

audit logging
Enables the audit logging component as defined in the start stanza in the **/etc/security/audit/config** file.

audit ranges
Writes the Trusted AIX audit ranges into the kernel if they are set up in the WPAR Audit Ranges (WAR) stanza of the **/etc/security/audit/config** file.

global-initiated WPAR auditing
Audits the WPARs, if they are stored in the WPARS stanza of the **/etc/security/audit/config** file. The auditing can be used only from global WPAR by specifying the **-@** *wparname* parameter in the command.

shutdown Terminates the collection of audit records and resets the configuration information by removing the definition of classes from the kernel tables. All the audit records are flushed from the kernel buffers into the bin files or audit streams, according to the specifications for the backend commands, which are contained in the **/etc/security/audit/bincmds** file for binmode auditing, and in the **/etc/security/audit/streamcmds** file for streammode auditing. The collection of audit data stops until you give the next **audit start** command. When you use the **-@** *wparname* parameter with this keyword, auditing is disabled for the specified WPAR.

off Suspends the auditing system, but leaves the configuration valid. Data collection pauses until you give the **audit on** command. The **-@** option is not supported with this keyword.

on [**panic** | **fullpath**] Restarts the auditing system after a suspension, if the system is properly configured (for example, if the **audit start** command was used initially and the configuration is still valid). If auditing has already started when the command is given, only bin data collection can be changed.

The **-@** option is not supported with this keyword.

If you specify the **panic** option, the system halts abruptly if bin data collection is enabled but cannot be written to a bin file. The **panic** option is not supported when you run it in a WPAR.

If you specify the **fullpath** option, the FILE_Open, FILE_Read and FILE_Write auditing events capture the full path name of a file.

query Queries the auditing status of the audit subsystem. If you specify the **-@** option, this keyword queries the auditing status of a global initiated WPAR. This keyword displays the current status of the audit subsystem in the following format:

```
auditing on {panic | fullpath} | auditing off

bin manager off |  is process number pid

audit events:
    audit class: audit event, audit event...
audit objects:
    object name: object mode: audit event
```

Security

Access Control

This command should grant execute (x) access to the root user and members of the audit group. The command should be **setuid** to the root user and have the **trusted computing base** attribute.

Files Accessed

Mode	File
r	**/etc/security/audit/config**
r	**/etc/security/audit/objects**
x	**/usr/sbin/auditbin**
x	**/usr/sbin/auditstream**

RBAC Disclaimer

Examples

1. To start the audit process, configure the audit system as described in ″Setting up Auditing″ in *Security*, and add the following line to the system initialization file (the **/etc/rc** file):

   ```
   /usr/sbin/audit start 1>&- 2>&-
   ```

   The audit process starts, as configured, each time the system is initialized.
2. To start the audit process for the WPAR named `wpar1` from the global WPAR, enter the following command:

```
/usr/sbin/audit start -@ wpar1
```

3. To terminate the operation of the auditing process, enter the following command:

   ```
   /usr/sbin/audit shutdown
   ```

   Data collection stops until the **audit start** command is specified again. The configuration of classes in the operating system kernel is lost.

   Note: The **audit shutdown** command should be in the **/etc/shutdown** file as well.

4. To terminate the auditing process of the WPAR named `wpar1` from global WPAR, enter the following command:

   ```
   /usr/sbin/audit shutdown -@ wpar1
   ```

   Data collection stops until the **audit start -@ wpar1** command is specified again. The configuration of classes in the operating system kernel is lost.

   Remember: The **audit shutdown** command, without any options, shuts down the auditing process of all WPARs started from the global WPAR.

5. To suspend the audit subsystem, enter the following command:

   ```
   /usr/sbin/audit off
   ```

6. To restart an audit process that was suspended by the **audit off** command, enter the following command:

   ```
   /usr/sbin/audit on
   ```

   The suspended state ends and audit records are generated again, as long as the system is configured correctly.

7. To display the current status of the auditing system, enter the following command:

   ```
   /usr/sbin/audit query
   ```

   The following is an example of an **audit query** status message:

   ```
   auditing on

   bin manager is process number 123

   audit events:
       authentication- USER_Login, USER_Logout
       administration- USER_Create, GROUP_Create

   audit objects:
       /etc/security/passwd :
         r = AUTH_Read
       /etc/security/passwd :
         w = AUTH_Write
   ```

   The query informs you that audit records are written when the specified users log in or log out, when the specified administrators create a user or a group, and when the system receives an authorized read or write instruction for the **/etc/security/passwd** file.

Files

/usr/sbin/audit	Contains the path of the **audit** command.
/etc/rc	Contains the system initialization commands.
/etc/security/audit/config	Contains audit configuration information.
/etc/security/audit/events	Lists the audit events and their tail format specifications.
/etc/security/audit/objects	

	Lists the audit events for each file (object).
/etc/security/audit/bincmds	Contains shell commands for processing audit bin data.
/etc/security/audit/streamcmds	Contains auditstream commands.

Related Information

The **auditbin** daemon, **auditcat** command, **auditconv** command, **auditpr** command, **auditselect** command, **auditstream** command, **login** command, **logout** command, **su** command.

The **audit** subroutine, **auditbin** subroutine, **auditevents** subroutine, **auditlog** subroutine, **auditproc** subroutine.

For general information on auditing, refer to Auditing Overview in *Security*.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to Securing the network in *Security*.

To see the steps you must take to establish an Auditing System, refer to Setting up Auditing in *Security*.

---

auditbin Daemon

Purpose

Manages bins of audit information.

Syntax

auditbin

Description

The **auditbin** daemon in the audit subsystem manages **bin1** and **bin2**, temporary bin files that alternately collect audit event data. The command also delivers bins of data records to backend commands for processing.

As audit events occur, the operating system kernel writes a record to a bin file. When a bin file is full, the **auditbin** daemon reads the **/etc/security/audit/bincmds** file and delivers the bin records to the backend commands defined in the file. Each line of the **/etc/security/audit/bincmds** file contains one or more commands with input and output that can be piped together or redirected. The **auditbin** daemon searches each command for the **$bin** string and the **$trail** string and substitutes the path names of the current bin file and the system trail file for these strings.

The **auditbin** daemon ensures that each command encounters each bin at least once, but does not synchronize access to the bins. When all the commands have run, the bin file is ready to collect more audit records.

If a command is unsuccessful, the **auditbin** daemon stops delivering data records and sends a message to the **/dev/tty** device every 60 seconds until the root user or a member of the audit group stops the command.

Security

Access Control

This command should grant execute (x) access to the root user and members of the audit group. The command should be **setuid** to the root user and have the **trusted computing base** attribute.

Files Accessed

Mode	File
r	**/etc/security/audit/config**
r	**/etc/security/audit/bincmds**
rw	Defined audit bins and trail file
x	All audit bin processing commands

RBAC Disclaimer

Examples

1. To configure the **auditbin** daemon, edit the start and bin stanzas of the **/etc/security/audit/config** file to include the following attribute definitions:

```
start:
         binmode = on

bin:
         trail = /audit/trail
         bin1 = /audit/bin1
         bin2 = /audit/bin2
         binsize = 25000
         cmds = /etc/security/audit/bincmds
```

2. To define the commands that process the audit trail, edit the **/etc/security/audit/bincmds** file to include one or more command lines, such as the following:

```
/usr/sbin/auditcat -p -o $trail $bin

/usr/sbin/auditselect -e "event == USER_Login" \
$bin | /usr/sbin/auditpr >> /etc/log
```

   The first command line appends compressed audit bins to the audit trail file. The second line selects `USER_Login` records from each bin file, passes them to the `auditpr` command for formatting, and appends the records to the `/etc/log` file.

Files

/usr/sbin/auditbin	Specifies the path to the **auditbin** daemon.
/audit/bin*x*	Specifies the path to the default bin collection files, with *x* indicating the bin number.
/etc/security/audit/config	Contains audit system configuration information.
/etc/security/audit/events	Contains the audit events of the system.
/etc/security/audit/objects	Contains audit events for audited objects (files).
/etc/security/audit/bincmds	Contains the **auditbin** backend commands.
/etc/security/audit/streamcmds	Contains the **auditstream** commands.

Related Information

The **audit** command, **auditcat** command, **auditconv** command,**auditpr** command, **auditselect** command, **auditstream** command.

The **audit** subroutine, **auditbin** subroutine.

Auditing Overview in *Security*.

To see the steps you must take to establish an Auditing System, refer to Setting up Auditing in *Security*.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to Securing the network in *Security*.

---

auditcat Command

Purpose

Writes bins of audit records.

Syntax

auditcat [**-p** | **-u**] [**-o** *OutFile*] [**-r**] [*InFile*]

Description

The **auditcat** command is part of the audit subsystem, and is one of several backend commands that process the audit data records.

The **auditcat** command reads bin files of audit records from standard input or from the file specified by the *InFile* parameter. The command then processes the records and writes its output to standard output or to the file specified by the *0utFile* parameter. The output can be compressed or not, depending on the flag selected.

One major use of the command is appending compressed bin files to the end of the system audit trail file.

If the **/etc/security/audit/bincmds** file includes **$bin** as the input file, input comes from the current bin file, **bin1** or **bin2**. If the **/etc/security/audit/bincmds** file includes **$trail** as the output file, the records are written to the end of the system audit trail file.

If a bin file is not properly formed with a valid header and tail, an error is returned. See the **auditpr** command for information about audit headers and tails and the **auditbin** command for information on error recovery.

Flags

-o *OutFile*	Specifies the audit trail file to which the **auditcat** command writes records. If you specify **$trail** as the file for the *OutFile* parameter, the **auditbin** daemon substitutes the name of the system audit trail file.
-p	Specifies that the bin files be compressed (packed) upon output. The default value specifies that the bins not be compressed.
-r	Requests recovery procedures. File names for both the *InFile* and *OutFile* parameters must be specified for recovery to occur, so the command syntax must be **auditcat -o** *OutFile* **-r** *InFile*. The command checks to see if the bin file specified for the *InFile* parameter is appended and if not, appends the bin file to the file specified by the *OutFile* parameter. If the bin file is incomplete, the **auditcat** command adds a valid tail and then appends the bin file to the file specified by the *OutFile* parameter.
-u	Specifies that compressed trail files be uncompressed upon output.

Security

Access Control

This command should grant execute (x) access to the root user and members of the audit group. The command should be setuid to the root user and have the **trusted computing base** attribute.

RBAC Disclaimer

Examples

To configure the system to append audit bin data to the system audit trail file, add the following line to the **/etc/security/audit/bincmds** file:

```
/usr/sbin/auditcat  -o $trail $bin
```

When the **auditbin** daemon calls the **auditcat** command, the daemon replaces the **$bin** string with the path name of the current bin file, and replaces the **$trail** string with the name of the default audit trail file.

Files

/usr/sbin/auditcat	Specifies the path to the **auditcat** command.
/etc/security/audit/config	Contains audit system configuration information.
/etc/security/audit/events	Contains the audit events of the system.
/etc/security/audit/objects	Contains audit events for audited objects (files).
/etc/security/audit/bincmds	Contains auditbin backend commands.

Related Information

The **audit** command, **auditconv** command, **auditpr** command, **auditselect** command.

auditbin daemon.

For general information on auditing, refer to Auditing Overview in *Security*.

To see the steps you must take to establish an Auditing System, refer to Setting up Auditing in*Security*.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to Securing the network in *Security*.

---

auditconv Command

Purpose

Converts pre-AIX Version 4 format audit bins to AIX Version 4 format.

Syntax

auditconv *OldFile NewFile*

Description

The **auditconv** command converts audit records which were generated by previous versions of the operating system into the format used by AIX Version 4 and higher of the operating system.

Audit records are read from the file *OldFile*, and written to the file *NewFile*. Each audit record is updated with thread information, with a default thread identifier of zero.

Notes:

1. The *OldFile* and *NewFile* parameters must be different, and must not be currently in use by the audit system.
2. AIX Version 4 and higher of the operating system cannot work with pre-AIX Version 4 audit bins. Therefore, old bins must be converted using the **auditconv** command.

Security

Access Control

This command should grant execute (x) access to the root user and members of the audit group. The command should be setuid to the root user and have the **trusted computing base** attribute.

Files Accessed

Mode	File
r	**/etc/security/audit/events**
r	**/etc/passwd**
r	**/etc/group**

RBAC Disclaimer

Example

To convert the old audit file **pre_v4_auditbin**, storing the results in **converted_auditbin**, enter the following command:

```
/usr/sbin/auditconv pre_v4_auditbin converted_auditbin
```

Files

/usr/sbin/auditconv	Specifies the path of the **auditconv** command.
/etc/security/audit/config	Contains audit system configuration information.
/etc/security/audit/events	Contains the audit events of the system.
/etc/security/audit/objects	Contains information about audited objects (files).
/etc/security/audit/bincmds	Contains auditbin backend commands.
/etc/security/audit/streamcmds	Contains auditstream commands.

Related Information

The **audit** command, **auditbin** daemon, **auditcat** command, **auditpr** command, **auditselect** command, **auditstream** command.

The **audit** subroutine.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to Securing the network in *Security*.

To see the steps you must take to establish an Auditing System, refer to Setting up Auditing in *Security*.

auditmerge Command

Purpose

Combines multiple audit trails into a single trail.

Syntax

/usr/sbin/auditmerge [**-q**] *file* [*file* ...]

Description

The **auditmerge** command combines multiple audit trail files from potentially multiple machines into a single audit trail file. For each file with records remaining, the record that has the oldest time stamp is added to the output. If a record is found that has a negative time change, an optional warning message may be emitted. Processing continues and any such records are output with their time values unmodified.

The **auditmerge** command also is capable of adding CPU ID values from the bin header to each output record. The CPU ID value is encoded in the bin header and trailer for bins with a version number more recent than AIX 4.3.1.

The **-q** flag is used to control outputting warning messages. When a record with a negative time change is first seen, a single warning message is output. That message contains the name of the file containing the record and the time difference. These messages are suppressed when the **-q** flag is given on the command line.

Flags

-q Used to control outputting warning messages.

Security

Access Control: This command should grant execute (x) access to the root user and members of the audit group. The command should be setuid to the root user and have the **trusted computing base** attribute.

Examples

1. To merge two existing audit trails files from different hosts, enter:
   ```
   /usr/bin/auditmerge /audit/trail.calvin /audit/trail.hobbes > /audit/trail.merge
   ```
2. To merge two existing data files which were preselected for different user names, enter:
   ```
   /usr/bin/auditmerge /audit/trail.jim /audit/trail.julie > /audit/trail.both
   ```
3. To merge two data files without producing warnings about incorrect times, enter:
   ```
   /usr/bin/auditmerge -q /audit/jumbled.1 /audit/jumbled.2 > /audit/jumbled.output
   ```

Files

/etc/security/audit/hosts	Contains the CPU ID to hostname mappings.

Related Information

The **auditpr** command, **auditstream** command, **auditselect** command.

The **auditread** subroutine, **getauditthostattr** subroutine.

auditpr Command

Purpose

Formats bin or stream audit records to a display device or printer.

Syntax

auditpr [**-i** *inputfile*] [**-t 0** | **1** | **2**] [**-m** *Message*] [**-r**] [**-v**] [**-h** *field*[**,***field*]*]

Description

The **auditpr** command is part of the audit subsystem. This command reads audit records, in bin or stream format, from standard input and sends formatted records to standard output.

The output format is determined by the flags that are selected. If you specify the **-m** flag, a message is displayed before each heading. Use the **-t** and **-h** flags to change the default header titles and fields and the **-v** flag to append an audit trail. The **auditpr** command searches the local **/etc/passwd** file to convert user and group IDs to names.

An example of output using default header information follows:

```
event   login   status   time                            command
        wpar name
login   dick     OK       Fri  Feb;8   14:03:57    1990   login
        Global
 . . . . . trail portion . . . . .
```

For examples of audit trails, see the **/etc/security/audit/events** file where audit trail formats are defined.

Invalid records are skipped when possible, and an error message is issued. If the command cannot recover from an error, processing stops.

Flags

-h *field*[**,***field*]*

Selects the fields to display and the order in which to display them, by default **e**, **l, R**, **t**, and **c**. You can specify the following values:

e The audit event.

l The login name of the user.

R The audit status.

t The time the record was written.

c The command name.

r The real user name.

p The process ID.

P The ID of the parent process.

T The kernel thread ID. This is local to the process; different processes may contain threads with the same thread ID.

h The name of the host that generated the audit record. If there is no CPU ID in the audit record, the value **none** is used. If there is no matching entry for the CPU ID in the audit record, the 16 character value for the CPU ID is used instead.

i The IDs or the names of roles of the audited process.

E The effective privilege.

S The effective sensitivity label (SL).

I The effective integrity label (TL).

W The workload partition name.

-i *inputfile*

Indicates the path to the audit trail file. If the **-i** flag is not specified, the **auditpr** command reads data from **stdin**.

-m *"Message"*

Specifies a *Message* to be displayed with each heading. You must enclose the *Message* string in double quotation marks.

-r

Suppresses ID translation to the symbolic name.

-t {**0** | **1** | **2**}

Specifies when header titles are displayed. The default title consists of an optional message (see the **-m** flag) followed by the name of each column of output.

0 Ignores any title.

1 Displays a title once at the beginning of a series of records.

2 Displays a title before each record.

-v

Displays the trail of each audit record, using the format specifications in the **/etc/security/audit/events** file.

Security

Access Control

This command should grant execute (x) access to the root user and members of the audit group. The command should be **setuid** to the root user and have the **trusted computing base** attribute.

Files Accessed

Mode	File
r	/etc/security/audit/events
r	/etc/passwd
r	/etc/group

RBAC Disclaimer

Examples

1. To read the system audit trail file with default header titles and fields and an audit trail, enter:

   ```
   /usr/sbin/auditpr  -v < /audit/trail
   ```

   The **/audit/trail** file must contain valid audit bins or records.
2. To format from an audit trail file all the audit events caused by user `witte`, enter:

   ```
   /usr/sbin/auditselect -e"login == witte"\
   /audit/trail | auditpr  -v
   ```

   The resulting record is formatted with the default values (**e**, **c**, **l**, **R**, and **t**) and includes a trail.
3. To read records interactively from the audit device, enter:

   ```
   /usr/sbin/auditstream | /usr/sbin/auditpr -t0 -heRl
   ```

Files

/usr/sbin/auditpr	Specifies the path of the **auditpr** command.
/etc/security/audit/config	Contains audit system configuration information.
/etc/security/audit/events	Contains the audit events of the system.
/etc/security/audit/objects	Contains audit events for audited objects (files).
/etc/security/audit/bincmds	Contains auditbin backend commands.
/etc/security/audit/streamcmds	Contains auditstream commands.
/etc/security/audit/hosts	Contains the CPU ID to host name mappings.

Related Information

The **audit** command, **auditcat** command, **auditconv** command, **auditselect** command, **auditstream** command.

The **auditbin** daemon.

The **audit** subroutine.

The **events** file.

For general information on auditing, refer to Auditing Overview in *Security*.

To see the steps you must take to establish an Auditing System, refer to Setting up Auditing in *Security*.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to Securing the network in *Security*.

---

auditselect Command

Purpose

Selects audit records for analysis according to defined criteria.

Syntax

auditselect { **-e** *"Expression"* | **-f** *File*} [**-m**] [*Trail*]

Description

The **auditselect** command is part of the audit subsystem. The command is called by the **auditbin** daemon if it is configured in the **/etc/security/audit/bincmds** file as a backend command for processing bin files.

The **auditselect** command selects audit records that match identified criteria and writes the records to standard output. With the **auditselect** command, you can filter the audit trail to obtain specific records for analysis or select specific records for long-term storage. The command takes stream or bin input from the file specified by the *Trail* parameter or from standard input. If you specify the **$bin** string as the value of the *Trail* parameter, the **auditbin** daemon substitutes the path name of the current bin file when it calls the **auditselect** command. The selection criteria can be entered as an expression or from the file specified by the **-f** flag. If the bin files are compressed, the **auditselect** command unpacks them prior to processing.

For stream data, configure both the **auditstream** command and the **auditselect** command in the **/etc/security/audit/streamcmds** file, or enter both commands from the command line.

Flags

-e *"Expression"*	Defines the selection criteria. The *Expression* parameter consists of one or more terms joined by logical operators.
-f *File*	Specifies the *File* that contains the selection criteria.
-m	Specifies the output audit record with record extensions.

Creating Expressions

A valid expression consists of one or more terms joined by logical operators.

Logical Operators

Logical operators allow more than one term to be used in an expression. Normal precedence rules apply in evaluating expressions with more than one logical operator, and parentheses may be used to force the order of evaluation. The valid logical operators include the following:

&&	(And) The expression `term1 && term2` is true (selected) if both term1 and term2 are true.
II	(Or) The expression `term1 || term2` is true (selected) if either term1 or term2 is true.
!	(Not) The expression `!term1` is true (selected) if term1 is not true.

Terms

Each term of the expression has the following form:

Field Relational_Operator Value

Fields

Fields correspond to the information in the audit header of each record. Valid values for fields include the following:

`event`	Name of the audit event, for example, `FILE_Open`.
`command`	Name of the command that generated the audit event.

`result` Status of the audit event. The value of the `result` field must be one of the following:

- **OK**
- **FAIL**
- **FAIL_PRIV**
- **FAIL_AUTH**
- **FAIL_ACCESS**
- **FAIL_DAC** Indicates the event failed because of a discretionary access control (DAC) denial. Access Control Lists are a form of information repository that contain data relative to the rights of access (permission) to shared resources/objects. ACLs are categorized on DAC mechanism.

FAIL matches all other error codes.

`login` ID of the login user of the process that generated the audit event.

`real` ID of the real user of the process that generated the audit event.

`pid` ID of the process that generated the audit event.

`ppid` ID of the parent of the process that generated the audit event.

`tid` ID of the kernel thread that generated the event.

`time` Time of day the audit event was generated.

`date` Date the audit event was generated.

`host` Hostname of the machine that generated the record. The reserved name **UNKNOWN** can be used to match any machines that are not listed in the **/etc/security/audit/hosts** file.

Relational Operators

Relational operators are used to compare the field in the audit record to the specified value. Valid relational operators include:

== Equal to

!= Not equal to

< Less than

> Greater than

>= Greater than or equal to

<= Less than or equal to

Valid Terms

A valid term consists of a field, a relational operator, and a value. In addition, not all relational operators and values are valid for each field. The following are the valid combinations:

Field	Valid Operators	Valid Values
event	= =, ! =	Text string audit event name
result	= =, ! =	Text string audit status codes
command	= =, ! =	Text string command name
pid	all	Decimal integer process ID
ppid	all	Decimal integer process ID
login	all	Decimal integer user ID
login	= =, ! =	Text string user name
real	all	Decimal integer user ID
real	= =, ! =	Text string user name
tid	all	Decimal integer thread ID
time	all	String in the format specified by the current locale
date	all	String in the format specified by the current locale
host	= =, ! =	Text string host name or 16 character cpu ID

priv	= =, ! =	Privilege name
sl	= =, ! =	Sensitivity label name
tl	= =, ! =	Integrity label name
row	= =, ! =	Role name

Security

Access Control

This command should grant execute (x) access to the root user and members of the audit group. The command should be **setuid** to the root user and have the **trusted computing base** attribute.

RBAC Environment and Trusted AIX

This command implements and can perform privileged operations. Only privileged users can run such privileged operations. For more information about authorizations and privileges, refer to the RBAC section: Privileged Command Database in *Security*. To review the list of privileges and the authorizations associated with this command, refer to the **/etc/security/privcmds** database.

Examples

Configuration

1. To select bin-collected data records that match the USER_SU or USER_Login audit events, add the **auditselect** command to the **/etc/security/audit/bincmds** file by entering:

   ```
   /usr/sbin/auditselect -e "event== USER_SU || event== \
   USER_Login" $bin >> /audit/trail.login
   ```

   While auditing is enabled, the records for each initiation of a user session are read from the current bin file and written to the **/audit/trail.login** file.

2. To select stream-collected data records that match a user login that was unsuccessful, add the **auditselect** command to the **auditstream** stanza in the **/etc/security/audit/streamcmds** file by entering:

   ```
   /usr/sbin/auditstream -c authentication | \
   /usr/sbin/auditselect -e "event == \
   USER_Login && result == FAIL" | \
   /usr/sbin/auditpr -t 2 -v >> /dev/lpr2
   ```

   To produce a hardcopy audit trail, records of unsuccessful authentication events are written to the **/dev/lpr2** line printer.

Select Authentication or Login Events

1. To search an audit trail file for all events that involve authentication errors:

   ```
   /usr/sbin/auditselect -e "result == FAIL_AUTH"
   /audit/oldtrail | /usr/sbin/auditpr -t -helt -v
   ```

   The records of events that were unsuccessful because authentication was denied are printed. The header titles will be printed once, followed by the event, login ID, and time fields, and then the audit trail.

2. To select audit records that are generated when `smith` logs in during prime working hours during the first week in May of 1987, enter:

   ```
   /usr/sbin/auditselect  -f /aaa/bbb \
   /audit/trail1987 | /usr/sbin/auditpr
   ```

   The `/aaa/bbb` file must contain the following line:

```
command == login && login == smith &&
time >= 08:00:00 && time <= 17:00:00 &&
date >= 05/01/87 && date <= 05/05/87
```

String Comparison

1. To compare the name of the audit event to the USER_Login string, enter one of the following:

   ```
   "event == USER_Login"

   "event != USER_Login"
   ```

2. To find out if the **passwd** command generated the audit event, use:

   ```
   "command == passwd"
   ```

   To find out if the audit event was not generated by the **passwd** command, use:

   ```
   "command != passwd"
   ```

3. To compare the audit status to the OK result string, enter:

   ```
   "result == OK"
   ```

4. To compare the login or real user ID of the process that generated the audit event to a specific user ID (user ID 014 or the user name carol), enter one of the following:

   ```
   "login == 014"
   "login != carol"
   "login == 014 || login != carol"
   "real == carol"
   ```

5. To compare the ID of the process or the parent of the process that generated the audit event to the process ID 2006, enter one of the following:

   ```
   "pid == 2006"
   "pid != 2006"
   "ppid ==  2006"
   ```

   Note: Although login and real user IDs and process IDs can be compared with the inequality operators (**< =**, **> =**, **<**, **>**), it is normally unnecessary to do this.

6. To compare the time the audit event was generated to the 08:03:00 time string, enter one of the following:

   ```
   "time == 08:03:00"
   "time != 08:03:00"
   "time <  08:03:00"
   "time <= 08:03:00"
   "time >  08:03:00"
   "time >= 08:03:00"
   ```

   Audit records are selected that fit the indicated comparison to the 08:03:00 time string. The time string must agree with the format specified by the current locale.

7. To compare the date that the audit event was generated to the 05/05/89 date string, enter one of the following:

   ```
   "date == 05/03/89"
   "date != 05/03/89"
   "date <  05/03/89"
   "date <= 05/03/89"
   "date >  05/03/89"
   "date >= 05/03/89"
   ```

   Audit records are selected that fit the indicated comparison to the 05/05/89 date string. The date string must agree with the format specified by the current locale.

Files

/usr/sbin/auditselect	Specifies the path of the **auditselect** command.

/etc/rc	Contains the system initialization commands.
/etc/security/audit/config	Contains audit system configuration information.
/etc/security/audit/events	Contains the audit events of the system.
/etc/security/audit/objects	Contains audit events for audited objects (files).
/etc/security/audit/bincmds	Contains auditbin backend commands.
/etc/security/audit/streamcmds	Contains auditstream commands.
/etc/security/audit/hosts	Contains the CPU ID to hostname mappings.

Related Information

The **audit** command, **auditcat** command, **auditconv** command, **auditpr** command, **auditstream** command, **env** command.

auditbin daemon.

For general information on auditing, refer to Auditing Overview in *Security*.

To see the steps you must take to establish an Auditing System, refer to Setting up Auditing in *Security*.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to Securing the network in *Security*.

---

auditstream Command

Purpose

Creates a channel for reading audit records.

Syntax

auditstream [**-m**] [**-c** *Class ...*]

Description

The **auditstream** command is part of the audit subsystem. This command reads audit records from the **/dev/audit** file (the audit device) and copies the records to standard output in binary format. You can select a subset of the audit records by specifying audit classes (defined in the **/etc/security/audit/config** file) with the **-c** flag; otherwise, all currently enabled audit classes are copied.

Audit stream data can be displayed and processed as it is generated. For example, the command output can be piped to an audit backend command for further processing or redirected to a file. Both the **auditselect** command, which selects data records according to defined criteria, and the **auditpr** command, which formats the records for viewing or for printing, are examples of backend commands.

The **auditstream** command can be called from the command line or be configured to run multiple times as part of the audit system configuration. For information on configuring the **auditstream** command, refer to ″Setting up Auditing″ in *Security* and to the **/etc/security/audit/config** file.

> **Note:** The **auditstream** command should be run in the background.

Flags

-c *Class*	Specifies the audit classes to be copied. Each class must be configured in the **etc/security/audit/config** file as a list of comma-separated audit events. The default value is all the currently enabled audit events.
-m	Includes the processor ID, roles and privileges in each audit record.

Security

Access Control

This command should grant execute (x) access to the root user and members of the audit group. The command should be **setuid** to the root user and have the **trusted computing base** attribute.

Files Accessed

Mode	File
r	/dev/audit

RBAC Disclaimer

Examples

1. To configure the stream collection of audit data when the audit system is initialized, add the following to the stream stanza of the **/etc/security/audit/config** file:

   ```
   cmds  =  /etc/security/audit/streamcmds
   ```

   Then add the following to the start stanza:

   ```
   streammode=on
   ```

   Next, add to the /**etc/security/audit/streamcmds** file all the stream commands that should be executed when the auditing system is initialized. For example:

   ```
   /usr/sbin/auditstream  -c  authentication  |  \
   /usr/sbin/auditpr  -v  >  /dev/console

   /usr/sbin/auditstream  |  /usr/sbin/auditselect  -e  \
   "result  ==  FAIL_ACCESS"  |  \
   /usr/sbin/auditpr  -t  2  -v  >  /dev/lpr2
   ```

   The first command formats all records for events in the authentication class and writes them to the system console. The second command formats all records that resulted in an access denial and prints them on the printer **/dev/lp2**.

2. To record audit stream events on a line printer, enter:

   ```
   /usr/sbin/auditstream  |  /usr/sbin/auditselect  -e  "event  ==  \
   USER_Login  ||  event  ==  USER_SU"    |  \
   /usr/sbin/auditpr  -v  >  /dev/lp0  &
   ```

   This command formats and writes all user login and **su** events to the line printer.

Files

/usr/sbin/auditstream	Specifies the path of the **auditstream** command.
/etc/rc	Contains the system startup routines.
/dev/audit	Specifies the audit device.
/etc/security/audit/config	Contains audit system configuration information.
/etc/security/audit/events	Contains the audit events of the system.
/etc/security/audit/objects	Contains audit events for audited objects (files).
/etc/security/audit/bincmds	Contains auditbin backend commands.
/etc/security/audit/streamcmds	Contains auditstream commands.
/etc/security/audit/hosts	Contains host and processor IDs.

Related Information

The **audit** command, **auditcat** command, **auditconv** command, **auditpr** command, **auditselect** command.

The **auditbin** daemon.

For general information on auditing, refer to Auditing Overview in *Security*.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to Securing the network in *Security*.

To see the steps you must take to establish an Auditing System, refer to Setting up Auditing in *Security*.

---

authexec Command

Purpose

Runs a Role Based Access Control (RBAC) privileged command in a controlled manner.

Syntax

authexec *RBACcommandName*

Description

The **authexec** command runs a RBAC privileged command. When **authexec** is issued, users are authenticated against the roles defined in the **authroles** attribute for the RBAC command, *RBACcommandName*, in the RBAC privileged command database.

The **authexec** command in located in /usr/sbin/.

The user invoking **authexec** must have enough authorization to invoke the target command, *RBACcommandName*. The authenticating users should not be the same as the invoking user. The authenticating users must also have a valid non-blank password to successfully pass the authentication. No user can be authenticated more than once for any role. A maximum of sixteen roles can be configured for the RBAC privileged command.

A privileged command having the **authexec** attribute in the privileged command database cannot be run directly from shell or by using the exec subroutines in programs. Such commands have to be necessarily invoked using the **authexec** command.

This mechanism is not enforced when the command *RBACcommandName* is invoked by root in a root enabled RBAC system.

Parameters

RBACcommandName	Specifies the RBAC target command to run, including any flags or parameters. You must specify the absolute path of the target command, *RBACcommandName*.

Security

Access Control: All users can invoke this command.

Examples

If the command **usr/sbin/shutdown** is enabled for authenticated execution using the **authroles** attribute, then a user that is authorized to the shutdown command can run:

```
authexec  /usr/sbin/shutdown
```

The following example shows the **usr/sbin/shutdown** command that is enabled for authenticated execution using the **authrole** attribute:

```
/usr/sbin/shutdown:
accessauths=aix.system.boot.shutdown
innateprivs=PV_AZ_ROOT,PV_DAC_O,PV_DAC_R,PV_DAC_W,
PV_DAC_X,PV_PROC_PRIV,PV_PROC_SIG
secflags=FSF_EPS
authroles=isso,so,sa
```

Before the **shutdown** command is run, three distinct users having one of the three roles listed in **authroles** attribute have to be authenticated. In this example, **authroles** attribute specifies the **isso**, **so**, and **sa** roles. This command requires the access authorization aix.system.boot.shutdown to invoke the **shutdown** command. This authorization is typically associated with the **so** role. A user, other than the user invoking the **shutdown** command, with the role **so** in addition to users with the **isso** and **sa** roles must authenticate to successfully issue the command.

Files

/etc/security/users	Contains the extended attributes of users.
/etc/security/roles	Contains the attributes of roles.
/etc/security/authorizations	Contains the attributes of authorizations
/etc/security/privcmds	Contains the attributes of RBAC privileged commands.

Related Information

The **setsecattr** command and **lssecattr** command.

The **privcmds** file.

---

authrpt Command

Purpose

Reports the security capabilities of authorizations.

Syntax

authrpt [**-R***load_module*] [**-C**] [**-c** | **-f** | **-r** | **-u**] { auth1,auth2 ... }

Description

The **authrpt** command reports capability information of authorizations such as privileged commands, privileged files, role, and user information.

Either **–c**, **-f**, **-r** or **–u** flags can be specified.

When the **-c** option is specified, the privileged commands present in the **/etc/security/privcmds** database that can be executed by the authorizations is listed. The **–c** option can also be used to list the commands having ALLOW_ALL, ALLOW_GROUP and ALLOW_OWNER special authorizations.

When the **–f** option is specified, the list of privileged files present in the **/etc/security/privfiles** database that can be accessed by a user assigned the authorizations is listed.

When the **–u** option is specified, the list of users having the authorizations is displayed.

When the **–r** option is specified, the list of roles having the authorizations is listed.

The command takes a comma separated list of authorization names as input. When no option is specified, all the capability information such as commands, privileged files, roles and user information associated with the authorizations is listed.

Flags

-c	Specify that a report of privileged commands executable by the authorizations is to be obtained.
-f	Specify that a report of privileged file information accessible by the authorizations is to be obtained.
-u	Specify that a report of authorized users having the authorizations is to be obtained.
-r	Specify that a report of roles having the authorizations is to be obtained.
-R	Specifies the loadable module from which to obtain the report of authorization capabilities.

-C	Displays the authorization attributes in colon-separated records, as follows:
	`authorizaton:attribute1:attribute2: ...`
	`authorization1:value1:value2: ...`
	`authorization2:value1:value2: ...`

Exit status

0	Successful completion.
>0	An error occurred.

Security

Access Control: This command should grant execute (x) access to the root user.

This command can be executed by root or an authorized user with the “aix.security.auth.list” authorization.

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

Files

/etc/security/roles
/etc/security/authorizations
/etc/security/privcmds
/etc/security/privfiles

Examples

To report the commands associated with authorizations aix.fs and aix.system, use the following syntax:

```
authrpt –c aix.fs,aix.system
```

To report all capabilities of authorization aix.security, use the following syntax:

```
authrpt aix.security
```

To report all capabilities of authorization aix.security.user in colon separated format, use the following syntax:

```
authrpt –C aix.security.user
```

Information similar to the following appears:

```
#authorization:commands:privfiles:roles:users
aix.security.user:/usr/bin/mkuser,
/usr/bin/chuser:/etc/csh.cshrc,
/etc/csh.login:role1:Bob,Simon
```

Related Information

The **rolerpt** and **usrrpt** commands

The **/etc/security/privcmds** file in *AIX 5L Version 5.3 Files Reference*.

---

authqry Command

Purpose

Queries the usage of authorizations over a time period.

Syntax

authqry { **-c** [**-s**] | **-q** [**-F** *<trailListFile>*] [**-t** *<time_period_in_days>*] } *user*

Description

The **authqry** command queries information about the authorizations used by a user over a specified time frame.

When the **–c** option is specified, the user is configured for the auditing of role and authorization information. A class **rbacqry** is added to the **/etc/security/audit/config** file with events for auditing authorizations and roles. If the user is already being audited (user entry present in the configuration file), then the **rbacqry** class is added to the user. Otherwise the username is added to the **/etc/security/audit/config** with the **rbacqry** class parameter.

When the **-s** option is specified, the auditing subsystem is started / restarted.

When the **–q** option is specified, the audit data is queried for authorization information.

When the **–t** option is specified, the usage of authorizations from the date (specified through the **–t** option) to the current system date is queried and obtained. Without **–t** option, authorization usage over the period from which auditing was enabled for that user is obtained. The command displays the entire set of authorizations used during this time frame.

Note: The **authqry** command makes use of the auditing feature in AIX. For the **authqry** command to work as expected, auditing must be turned on, audit configuration for the user must be enabled and a time frame must be specified in days.

Flags

-c Specifies to configure the user for auditing of authorization usage.

-s	Start auditing subsystem if it is turned off. Restart if already turned on.
-q	Specifies to query audit data for authorization usage over a specified time period.
-F	The **–F** option reads the names of the audit trails to obtain audit information from the *trailListFile*. The names of audit trail files should be one per line of text. If the **–F** option is not specified, the system **/audit/trail** file is taken by default as the file to obtain audit information from.
-t	Specify the number of days from the current date to get the authorization usage.

Exit status

0	Successful completion.
>0	An error occurred.

Security

Access Control: This command should grant execute (x) access to only the root user.

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

Files

/etc/security/authorizations
/audit/trail

Examples

To query authorizations by Bob, use the following syntax:

```
authqry Bob
```

To query authorizations used by Simon for the past 20 days, use the following syntax:

```
authqry –t 20 Simon
```

Related Information

The **roleqry** and **audit** commands

The events file.

For general information on auditing, refer to *Auditing Overview* in *Security*.

To see the steps you must take to establish an Auditing System, refer to *Setting up Auditing* in *Security*.

---

autoconf6 Command

Purpose

Automatically configures IPv6 network interfaces at boot time.

Syntax

autoconf6 [**-a**] [**-A**] [**-i**] [**-s**] [**-6**] [**-M**] [**-O**] [**-R**] [**-c**] [**-v**] [**-m** *main_interface*] [*interface_name* ...]

Description

The **autoconf6** command is used at boot time to assign link-local addresses to ND-capable network interfaces. The **autoconf6** command initializes also the loopback interface, the automatic tunnels if needed, and adds some needed routes. It can also be used at any time to set link-local addresses and automatic tunnelling on newly configured ethernet-like interfaces.

Flags

-a	Configures and turns up all acceptable interfaces that are already configured with IPv4.
-A	Configures and turns up all acceptable interfaces.
-i	Configures and turns up the interfaces in the argument list. Without the **-a** and **-i** flags only the interfaces already up are configured.
-m *main_interface*	Specifies the main interface. You can also use the **no** command with the argument, **main_if6**.
-s	Installs the SIT interfaces and IPv4-compatible programs. Without this flag, the SIT interfaces are configured only if an SIT interface is already up.
-6	The SIT interface and IPv4-compatible interoperability are not installed or modified.
-M	(Debug) Do not modify existing IPv6 multicast routes.
-O	(Debug) Do not configure the loopback interface.
-R	(Debug) Do not install a default IPv6 route.
-c	Old compatibility flag for those who have bad LL addresses.
-v	Verbose output. The program displays what it is doing and/or what it is failing.
interface_name	Specifies the names of the interfaces that should be configured. This is used with the **-i** flag. If the **-i** flag is given and no interface_names are specified, no interfaces are configured. If an interface_name is given and the **-i** flag is not specified, a usage message is displayed. If ibX is specified as the interface name, the ibX interface is configured with an IPv6 address based on the EUI-64 for the InfiniBand port. To use the ibX interface with the **autoconf6** command, the ibX interface must be previously configured with an IPv4 address.

Messages

Messages indicate the different actions done and/or problems encountered by **autoconf6**.

Related Information

The **ifconfig** command, **ndpd-host** command, **ndpd-router** command, and **route** command.

---

automount Daemon

Purpose

Mounts automatic mount points.

Syntax

/usr/sbin/automount [**-m**] [**-n**] [**-v**] [**-t** *duration*] [**-i** *interval*] [**-f** *file*] [**-s** *timeout*] [**-D** *value*] [**-d** *value*]

Description

The **automount** command is used as an administration tool for **AutoFS**. It installs **AutoFS** mount points and associates an **automount** map with each mount point. The **AutoFS** file system monitors attempts to access directories within it and notifies the **automountd** daemon. The daemon uses the map to locate a file system, which it then mounts at the point of reference within the **AutoFS** file system.

The previous **automount** behavior can be specified if the **COMPAT_AUTOMOUNT** environment variable is set to any value before running the **automount** command. The current behavior became the default behavior in AIX 5.0.

If the file system is not accessed within an appropriate interval (ten minutes by default), the **automountd** daemon unmounts the file system.

If the **automountd** daemon has not been started the **automount** command attempts to start it using **SRC**.

Maps

Automount maps specify the mount points to be automatically mounted when accessed, and what should be mounted over those mount points. The **/etc/auto_master** map file specifies the initial mount points, known as *keys*, and their corresponding maps that determine which remote filesystem is mounted over it. The format of the **/etc/auto_master** file is:

```
/key    map
```

Note: The **/etc/auto_master** file is only read when the **automount** command is initially executed. Changes to it will not take effect until the **automount** command is run again.

The most common maps are direct maps, indirect maps, and host maps.

Direct maps require a special key (/-) in the **/etc/auto_master** file, and their map is a file with the following format:

```
/directkey     [-options]    server:/dir
```

When a user accesses the **/directkey** directory, the **automountd** daemon will mount **server:/dir** over **/directkey**.

Indirect maps have the following format:

```
indirectkey    [-options]    server:/dir
```

When a user accesses the **/key/indirectkey** directory, the **automountd** daemon will mount **server:/dir** over **/key/indirectkey**.

Host maps require a special map (-hosts) in the **/etc/auto_master** file. The **automountd** daemon will create a subdirectory under the **/key** directory for every server listed in the **/etc/hosts** file. When a user accesses the **/key/server** directory, the **automountd** daemon will mount the server's exported directories over the **/key/server** directory.

Alternate Map Locations: Automount maps may also be located on NIS/NIS+ and LDAP servers. The **automount** command will look for maps as files on the local system by default, unless the automount entry in the **/etc/irs.conf** file is changed. For example:

```
automount nis_ldap
```

It is possible to specify more than one name service, in the order that they will be used, by using a whitespace separated list. For example, to indicate that LDAP maps should be used first, followed by local files, the automount entry would be the following:

```
automount nis_ldap files
```

The valid values for the automount entry are `files`, `nis`, `nisplus`, and `nis_ldap`.

For more information on how to manage automount maps in NIS/NIS+, see Managing NIS Automount Maps in *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*.

For more information on how to manage automount maps in LDAP, see the Managing LDAP Automount Maps section in PC-NFS in *Networks and communication management*.

Flags

-d *value*	Specifies the debug level of the autofs extension and **automount** daemon.
-D *value*	Specifies an environment variable and its value.
-f *file*	Specifies a new master map file to use. The default is **/etc/auto_master**.
-i *Interval*	Specifies the amount of time, in seconds, that an inactive autofs mounted directory exists.
-m	Specifies not to search NIS for automount maps.
-n	Specifies the nobrowse option.
-s *timeout*	Specifies the amount of time, in seconds, before a new process is forked off if a mount takes too long. The minimum value is 30.
-t *Duration*	Specifies the amount of time, in seconds, that the auto unmount process sleeps before it starts to work again. The minimum value is 21. The default value is 120. The maximum value is 600.
-v	Displays on standard output verbose status and warning messages.

Files

/etc/auto_master	The default map file used to create the initial automount keys.
/etc/hosts	Specifies servers that will be used in automount host maps.
/etc/irs.conf	Specifies the location of the automount maps.

Related Information

The **mount** command.

Managing NIS Automount Maps in *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide* discusses map formatting, multiple mounts, special maps, and the **auto.master** NIS configuration map file.

Managing LDAP Automount Maps section in PC-NFS in *Networks and communication management*

Network File System (NFS) Overview for System Management in *Networks and communication management*.

List of NFS commands in *Networks and communication management*.

---

automountd Daemon

Purpose

AutoFS mount and unmount daemon.

Syntax

/usr/sbin/automountd [**-n**] [**-T**] [**-v**] [**-D** *name*=*value*]

Description

The **automountd** daemon is an RPC server that processes and answers requests from the local AutoFS filesystem kernel extension. It uses local files or name service maps to locate file systems to be mounted.

Maps

For a description on map files see the information on Maps in the automount daemon.

Flags

-D*name*=*Value*	Assigns a value to the indicated **automountd** daemon environment variable.
-n	Sets the **nobrowse** option on all maps by default.
-T	Traces RPC server calls, displaying it on standard output.
-v	Displays on standard output verbose status and warning messages.

Related Information

The **df** command, **mount** command, **automount** daemon.

How to Manage NIS automount Maps in *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide* discusses map formatting, multiple mounts, special maps, and the **auto_master/auto.master** NIS configuration map file.

List of NFS commands in *Networks and communication management*.

Network File System (NFS) Overview for System Management in *Networks and communication management*.

---

autopush Command

Purpose

Configures lists of automatically pushed STREAMS modules.

Syntax

autopush -f *File*

autopush -r -M *Major* **-m** *Minor*

autopush -g -M *Major* **-m** *Minor*

Description

The **autopush** command configures the list of modules to be automatically pushed onto the stream when a device is opened. It can also remove a previous setting or obtain information on a setting.

Flags

-f *File* Sets up the **autopush** configuration for each driver according to the information stored in the specified file.

The file specified by the *File* parameter consists of lines consisting of at least four fields per line. Each field is separated by a character space as shown in the following example:

```
maj_ min_ last_min_ mod1 mod2 . . . modn
```

The first three fields are integers that specify the major device number, minor device number, and last minor device number. The subsequent fields represent the names of modules. If the value of the `min_` field is -1, then all minor devices of a major driver specified by the `maj_` field are configured and the value of the `last_min_` field is ignored. If the value of the `last_min_` field is 0, then only a single minor device is configured. To configure a range of minor devices for a particular major, the value of the `min_` field must be less than the value of the `last_min_` field.

The last fields of a line in the **autopush** file represent the list of module names. Each module name is separated by a character space. The maximum number of modules that can be automatically pushed on a stream is eight, and they are pushed onto the stream in the order they are listed. Comment lines start with a # (pound sign).

-r Removes the previous configuration setting of a particular major and minor device number.

-g Obtains the current configuration setting of a particular major and minor device number. It also returns the starting minor device number if the request corresponds to a setting of a range.

-M *Major* Specifies a major device number.

-m *Minor* Specifies a minor device number.

This operating system provides an enhancement to the **autopush** command that makes it easier to specify major numbers. The name of a driver can be specified instead of its major number anywhere the major number is normally used.

Parameters

File Contains at least the major device number, minor device number, last minor device number and modules.

Major Specifies a major device number.

Minor Specifies a minor device number.

Examples

1. To configure a list of automatically pushed Streams modules, type:

   ```
   autopush -f File
   ```

2. To remove the previous configuration, type:

   ```
   autopush -r -M Major -m Minor
   ```

3. To show the current configuration, type:

   ```
   autopush -g -M Major -m Minor
   ```

Related Information

The **streamio** operations.

List of Streams Commands.

STREAMS Overview in *AIX 5L Version 5.3 Communications Programming Concepts*.

awk Command

Purpose
Finds lines in files that match a pattern and performs specified actions on those lines.

Syntax
awk [**-u**] [**-F** *Ere*] [**-v** *Assignment*] ... { **-f** *ProgramFile* | **'***Program***'** } [[*File* ... | *Assignment* ...]] ...

Description
The **awk** command utilizes a set of user-supplied instructions to compare a set of files, one line at a time, to extended regular expressions supplied by the user. Then actions are performed upon any line that matches the extended regular expressions.

The pattern searching of the **awk** command is more general than that of the **grep** command, and it allows the user to perform multiple actions on input text lines. The **awk** command programming language requires no compiling, and allows the user to use variables, numeric functions, string functions, and logical operators.

The **awk** command is affected by the **LANG**, **LC_ALL**, **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES**, **LC_NUMERIC**, **NLSPATH**, and **PATH** environment variables.

The following topics are covered in this article:
- Input for the awk Command
- Output for the awk Command
- File Processing with Records and Fields
- The awk Command Programming Language
  - Patterns
  - Actions
  - Variables
  - Special Variables
- Flags
- Examples

Input for the awk Command
The **awk** command takes two types of input: input text files and program instructions.

Input Text Files
Searching and actions are performed on input text files. The files are specified by:
- Specifying the *File* variable on the command line.
- Modifying the special variables **ARGV** and **ARGC**.
- Providing standard input in the absence of the *File* variable.

If multiple files are specified with the *File* variable, the files are processed in the order specified.

Program Instructions
Instructions provided by the user control the actions of the **awk** command. These instructions come from either the `*Program*' variable on the command line or from a file specified by the **-f** flag together with the *ProgramFile* variable. If multiple program files are specified, the files are concatenated in the order specified and the resultant order of instructions is used.

Output for the awk Command

The **awk** command produces three types of output from the data within the input text file:

- Selected data can be printed to standard output, without alteration to the input file.
- Selected portions of the input file can be altered.
- Selected data can be altered and printed to standard output, with or without altering the contents of the input file.

All of these types of output can be performed on the same file. The programming language recognized by the **awk** command allows the user to redirect output.

File Processing with Records and Fields

Files are processed in the following way:

1. The **awk** command scans its instructions and executes any actions specified to occur before the input file is read.

   The **BEGIN** statement in the **awk** programming language allows the user to specify a set of instructions to be done before the first record is read. This is particularly useful for initializing special variables.
2. One record is read from the input file.

   A record is a set of data separated by a record separator. The default value for the record separator is the new-line character, which makes each line in the file a separate record. The record separator can be changed by setting the **RS** special variable.
3. The record is compared against each pattern specified by the **awk** command's instructions.

   The command instructions can specify that a specific field within the record be compared. By default, fields are separated by white space (blanks or tabs). Each field is referred to by a field variable. The first field in a record is assigned the **$1** variable, the second field is assigned the **$2** variable, and so forth. The entire record is assigned to the **$0** variable. The field separator can be changed by using the **-F** flag on the command line or by setting the **FS** special variable. The **FS** special variable can be set to the values of: blank, single character, or extended regular expression.
4. If the record matches a pattern, any actions associated with that pattern are performed on the record.
5. After the record is compared to each pattern, and all specified actions are performed, the next record is read from input; the process is repeated until all records are read from the input file.
6. If multiple input files have been specified, the next file is then opened and the process repeated until all input files have been read.
7. After the last record in the last file is read, the **awk** command executes any instructions specified to occur after the input processing.

   The **END** statement in the **awk** programming language allows the user to specify actions to be performed after the last record is read. This is particularly useful for sending messages about what work was accomplished by the **awk** command.

The awk Command Programming Language

The **awk** command programming language consists of statements in the form:

Pattern { *Action* }

If a record matches the specified pattern, or contains a field which matches the pattern, the associated action is then performed. A pattern can be specified without an action, in which case the entire line containing the pattern is written to standard output. An action specified without a pattern is performed for every input record.

Patterns

There are four types of patterns used in the **awk** command language syntax:

- Regular Expressions
- Relational Expressions
- Combinations of Patterns
- BEGIN and END Patterns.

Regular Expressions

The extended regular expressions used by the **awk** command are similar to those used by the **grep** or **egrep** command. The simplest form of an extended regular expression is a string of characters enclosed in slashes. For an example, suppose a file named `testfile` had the following contents:

```
smawley, andy
smiley, allen
smith, alan
smithern, harry
smithhern, anne
smitters, alexis
```

Entering the following command line:

```
awk '/smi/' testfile
```

would print to standard output of all records that contained an occurrence of the string `smi`. In this example, the program `'/smi/'` for the **awk** command is a pattern with no action. The output is:

```
smiley, allen
smith, alan
smithern, harry
smithhern, anne
smitters, alexis
```

The following special characters are used to form extended regular expressions:

Character	Function

+	Specifies that a string matches if one or more occurrences of the character or extended regular expression that precedes the **+** (plus) are within the string. The command line:
	`awk '/smith+ern/' testfile`
	prints to standard output any record that contained a string with the characters `smit`, followed by one or more `h` characters, and then ending with the characters `ern`. The output in this example is:
	`smithern, harry`
	`smithhern, anne`

?	Specifies that a string matches if zero or one occurrences of the character or extended regular expression that precedes the ? (question mark) are within the string. The command line:
	`awk '/smith?/' testfile`
	prints to standard output of all records that contain the characters `smit`, followed by zero or one instance of the `h` character. The output in this example is:
	`smith, alan`
	`smithern, harry`
	`smithhern, anne`
	`smitters, alexis`

Character	Function

|	Specifies that a string matches if either of the strings separated by the | (vertical line) are within the string. The command line:
	`awk '/allen`
	`|`
	`alan /' testfile`
	
	prints to standard output of all records that contained the string `allen` or `alan`. The output in this example is:
	`smiley, allen`
	`smith, alan`

( )	Groups strings together in regular expressions. The command line:
	`awk '/a(ll)?(nn)?e/' testfile`
	
	prints to standard output of all records with the string `ae` or `alle` or `anne` or `allnne`. The output in this example is:
	`smiley, allen`
	`smithhern, anne`

{*m*}	Specifies that a string matches if exactly *m* occurrences of the pattern are within the string. The command line:
	`awk '/l{2}/' testfile`
	
	prints to standard output
	`smiley, allen`

{*m,*}	Specifies that a string matches if at least *m* occurrences of the pattern are within the string. The command line:
	`awk '/t{2,}/' testfile`
	
	prints to standard output:
	`smitters, alexis`

{*m, n*}	Specifies that a string matches if between *m* and *n*, inclusive, occurrences of the pattern are within the string (where *m* <= *n*). The command line:
	`awk '/er{1, 2}/' testfile`
	
	prints to standard output:
	`smithern, harry`
	`smithern, anne`
	`smitters, alexis`

[*String*]	Signifies that the regular expression matches any characters specified by the *String* variable within the square brackets. The command line:
	`awk '/sm[a-h]/' testfile`
	
	prints to standard output of all records with the characters `sm` followed by any character in alphabetical order from `a` to `h`. The output in this example is:
	`smawley, andy`

[^ *String*]	A ^ (caret) within the [ ] (square brackets) and at the beginning of the specified string indicates that the regular expression *does not* match any characters within the square brackets. Thus, the command line:
	`awk '/sm[^a-h]/' testfile`
	
	prints to standard output:
	`smiley, allen`
	`smith, alan`
	`smithern, harry`
	`smithhern, anne`
	`smitters, alexis`

Character	Function

~,!~	Signifies a conditional statement that a specified variable matches (tilde) or does not match (tilde, exclamation point) the regular expression. The command line:
	`awk '$1 ~ /n/' testfile`
	prints to standard output of all records whose first field contained the character `n`. The output in this example is:
	`smithern, harry`
	`smithhern, anne`

^	Signifies the beginning of a field or record. The command line:
	`awk '$2 ~ /^h/' testfile`
	prints to standard output of all records with the character `h` as the first character of the second field. The output in this example is:
	`smithern, harry`

$	Signifies the end of a field or record. The command line:
	`awk '$2 ~ /y$/' testfile`
	prints to standard output of all records with the character `y` as the last character of the second field. The output in this example is:
	`smawley, andy`
	`smithern, harry`

. (period)	Signifies any one character except the terminal new-line character at the end of a space. The command line:
	`awk '/a..e/' testfile`
	prints to standard output of all records with the characters `a` and e separated by two characters. The output in this example is:
	`smawley, andy`
	`smiley, allen`
	`smithhern, anne`

***(asterisk)	Signifies zero or more of any characters. The command line:
	`awk '/a.*e/' testfile`
	prints to standard output of all records with the characters `a` and e separated by zero or more characters. The output in this example is:
	`smawley, andy`
	`smiley, allen`
	`smithhern, anne`
	`smitters, alexis`

**** (backslash)	The escape character. When preceding any of the characters that have special meaning in extended regular expressions, the escape character removes any special meaning for the character. For example, the command line: `/a\/\//` would match the pattern a //, since the backslashes negate the usual meaning of the slash as a delimiter of the regular expression. To specify the backslash itself as a character, use a double backslash. See the following item on escape sequences for more information on the backslash and its uses.

Recognized Escape Sequences: The **awk** command recognizes most of the escape sequences used in C language conventions, as well as several that are used as special characters by the **awk** command itself. The escape sequences are:

Escape Sequence	Character Represented
\″	\″ (double-quotation) mark
\/	/ (slash) character

Escape Sequence	Character Represented
ddd	Character whose encoding is represented by a one-, two- or three-digit octal integer, where *d* represents an octal digit
\\	\ (backslash) character
\a	Alert character
\b	Backspace character
\f	Form-feed character
\n	New-line character (see following note)
\r	Carriage-return character
\t	Tab character
\v	Vertical tab.

Note: Except in the **gsub**, **match**, **split**, and **sub** built-in functions, the matching of extended regular expressions is based on input records. Record-separator characters (the new-line character by default) cannot be embedded in the expression, and no expression matches the record-separator character. If the record separator is not the new-line character, then the new-line character can be matched. In the four built-in functions specified, matching is based on text strings, and any character (including the record separator) can be embedded in the pattern so that the pattern matches the appropriate character. However, in all regular-expression matching with the **awk** command, the use of one or more NULL characters in the pattern produces undefined results.

Relational Expressions

The relational operators < (less than), > (greater than), <= (less than or equal to), >= (greater than or equal to), = = (equal to), and ! = (not equal to) can be used to form patterns. For example, the pattern:

```
$1 < $4
```

matches records where the first field is less than the fourth field. The relational operators also work with string values. For example:

```
$1 =! "q"
```

matches all records where the first field is not a `q`. String values can also be matched on collation values. For example:

```
$1 >= "d"
```

matches all records where the first field starts with a character that is `a`, `b`, `c`, or `d`. If no other information is given, field variables are compared as string values.

Combinations of Patterns

Patterns can be combined using three options:

- Ranges are specified by two patterns separated with a , (comma). Actions are performed on every record starting with the record that matches the first pattern, and continuing through and including the record that matches the second pattern. For example:

  ```
  /begin/,/end/
  ```

  matches the record containing the string `begin`, and every record between it and the record containing the string `end`, including the record containing the string `end`.
- Parentheses ( ) group patterns together.
- The boolean operators || (or), && (and), and ! (not) combine patterns into expressions that match if they evaluate true, otherwise they do not match. For example, the pattern:

  ```
  $1 == "al" && $2 == "123"
  ```

  matches records where the first field is `al` and the second field is `123`.

BEGIN and END Patterns

Actions specified with the **BEGIN** pattern are performed before any input is read. Actions specified with the **END** pattern are performed after all input has been read. Multiple **BEGIN** and **END** patterns are allowed and processed in the order specified. An **END** pattern can precede a **BEGIN** pattern within the program statements. If a program consists only of **BEGIN** statements, the actions are performed and no input is read. If a program consists only of **END** statements, all the input is read prior to any actions being taken.

Actions

There are several types of action statements:

- Action Statements
- Built-in Functions
- User-Defined Functions
- Conditional Statements
- Output Actions

Action Statements

Action statements are enclosed in { } (braces). If the statements are specified without a pattern, they are performed on every record. Multiple actions can be specified within the braces, but must be separated by new-line characters or ; (semicolons), and the statements are processed in the order they appear. Action statements include:

Arithmetical Statements

The mathematical operators + (plus), - (minus), / (division), ^ (exponentiation), * (multiplication), % (modulus) are used in the form:

```
Expression Operator Expression
```

Thus, the statement:

```
$2 = $1 ^ 3
```

assigns the value of the first field raised to the third power to the second field.

Unary Statements

The unary **-** (minus) and unary **+** (plus) operate as in the C programming language:

```
+Expression or -Expression
```

Increment and Decrement Statements

The pre-increment and pre-decrement statements operate as in the C programming language:

```
++Variable or --Variable
```

The post-increment and post-decrement statements operate as in the C programming language:

```
Variable++ or Variable--
```

Assignment Statements

The assignment operators += (addition), -= (subtraction), /= (division), and *= (multiplication) operate as in the C programming language, with the form:

```
Variable += Expression
Variable -= Expression
Variable /= Expression
Variable *= Expression
```

For example, the statement:

```
$1 *= $2
```

multiplies the field variable **$1** by the field variable **$2** and then assigns the new value to **$1**.

The assignment operators ^= (exponentiation) and %= (modulus) have the form:

```
Variable1^=Expression1
```

AND

```
Variable2%=Expression2
```

and they are equivalent to the C programming language statements:

```
Variable1=pow(Variable1, Expression1)
```

AND

```
Variable2=fmod(Variable2, Expression2)
```

where `pow` is the **pow** subroutine and `fmod` is the **fmod** subroutine.

String Concatenation Statements

String values can be concatenated by stating them side by side. For example:

```
$3 = $1 $2
```

assigns the concatenation of the strings in the field variables **$1** and **$2** to the field variable **$3**.

Built-In Functions

The **awk** command language uses arithmetic functions, string functions, and general functions. The close Subroutine statement is necessary if you intend to write a file, then read it later in the same program.

Arithmetic Functions: The following arithmetic functions perform the same actions as the C language subroutines by the same name:

atan2(*y*, *x* **)**	Returns arctangent of *y*/*x*.
cos(*x* **)**	Returns cosine of *x*; *x* is in radians.
sin(*x* **)**	Returns sin of *x*; *x* is in radians.
exp(*x* **)**	Returns the exponential function of *x*.
log(*x* **)**	Returns the natural logarithm of *x*.
sqrt(*x* **)**	Returns the square root of *x*.
int(*x* **)**	Returns the value of *x* truncated to an integer.
rand( )	Returns a random number *n*, with 0 <= *n* < 1.
srand([*Expr*] **)**	Sets the seed value for the **rand** function to the value of the *Expr* parameter, or use the time of day if the *Expr* parameter is omitted. The previous seed value is returned.

String Functions: The string functions are:

gsub(*Ere*, *Repl*, [*In*] **)**	Performs exactly as the **sub** function, except that all occurrences of the regular expression are replaced.

sub(*Ere*, *Repl*, [*In*] **)**	Replaces the first occurrence of the extended regular expression specified by the *Ere* parameter in the string specified by the *In* parameter with the string specified by the *Repl* parameter. The **sub** function returns the number of substitutions. An & (ampersand) appearing in the string specified by the *Repl* parameter is replaced by the string in the *In* parameter that matches the extended regular expression specified by the *Ere* parameter. If no *In* parameter is specified, the default value is the entire record (the **$0** record variable).
index(*String1*, *String2* **)**	Returns the position, numbering from 1, within the string specified by the *String1* parameter where the string specified by the *String2* parameter occurs. If the *String2* parameter does not occur in the *String1* parameter, a 0 (zero) is returned.
length [(*String***)]**	Returns the length, in characters, of the string specified by the *String* parameter. If no *String* parameter is given, the length of the entire record (the **$0** record variable) is returned.
blength [(*String***)]**	Returns the length, in bytes, of the string specified by the *String* parameter. If no *String* parameter is given, the length of the entire record (the **$0** record variable) is returned.
substr(*String*, *M*, [*N*] **)**	Returns a substring with the number of characters specified by the *N* parameter. The substring is taken from the string specified by the *String* parameter, starting with the character in the position specified by the *M* parameter. The *M* parameter is specified with the first character in the *String* parameter as number 1. If the *N* parameter is not specified, the length of the substring will be from the position specified by the *M* parameter until the end of the *String* parameter.
match(*String*, *Ere* **)**	Returns the position, in characters, numbering from 1, in the string specified by the *String* parameter where the extended regular expression specified by the *Ere* parameter occurs, or else returns a 0 (zero) if the *Ere* parameter does not occur. The **RSTART** special variable is set to the return value. The **RLENGTH** special variable is set to the length of the matched string, or to -1 (negative one) if no match is found.
split(*String*, *A*, [*Ere*] **)**	Splits the string specified by the *String* parameter into array elements *A*[1], *A*[2], . . ., *A*[*n*], and returns the value of the *n* variable. The separation is done with the extended regular expression specified by the *Ere* parameter or with the current field separator (the **FS** special variable) if the *Ere* parameter is not given. The elements in the *A* array are created with string values, unless context indicates a particular element should also have a numeric value.

tolower(*String* **)** Returns the string specified by the *String* parameter, with each uppercase character in the string changed to lowercase. The uppercase and lowercase mapping is defined by the **LC_CTYPE** category of the current locale.

toupper(*String* **)** Returns the string specified by the *String* parameter, with each lowercase character in the string changed to uppercase. The uppercase and lowercase mapping is defined by the **LC_CTYPE** category of the current locale.

sprintf(*Format***,** *Expr***,** *Expr***,** . . . **)** Formats the expressions specified by the *Expr* parameters according to the **printf** subroutine format string specified by the *Format* parameter and returns the resulting string.

General Functions: The general functions are:

close(*Expression* **)** Close the file or pipe opened by a **print** or **printf** statement or a call to the **getline** function with the same string-valued *Expression* parameter. If the file or pipe is successfully closed, a 0 is returned; otherwise a non-zero value is returned. The **close** statement is necessary if you intend to write a file, then read the file later in the same program.

system(*Command* **)** Executes the command specified by the *Command* parameter and returns its exit status. Equivalent to the **system** subroutine.

Expression | **getline** [*Variable*] Reads a record of input from a stream piped from the output of a command specified by the *Expression* parameter and assigns the value of the record to the variable specified by the *Variable* parameter. The stream is created if no stream is currently open with the value of the *Expression* parameter as its command name. The stream created is equivalent to one created by a call to the **popen** subroutine with the *Command* parameter taking the value of the *Expression* parameter and the *Mode* parameter set to a value of **r**. Each subsequent call to the **getline** function reads another record, as long as the stream remains open and the *Expression* parameter evaluates to the same string. If a *Variable* parameter is not specified, the **$0** record variable and the **NF** special variable are set to the record read from the stream.

getline [*Variable*] **<** *Expression* Reads the next record of input from the file named by the *Expression* parameter and sets the variable specified by the *Variable* parameter to the value of the record. Each subsequent call to the **getline** function reads another record, as long as the stream remains open and the *Expression* parameter evaluates to the same string. If a *Variable* parameter is not specified, the **$0** record variable and the **NF** special variable are set to the record read from the stream.

getline [*Variable*] — Sets the variable specified by the *Variable* parameter to the next record of input from the current input file. If no *Variable* parameter is specified, **$0** record variable is set to the value of the record, and the **NF**, **NR**, and **FNR** special variables are also set.

Note: All forms of the **getline** function return 1 for successful input, zero for end of file, and -1 for an error.

User-Defined Functions

User-defined functions are declared in the following form:

```
function Name (Parameter, Parameter,...)  { Statements }
```

A function can be referred to anywhere in an **awk** command program, and its use can precede its definition. The scope of the function is global.

Function parameters can be either scalars or arrays. Parameter names are local to the function; all other variable names are global. The same name should not be used for different entities; for example, a parameter name should not be duplicated as a function name, or special variable. Variables with global scope should not share the name of a function. Scalars and arrays should not have the same name in the same scope.

The number of parameters in the function definition does not have to match the number of parameters used when the function is called. Excess formal parameters can be used as local variables. Extra scalar parameters are initialized with a string value equivalent to the empty string and a numeric value of 0 (zero); extra array parameters are initialized as empty arrays.

When invoking a function, no white space is placed between the function name and the opening parenthesis. Function calls can be nested and recursive. Upon return from any nested or recursive function call, the values of all the calling function's parameters shall be unchanged, except for array parameters passed by reference. The **return** statement can be used to return a value.

Within a function definition, the new-line characters are optional before the opening { (brace) and after the closing } (brace).

An example of a function definition is:

```
function average ( g,n)
  {
        for (i in g)
           sum=sum+g[i]
        avg=sum/n
        return avg
  }
```

The function average is passed an array, `g`, and a variable, `n`, with the number of elements in the array. The function then obtains an average and returns it.

Conditional Statements

Most conditional statements in the **awk** command programming language have the same syntax and function as conditional statements in the C programming language. All of the conditional statements allow the use of { } (braces) to group together statements. An optional new-line can be used between the expression portion and the statement portion of the conditional statement, and new-lines or ; (semicolon) are used to separate multiple statements in { } (braces). Six conditional statements in C language are:

if Requires the following syntax:

if (*Expression*) { *Statement }* [**else** *Action*]

while Requires the following syntax:

while (*Expression*) { *Statement* }

for Requires the following syntax:

for (*Expression* ; *Expression* ; *Expression*) { *Statement* }

break Causes the program loop to be exited when the **break** statement is used in either a **while** or **for** statement.

continue Causes the program loop to move to the next iteration when the **continue** statement is used in either a **while** or **for** statement.

Five conditional statements in the **awk** command programming language that do not follow C-language rules are:

for...in Requires the following syntax:

for (*Variable* **in** *Array*) { *Statement* }

The **for**...**in** statement sets the *Variable* parameter to each index value of the *Array* variable, one index at a time and in no particular order, and performs the action specified by the *Statement* parameter with each iteration. See the **delete** statement for an example of a **for**...**in** statement.

if...in Requires the following syntax:

if (*Variable* **in** *Array*) { *Statement* }

The **if**...**in** statement searches for the existence of the *Array* element. The statement is performed if the *Array* element is found.

delete Requires the following syntax:

delete *Array* [*Expression*]

The **delete** statement deletes both the array element specified by the *Array* parameter and the index specified by the *Expression* parameter. For example, the statements:

```
for (i in g)
   delete g[i];
```

would delete every element of the `g[]` array.

exit Requires the following syntax:

exit [*Expression*]

The **exit** statement first invokes all **END** actions in the order they occur, then terminates the **awk** command with an exit status specified by the *Expression* parameter. No subsequent **END** actions are invoked if the **exit** statement occurs within an **END** action.

Requires the following syntax:

*Comment*

The **#** statement places comments. Comments should always end with a new-line but can begin anywhere on a line.

next Stops the processing of the current input record and proceeds with the next input record.

Output Statements

Two output statements in the **awk** command programming language are:

print Requires the following syntax:

print [*ExpressionList*] [*Redirection*] [*Expression*]

The **print** statement writes the value of each expression specified by the *ExpressionList* parameter to standard output. Each expression is separated by the current value of the **OFS** special variable, and each record is terminated by the current value of the **ORS** special variable.

The output can be redirected using the *Redirection* parameter, which can specify the three output redirections with the > (greater than), >> (double greater than), and the | (pipe). The *Redirection* parameter specifies how the output is redirected, and the *Expression* parameter is either a path name to a file (when *Redirection* parameter is > or >>) or the name of a command (when the *Redirection* parameter is a |).

printf Requires the following syntax:

printf *Format* [, *ExpressionList*] [*Redirection*] [*Expression*]

The **printf** statement writes to standard output the expressions specified by the *ExpressionList* parameter in the format specified by the *Format* parameter. The **printf** statement functions exactly like the **printf** command, except for the c conversion specification (%c). The *Redirection* and *Expression* parameters function the same as in the **print** statement.

For the c conversion specification: if the argument has a numeric value, the character whose encoding is that value will be output. If the value is zero or is not the encoding of any character in the character set, the behavior is undefined. If the argument does not have a numeric value, the first character of the string value will be output; if the string does not contain any characters the behavior is undefined.

Note: If the *Expression* parameter specifies a path name for the *Redirection* parameter, the *Expression* parameter should be enclosed in double quotes to insure that it is treated as a string.

Variables

Variables can be scalars, field variables, arrays, or special variables. Variable names cannot begin with a digit.

Variables can be used just by referencing them. With the exception of function parameters, they are not explicitly declared. Uninitialized scalar variables and array elements have both a numeric value of 0 (zero) and a string value of the null string (″ ″).

Variables take on numeric or string values according to context. Each variable can have a numeric value, a string value, or both. For example:

```
x = "4" + "8"
```

assigns the value of 12 to the variable x. For string constants, expressions should be enclosed in ″ ″ (double quotation) marks.

There are no explicit conversions between numbers and strings. To force an expression to be treated as a number, add 0 (zero) to it. To force an expression to be treated as a string, append a null string (″ ″).

Field Variables

Field variables are designated by a **$** (dollar sign) followed by a number or numerical expression. The first field in a record is assigned the **$1** variable, the second field is assigned to the **$2** variable, and so forth. The **$0** field variable is assigned to the entire record. New field variables can be created by assigning a value to them. Assigning a value to a non-existent field, that is, any field larger than the current value of **$NF** field variable, forces the creation of any intervening fields (set to the null string), increases the value of the **NF** special variable, and forces the value of **$0** record variable to be recalculated. The new fields

are separated by the current field separator (which is the value of the **FS** special variable). Blanks and tabs are the default field separators. To change the field separator, use the **-F** flag, or assign the **FS** special variable a different value in the **awk** command program.

Arrays

Arrays are initially empty and their sizes change dynamically. Arrays are represented by a variable with subscripts in [ ] (square brackets). The subscripts, or element identifiers, can be numbers of strings, which provide a type of associative array capability. For example, the program:

```
/red/  { x["red"]++ }
/green/ { y["green"]++ }
```

increments counts for both the `red` counter and the `green` counter.

Arrays can be indexed with more than one subscript, similar to multidimensional arrays in some programming languages. Because programming arrays for the **awk** command are really one dimensional, the comma-separated subscripts are converted to a single string by concatenating the string values of the separate expressions, with each expression separated by the value of the **SUBSEP** environmental variable. Therefore, the following two index operations are equivalent:

```
x[expr1, expr2,...exprn]
```

AND

```
x[expr1SUBSEPexpr2SUBSEP...SUBSEPexprn]
```

When using the **in** operator, a multidimensional *Index* value should be contained within parentheses. Except for the **in** operator, any reference to a nonexistent array element automatically creates that element.

Special Variables

The following variables have special meaning for the **awk** command:

ARGC The number of elements in the **ARGV** array. This value can be altered.

ARGV The array with each member containing one of the *File* variables or *Assignment* variables, taken in order from the command line, and numbered from 0 (zero) to **ARGC** -1. As each input file is finished, the next member of the **ARGV** array provides the name of the next input file, unless:

- The next member is an *Assignment* statement, in which case the assignment is evaluated.
- The next member has a null value, in which case the member is skipped. Programs can skip selected input files by setting the member of the **ARGV** array that contains that input file to a null value.
- The next member is the current value of **ARGV** [**ARGC** -1], which the **awk** command interprets as the end of the input files.

CONVFMT The **printf** format for converting numbers to strings (except for output statements, where the **OFMT** special variable is used). The default is ″%.6g″.

ENVIRON An array representing the environment under which the **awk** command operates. Each element of the array is of the form:

ENVIRON [″*Environment VariableName*″ **]** = *EnvironmentVariableValue*

The values are set when the **awk** command begins execution, and that environment is used until the end of execution, regardless of any modification of the **ENVIRON** special variable.

FILENAME The path name of the current input file. During the execution of a **BEGIN** action, the value of **FILENAME** is undefined. During the execution of an **END** action, the value is the name of the last input file processed.

FNR The number of the current input record in the current file.

FS The input field separator. The default value is a blank. If the input field separator is a blank, any number of locale-defined spaces can separate fields. The **FS** special variable can take two additional values:

- With **FS** set to a single character, fields are separated by each single occurrence of the character.
- With **FS** set to an extended regular expression, each occurrence of a sequence matching the extended regular expression separates fields.

NF The number of fields in the current record, with a limit of 99. Inside a **BEGIN** action, the **NF** special variable is undefined unless a **getline** function without a *Variable* parameter has been issued previously. Inside an **END** action, the **NF** special variable retains the value it had for the last record read, unless a subsequent, redirected, **getline** function without a *Variable* parameter is issued prior to entering the **END** action.

NR The number of the current input record. Inside a **BEGIN** action the value of the **NR** special variable is 0 (zero). Inside an **END** action, the value is the number of the last record processed.

OFMT The **printf** format for converting numbers to strings in output statements. The default is ″% .6g″.

OFS The output field separator (default is a space).

ORS The output record separator (default is a new-line character).

RLENGTH The length of the string matched by the **match** function.

RS Input record separator (default is a new-line character). If the **RS** special variable is null, records are separated by sequences of one or more blank lines; leading or trailing blank lines do not result in empty records at the beginning or end of input; and the new-line character is always a field separator, regardless of the value of the **FS** special variable.

RSTART The starting position of the string matched by the **match** function, numbering from 1. Equivalent to the return value of the **match** function.

SUBSEP Separates multiple subscripts. The default is \031.

Flags

-f *ProgramFile* Obtains instructions for the **awk** command from the file specified by the *ProgramFile* variable. If the **-f** flag is specified multiple times, the concatenation of the files, in the order specified, will be used as the set of instructions.

-u Displays the output in an unbuffered mode. If this flag is used, the **awk** command does not buffer the output. Instead, it displays the output instantaneously. By default, the **awk** command displays the output in a buffered mode.

-F *Ere* Uses the extended regular expression specified by the *Ere* variable as the field separator. The default field separator is a blank.

-v *Assignment* Assigns a value to a variable for the **awk** command's programming language. The *Assignment* parameter is in the form of *Name = Value*. The *Name* portion specifies the name of the variable and can be any combination of underscores, digits, and alphabetic characters, but it must start with either an alphabetic character or an underscore. The *Value* portion is also composed of underscores, digits, and alphabetic characters, and is treated as if it were preceded and followed by a ″ (double-quotation character, similar to a string value). If the *Value* portion is numeric, the variable will also be assigned the numeric value.

The assignment specified by the **-v** flag occurs before any portion of the **awk** command's program is executed, including the **BEGIN** section.

Assignment Assigns a value to a variable for the **awk** command's programming language. It has the same form and function as the *Assignment* variable with the **-v** flag, except for the time each is processed. The *Assignment* parameter is processed just prior to the input file (specified by the *File* variable) that follows it on the command line. If the *Assignment* parameter is specified just prior to the first of multiple input files, the assignments are processed just after the **BEGIN** sections (if any). If an *Assignment* parameter occurs after the last file, the assignment is processed before the **END** sections (if any). If no input files are specified, the assignments are processed the standard input is read.

File Specifies the name of the file that contains the input for processing. If no *File* variable is specified, or if a **-** (minus) sign is specified, standard input is processed.

'*Program*' Contains the instructions for the **awk** command. If the **-f** flag is not specified, the *Program* variable should be the first item on the command line. It should be bracketed by ' ' (single quotes).

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

You can alter the exit status within the program by using the **exit** [*Expression*] conditional statement.

Examples

1. To display the lines of a file that are longer than 72 characters, enter:

   ```
   awk  'length  >72'  chapter1
   ```

   This selects each line of the `chapter1` file that is longer than 72 characters and writes these lines to standard output, because no *Action* is specified. A tab character is counted as 1 byte.
2. To display all lines between the words `start` and `stop`, including `"start"` and `"stop"`, enter:

   ```
   awk  '/start/,/stop/'  chapter1
   ```
3. To run an **awk** command program, `sum2.awk`, that processes the file, `chapter1`, enter:

   ```
   awk  -f  sum2.awk  chapter1
   ```

   The following program, `sum2.awk`, computes the sum and average of the numbers in the second column of the input file, `chapter1`:

   ```
       {
          sum += $2
       }
   END {
          print "Sum: ", sum;
          print "Average:", sum/NR;
       }
   ```

   The first action adds the value of the second field of each line to the variable `sum`. All variables are initialized to the numeric value of 0 (zero) when first referenced. The pattern **END** before the second action causes those actions to be performed after all of the input file has been read. The **NR** special variable, which is used to calculate the average, is a special variable specifying the number of records that have been read.
4. To print the first two fields in opposite order, enter:

   ```
   awk '{ print $2, $1 }' chapter1
   ```
5. The following **awk** program

   ```
   awk -f sum3.awk chapter2
   ```

   prints the first two fields of the file `chapter2` with input fields separated by comma and/or blanks and tabs, and then adds up the first column, and prints the sum and average:

   ```
   BEGIN  {FS = ",|[ \t]+"}
          {print $1, $2}
          {s += $1}
   END    {print "sum is",s,"average is", s/NR }
   ```

Related Information

The **egrep** command, **fgrep** command, **grep** command, **lex** command, **printf** command, **sed** command.

The **popen** subroutine, **printf** subroutine, **system** subroutine.

---

back Command

Purpose

Starts the backgammon game.

Syntax

back

Description

The **back** command provides you with a partner for backgammon. You select one of the following three skill levels: beginner, intermediate, or expert. You can choose to roll your own dice during your turns, and you are asked if you want to move first.

Important locations on the computer-generated board are:

- 0 is the bar for removed white pieces.
- 1 is white's extreme inner table.
- 24 is brown's extreme inner table.
- 25 is the bar for removed brown pieces.

For details on how to make your moves, enter `Y` when prompted for `Instructions?` at the beginning of the game. During play, you are prompted for `move?`. Either enter a numerical move or press ? (question mark) key for a list of move choices.

When the game is finished, you are asked if you want to save game information. Entering `Y` stores game data in the **back.log** file in your current directory.

The **back** command plays only the forward game, even at the expert level. It objects if you try to make too many moves in a turn, but not if you make too few. Doubling is not permitted.

To quit the game, press the Interrupt (Ctrl-C) key sequence.

Files

/usr/games	Location of the system's games.
/usr/games/lib/backrules	Location of the rules file.
/tmp/b*	Location of the log temp file.
back.log	Contains data from previously played games.

Related Information

The **arithmetic** command, **bj** command, **craps** command, **fish** command, **fortune** command, **hangman** command, **moo** command, **number** command, **quiz** command, **ttt** command, **turnoff** command, **turnon** command, **wump** command.

backsnap Command

Purpose

Provides an interface to create a snapshot for a JFS2 file system and perform a backup of the snapshot.

Syntax

backsnap [**-R**] { **-m** *MountPoint* **-s size**=*Size* | **-n** *snapshotName* } [*BackupOptions*] *FileSystem*

Description

Provides an interface to create a snapshot for a JFS2 file system and perform a backup of the snapshot. The **restore** command can be used to retrieve the backup.

Flags

-m *MountPoint*	Specifies the path of where the external snapshot created should be mounted.
-R	Specifies that the snapshot created by this command will be removed when the backup completes.
-s size=*Size*	Specifies the size to create the new logical volume for the external snapshot. If *Size* is followed by an M, the value is treated as megabytes. If *Size* is followed by a G, the value is treated as gigabytes. Otherwise, the value is treated as 512-byte blocks.
-n *snapshotName*	Specifies the name of the internal snapshot to be created. The JFS2 file system must be enabled to use internal snapshots.

Parameters

BackupOptions	Any other options are passed to the **backup** command when the backup of the snapshot is performed. Minimally, it is required to specify the type of backup desired. For backup by name, the **-i** option must be specified along with the device for the backup. For **backup** by inode, the level option, -[0-9], must be specified along with the device for the backup. Use the **restore** command to retrieve the backup.
FileSystem	Specifies the JFS2 file system to create a snapshot of and backup.

Exit Status

0 The command completed successfully.

>0 An error occurred.

Examples

1. To create a snapshot for the **/home/janet/sb** file system and perform a backup on the snapshot by name, enter:

```
backsnap -m /tmp/snapshot/janetsb -s size=16M -i -f/dev/rmt0 /home/janet/sb
```

This command creates a logical volume of size 16 megabytes and then creates a snapshot for the **/home/janet/sb** file system on the newly created logical volume. It then mounts the snapshot on **/tmp/snapshot/janetsb** and backs up the files and directories in that file system by name to the **/dev/rmt0** device.

2. To create a snapshot for the **/home/janet/sb** file system and perform a backup on the snapshot by inode, enter:

```
backsnap -R -m /tmp/snapshot/janetsb -s size=16M -0 -f /dev/rmt0 /home/janet/sb
```

This command creates a logical volume of size 16 megabytes and then creates a snapshot for the **/home/janet/sb** file system on the newly created logical volume. It then mounts the snapshot on **/tmp/snapshot/janetsb** and backs up the data in the snapshot by inode to the **/dev/rmt0** device. After the backup completes, the snapshot is deleted.

Files

/usr/sbin/backsnap	Contains the **backsnap** command.

Related Information

The **backup** command, **restore** command, **snapshot** command.

---

backup Command

Purpose

Backs up files and file systems.

Syntax

To Back Up Files by Name

backup -i [**-b** *Number*] [**-p** [**-e** *RegularExpression*]] [**-E**{**force**|**ignore**|**warn**}] [**-f** *Device*] [**-I** *Number*] [**-U**] [**-O**] [**-o**] [**-q**] [**-v**] [**-Z**]

To Back Up File Systems by i-node

backup [[**-***Level*] [**-b** *Number*] [**-c**] [**-f** *Device*] [**-L** *Length*] [**-n** *snapshotName*] [**-U**] [**-O**] [**-u**]] [*FileSystem*] | [**-w** | **-W**] [**-Z**]

Description

The **backup** command creates copies of your files on a backup medium, such as a magnetic tape or diskette. The copies are in one of the two backup formats:

- Specific files backed up by name using the **-i** flag.
- Entire file system backed up by i-node using the *Level* and *FileSystem* parameters.

If you issue the **backup** command without any parameters, it defaults to a level 9 i-node backup of the root file system to the **/dev/rfd0** device. The default syntax is:

```
-9uf/dev/rfd0 /dev/rhd4
```

The default backup device is **/dev/rfd0**. If flags are specified that are not appropriate for the specified backup device, the **backup** command displays an error message and continues with the backup.

A single backup can span multiple volumes.

Notes:

1. Running the **backup** command results in the loss of all material previously stored on the selected output medium.
2. Data integrity of the archive may be compromised if a file is modified during system backup. Keep system activity at a minimum during the system backup procedure.
3. If a backup is made to a tape device with the device block size set to 0, it might be difficult to restore data from the tape unless the default write size was used with the **backup** command. The default write size for the **backup** command can be read by the **restore** command when the tape device block size is 0.

   In other words, the **-b** flag should not be specified when the tape device block size is 0. If the **-b** flag of the **backup** command is specified and is different from the default size, the same size must be specified with the **-b** flag of the **restore** command when the archived files are restored from the tape.
4. Do not attempt to back up a logical volume.

Backing Up Files by Name

To back up by name, use the **-i** flag. The **backup** command reads standard input for the names of the files to be backed up.

File types can be special files, regular files, or directories. When the file type is a directory, only the directory is backed up. The files under the directory are not backed up, unless they are explicitly specified.

Notes:

1. Files are restored using the same path names as the archived files. Therefore, to create a backup that can be restored from any path, use full path names for the files that you want to back up.
2. When backing up files that require multiple volumes, do not enter the list of file names from the keyboard. Instead, pipe or redirect the list from a file to the **backup** command.

   When you enter the file names from the keyboard and the backup process needs a new tape or diskette, the command ″loses″ any file names already entered but not yet backed up. To avoid this problem, enter each file name only after the archived message for the previous file has been displayed. The archived message consists of the character `a` followed by the file name.
3. If you specify the **-p** flag, only files of less than 2GB are packed.

Backing Up File Systems by i-node

To back up a file system by i-node, specify the **-***Level* and *FileSystem* parameters. When used in conjunction with the **-u** flag, the **-***Level* parameter provides a method of maintaining a hierarchy of incremental backups for each file system. Specify the **-u** flag and set the **-***Level* parameter to `n` to back up only those files that have been modified since the `n-1` level backup. Information regarding the date, time, and level of each incremental backup is written to the **/etc/dumpdates** file. The possible backup levels are 0 to 9. A level 0 backup archives all files in the file system. If the **/etc/dumpdates** file contains no backup information for a particular file system, specifying any level causes all files in that file system to be archived.

The *FileSystem* parameter can specify either the physical device name (block or raw name) or the name of the directory on which the file system is mounted. The default file system is the root (**/**) file system.

Users must have read access to the file system device (such as **/dev/hd4**) or have Backup authorization in order to perform backups by i_node.

Notes:

1. You must first unmount a file system before backing it up by i-node. If you attempt to back up a mounted file system, a warning message is displayed. The **backup** command continues, but the created backup may contain inconsistencies because of changes that may have occurred in the file system during the backup operation.
2. Backing up file systems by i-node truncates the **uid** or **gid** of files having a **uid** or **gid** greater than 65535. When restored, these files may have different values for the **uid** and **gid** attributes. To retain the values correctly, always back up by name files having a **uid** or **gid** greater than 65535.
3. You can archive only JFS (Journaled File System) file systems when backing up by i-node. Back up any non-JFS file systems by file name or by using other archive commands, such as the **pax**, **tar**, or **cpio** command.
4. The **-Z** flag is mandatory for backing up encrypted file systems.

Flags

-b *Number*
For backups by name, specifies the number of 512-byte blocks; for backups by i-node, specifies the number of 1024-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 and 32 for backups by i-node.

The write size is the number of blocks multiplied by the block size. The default write size for the **backup** command writing to tape devices is 51200 (100 * 512) for backups by name and 32768 (32 * 1024) for backups by i-node. The write size must be an even multiple of the tape's physical block size.

The value of the **-b** flag is always ignored when the **backup** command writes to diskette. In this case, the command always writes in clusters that occupy a complete track.

-c
Specifies that the tape is a cartridge, not a nine-track.

-e *RegularExpression*
Specifies that the files with names matching the regular expression are not to be packed. A regular expression is a set of characters, meta characters, and operators that define a string or group of strings in a search pattern. It can also be a string containing wildcard characters and operations that define a set of one or more possible strings. The **-e** flag is applied only when the **-p** flag is specified.

-E
For backups by name, the **-E** option requires one of the following arguments. If you omit the **-E** option, **warn** is the default behavior.

force Fails the backup 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 *Device*

Specifies the output device. To send output to a named device, specify the *Device* variable as a path name (such as `/dev/rmt0`). To send output to the standard output device, specify a - (minus sign). The - (minus) feature enables you to pipe the output of the **backup** command to the **dd** command.

You can also specify a range of archive devices. The range specification must be in the following format:

`/dev/deviceXXX-YYY`

where *XXX* and *YYY* are whole numbers, and *XXX* must always be less than *YYY*; for example, `/dev/rfd0-3`.

All devices in the specified range must be of the same type. For example, you can use a set of 8mm, 2.3GB tapes or a set of 1.44MB diskettes. All tape devices must be set to the same physical tape block size.

If the *Device* variable specifies a range, the **backup** command automatically goes from one device in the range to the next. After exhausting all of the specified devices, the **backup** command halts and requests that new volumes be mounted on the range of devices.

-i

Specifies that files be read from standard input and archived by file name. If relative path names are used, files are restored (with the **restore** command) relative to the current directory at restore time. If full path names are used, files are restored to those same names.

-l *Number*

(lowercase L) Limits the total number of blocks to use on the diskette device. The value specified must be a non-zero multiple of the number of sectors per diskette track. This option applies to by-name backups only. See the format command for information about sectors per diskette track.

-L *Length*

Specifies the length of the tape in bytes. This flag overrides the **-c**, **-d**, and **-s** flags. You can specify the size with a suffix of b, k, m, or g to represent Blocks (512 bytes), Kilo (1024 bytes), Mega (1024 Kilobytes), or Giga (1024 Megabytes), respectively. To represent a tape length of 2 Gigabytes, enter `-L 2g`.

Note: Use the **-L** flag for i-node backups only.

-n *snapshotName*

Specifies the name of the internal snapshot to back up. You must mount the file system containing the snapshot. The **-n** flag is used for backups by the i-node only.

-o

Creates a Version 2-compatible backup by name. This flag is required for compatibility with Version 2 systems because backups by name that are created by a version higher than 2 cannot be restored on Version 2 systems. To create a Version 2-compatible backup by name, use the **-o** flag along with other flags required for backups by name.

Files with attributes and values, such as user IDs and group IDs, that are too large for Version 2 systems will not be backed up. A message is displayed for each such file and each value that is too large.

-O

Creates a non-Trusted AIX security attributes backup.
Note: The **-O** flag only applies for systems running Trusted AIX. See Trusted AIX in the *Security* for more information.

-p

Specifies that the files be packed, or compressed, before they are archived. Only files of less than 2GB are packed.

Note: This option should only be used when backing up files from an inactive filesystem. Modifying a file when a backup is in progress may result in corruption of the backup and an inability to recover the data. When backing up to a tape device which performs compression, this option can be omitted.

-q Indicates that the removable medium is ready to use. When you specify the **-q** flag, the **backup** command proceeds without prompting you to prepare the backup medium and press the Enter key to continue. This option applies only to the first volume; you are prompted for subsequent volumes. The **-q** flag applies only to backups by name.

-U Specifies to backup any ACLs or named extended attributes. Without this option the image will include only AIXC ACLs and PCLs in the archive along with the other regular file data. For files containing NFS4 ACLs, conversion to AIXC will happen by default during archival.

-u Updates the **/etc/dumpdates** file with the raw device name of the file system and the time, date, and level of the backup. You must specify the **-u** flag if you are making incremental backups. The **-u** flag applies only to backups by i-node.

-v Causes the **backup** command to display additional information about the backup. When using the **-v** flag, the size of the file as it exists on the archive is displayed in bytes. Additionally, a total of these file sizes is displayed when all files have been processed. Directories are listed with a size of 0. Symbolic links are listed with the size of the symbolic link. Hard links are listed with the size of the file, which is how hard links are archived. Block and character devices, if they were backed up, are listed with a size of 0.

When the **-v** flag is not specified, the **backup** command displays only the names of the files being archived. This option is used only when backing up by file name.

-w Currently disabled. If the **-w** flag is specified, no other flags are applied.

-W Displays, for each file system in the **/etc/dumpdates** file, the most recent backup date and level. If the **-W** option is specified, no other flags are applied.

-*Level* Specifies the backup level (0 to 9). The default level is 9.

-Z Backs up the Encrypted File System (EFS) information for all of the files, directories, and file systems. The EFS information is extracted by default. **Note:** Archives created with **-Z** option can be restored only on AIX 6.1 or later releases.

Security

On Trusted AIX systems, only users with the **aix.fs.manage.backup** authorization can run the **backup** command.

RBAC Disclaimer

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

1. To backup all the files and subdirectories in the `/home` directory using full path names, enter:

   ```
   find /home -print | backup  -i  -f /dev/rmt0
   ```

   The **-i** flag specifies that files will be read from standard input and archived by file name. The **find** command generates a list of all the files in the `/home` directory. The files in this list are full path names. The | (pipe symbol) causes this list to be read from standard input by the **backup** command. The **-f**

flag directs the **backup** command to write the files to the /dev/rmt0 tape device. Because the files are archived using full path names, they will be written to the same paths when restored.

2. To backup all the files and subdirectories in the /home/mike directory using relative path names, enter:

```
cd /home/mike
find . -print | backup  -i  -v  -q
```

   Each file name in the list generated by the **find** command is preceded by **./** (dot, slash). Because the files are backed up using relative path names, they will be written to the current directory when restored. The **-v** flag causes the **backup** command to display additional information about the backup. The files are written to the default backup device **/dev/rfd0**.

3. To backup the **/** (root) file system, enter:

```
backup  -0  -u  -f /dev/rmt0 /
```

   The 0 level specifies that all the files in the **/** (root) file system be backed up. The **-u** flag causes the **backup** command to update the **/etc/dumpdates** file for this backup.

4. To backup all the files in the **/** (root) file system that have been modified since the last level 0 backup, enter:

```
backup  -1  -u  -f /dev/rmt0 /
```

   If the **/etc/dumpdates** file does not have an entry for a level 0 backup of the / (root) system, all the files in the file system are backed up.

5. To create an archive with Extended Attributes and ACLs, enter:

```
ls /etc/passwd | backup -ivUf arch.bk
```

6. To create an archive without Trusted AIX security attributes, enter:

```
ls /etc/passwd | backup -ivOf arch.bk
```

Files

/etc/filesystems	Contains file system mount information.
/etc/dumpdates	Specifies log for incremental by i-node backups.
/dev/rfd0	Specifies default backup device.
/dev/rhd4	Specifies device where the default file system (root) is located.
/usr/sbin/backup	Contains the **backup** command.

Related Information

The **dd** command, **find** command, **rdump** command, **restore** command.

The **dumpdates** file, **filesystems** file, **rmt** special file.

The System backup in *Operating system and device management* provides information about different methods of backing up, restoring process, different types of backup media, and guidelines for backup policies.

The Directories in *Operating system and device management* explains working with directories and path names.

The File systems in *Operating system and device management* explains file system types, management, structure, and maintenance.

The Mounting in *Operating system and device management* explains mounting files and directories, mount points, and automatic mounts.

The System Management Interface Tool (SMIT) in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* explains the structure, main menus, and tasks that are done with SMIT.

Trusted AIX chapter in the *Security*.

banner Command

Purpose

Writes ASCII character strings in large letters to standard output.

Syntax

banner *String*

Description

The **banner** command writes ASCII character *String*s to standard output in large letters. Each line in the output can be up to 10 uppercase or lowercase characters in length. On output, all characters appear in uppercase, with the lowercase input characters appearing smaller than the uppercase input characters.

Each word you input appears on a separate line on the screen. When you want to display more than one word to a line, use quotation marks to specify which words will appear on one line.

Examples

1. To display a banner at the workstation, enter:

   ```
   banner SMILE!
   ```

2. To display more than one word on a line, enclose the text in quotation marks, as follows:

   ```
   banner "Out to" Lunch
   ```

   This displays `Out to` on one line and `Lunch` on the next.

Files

/usr/bin/banner	Contains the **banner** command.

Related Information

The **echo** command.

The Input and output redirection overview in *Operating system and device management* describes how the operating system processes input and output and how to use redirect and pipe symbols.

basename Command

Purpose

Returns the base file name of a string parameter.

Syntax

basename *String* [*Suffix*]

Description

The **basename** command reads the *String* parameter, deletes any prefix that ends with a / (slash) and any specified *Suffix* parameter, and writes the remaining base file name to standard output. The **basename** command applies the following rules in creating the base file name:

1. If the *String* parameter is a // (double slash), or if the *String* parameter consists entirely of slash characters, change the string to a single / (slash). Skip steps 2 through 4.
2. Remove any trailing / characters from the specified string.
3. If there are any / characters remaining in the *String* parameter, remove the prefix of the string up to and including the last / character.
4. If a *Suffix* parameter is specified and is identical to the characters remaining in the string, the string is not modified. For example, entering:

   ```
   K > basename /u/dee/desktop/cns.boo cns.boo
   ```

   results in:

   ```
   cns.boo
   ```

   If a *Suffix* parameter is specified and is not identical to all the characters in the string but is identical to a suffix in the string, the specified suffix is removed. For example, entering:

   ```
   K > basename /u/dee/desktop/cns.boo .boo
   ```

   results in:

   ```
   cns
   ```

   Failure to find the specified suffix within a string is not considered an error.

The **basename** and **dirname** commands are generally used inside command substitutions within a shell script to specify an output file name that is some variation of a specified input file name.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

1. To display the base name of a shell variable, enter:

   ```
   basename $WORKFILE
   ```

   The command displays the base name of the value assigned to the shell variable `WORKFILE`. If the value of the `WORKFILE` variable is the **/home/jim/program.c** file, then the command displays `program.c`.
2. To construct a file name that is the same as another file name, except for its suffix, enter:

   ```
   OFILE=`basename $1 .c`.o
   ```

   This command assigns to the `OFILE` file the value of the first positional parameter (`$1`), but with its `.c` suffix changed to `.o`. If `$1` is the **/home/jim/program.c** file, `OFILE` becomes `program.o`. Because `program.o` is only a base file name, it identifies a file in the current directory.

   Note: The ` (grave accent) specifies command substitution.

Files

/usr/bin/basename	Contains the **basename** command.

Related Information

The **dirname** command, **sh** command.

---

batch Command

Purpose

Runs jobs when the system load level permits.

Syntax

batch

Description

The **batch** command reads from standard input the names of commands to be run at a later time and runs the jobs when the system load level permits. The **batch** command mails you all output from standard output and standard error for the scheduled commands, unless you redirect that output. It also writes the job number and the scheduled time to standard error.

When the **batch** command is executed, it retains variables in the shell environment, and the current directory; however, it does not retain open file descriptors, traps, and priority.

The **batch** command is equivalent to entering the **at -q b -m now** command. The **-q b** flag specifies the **at** queue for batch jobs.

Exit Status

This command returns the following exit values:

0	Successful completion
>0	An error occurred.

Examples

To run a job when the system load permits, enter:

```
batch <<!
longjob
!
```

This example shows the use of a ″Here Document″ to send standard input to the **batch** command.

Files

/usr/bin/batch	Contains the **batch** command.
/bin/batch	Symbolic link to the **batch** command.
/var/adm/cron	Indicates the main **cron** daemon directory.
/var/spool/cron/atjobs	Indicates the spool area.

Related Information

at command, **bsh** command, **csh** command, **kill** command, **ksh** command, **mail** command, **nice** command, **ps** command.

Daemons: **cron**.

Input and output redirection overview in *Operating system and device management* describes how the operating system processes input and output.

Korn shell or POSIX shell built-in commands and Bourne shell built-in commands in *Operating system and device management*.

National Language Support Overview for Programming in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* explains collating sequences, equivalence classes, and locale.

Shells in *Operating system and device management* describes what shells are, the different types of shells, and how shells affect the way commands are interpreted.

---

battery Command

Purpose

Controls or queries battery information.

Syntax

battery [**-d**]

Description

The **battery** command controls or queries the battery. If the **battery** command is invoked without **-d** option, the following battery information is displayed:

```
battery type: NiCd or NiMH
current battery usage: charging, discharging, in use, fully charged
battery capacity
current remaining capacity
full charge count
```

If the **battery** command is invoked with **-d** option, the following battery information is also displayed:

```
discharge quantity
discharge time
```

If you use 50% of a battery's capacity and charge it every time (about 20 to 30 times), then the battery cannot be used at more than 50% of its capacity. This is called the *memory effect of battery*. If, then, the battery is discharged (made empty) and then recharged, the battery can be used at 100% again.

Flags

-d Discharges the battery so you can reset the memory effect of battery.

Security

Access Control: Any User

Auditing Events: N/A

Examples

1. To show current battery status, enter:

   ```
   battery
   ```

   Something similar to the following displays:

   ```
   battery type: NiMH
   current battery usage: in use
   battery capacity: 3200 (mAH)
   current remaining capacity: 1800 (mAH) [57%]
   full charge count: 3
   ```

Files

/usr/bin/battery	Contains the **battery** command.

---

bc Command

Purpose

Provides an interpreter for arbitrary-precision arithmetic language.

Syntax

bc [**-c**] [**-l**] [*File* ...]

Description

The **bc** command is an interactive process that provides arbitrary-precision arithmetic. The **bc** command first reads any input files specified by the *File* parameter and then reads the standard input. The input files must be text files containing a sequence of commands, statements, or function definitions that the **bc** command can read and execute.

The **bc** command is a preprocessor for the **dc** command. It calls the **dc** command automatically, unless the **-c** (compile only) flag is specified. If the **-c** flag is specified, the output from the **bc** command goes to standard output.

The **bc** command allows you to specify an input and output base for operations in decimal, octal, or hexadecimal. The default is decimal. The command also has a scaling provision for decimal point notation. The **bc** command always uses the . (period) to represent the radix point, regardless of any decimal point character specified as part of the current locale.

The syntax for the **bc** command is similar to that of the C language. You can use the **bc** command to translate between bases by assigning the **ibase** keyword to the input base and the **obase** keyword to the output base. A range of 2-16 is valid for the **ibase** keyword. The **obase** keyword ranges from 2 up to the limit set by the **BC_BASE_MAX** value defined in the **/usr/include/sys/limits.h** file. Regardless of the **ibase** and **obase** settings, the **bc** command recognizes the letters A-F as their hexadecimal values 10-15.

The output of the **bc** command is controlled by the program read. Output consists of one or more lines containing the value of all executed expressions without assignments. The radix and precision of the output are controlled by the values of the **obase** and **scale** keywords.

Further information about the way in which the **bc** command processes information from a source file is described in the following sections:

- Grammar
- Lexical Conventions

- Identifiers and Operators
- Expressions
- Statements
- Function Calls
- Functions in -l Math Library

Grammar

The following grammar describes the syntax for the **bc** program, where `program` stands for any valid program:

```
%token  EOF NEWLINE STRING LETTER NUMBER
%token  MUL_OP
/*      '*', '/', '%'                           */
%token  ASSIGN_OP
/*      '=', '+=', '-=', '*=', '/=', '%=', '^='  */
%token  REL_OP
/*      '==', '<=', '>=', '!=', '<', '>'          */
%token  INCR_DECR
/*      '++', '--'                                */
%token  Define    Break    Quit    Length
/*      'define', 'break', 'quit', 'length'       */
%token  Return    For    If    While    Sqrt
/*      'return', 'for', 'if', 'while', 'sqrt'    */
%token  Scale    Ibase    Obase    Auto
/*      'scale', 'ibase', 'obase', 'auto'         */
%start   program
%%
program           : EOF
                  | input_item program
                  ;
input_item        : semicolon_list NEWLINE
                  | function
                  ;
semicolon_list    : /* empty */
                  | statement
                  | semicolon_list ';' statement
                  | semicolon_list ';'
                  ;
statement_list    : /* empty */
                  | statement
                  | statement_list NEWLINE
                  | statement_list NEWLINE statement
                  | statement_list ';'
                  | statement_list ';' statement
                  ;
statement         : expression
                  | STRING
                  | Break
                  | Quit
                  | Return
                  | Return '(' return_expression ')'
                  | For '(' expression ';'
                       relational_expression ';'
                       expression ')' statement
                  | If '(' relational_expression ')' statement
                  | While '(' relational_expression ')' statement
                  | '{' statement_list '}'
                  ;
```

```
function          : Define LETTER '(' opt_parameter_list ')'
                        '{' NEWLINE opt_auto_define_list
                        statement_list '}'
                  ;
opt_parameter_list:/* empty */
                  | parameter_list
                  ;
parameter_list    : LETTER
                  | define_list ',' LETTER
                  ;
opt_auto_define_list
                  : /* empty */
                  | Auto define_list NEWLINE
                  | Auto define_list ';'
                  ;
define_list       : LETTER
                  | LETTER '[' ']'
                  | define_list ',' LETTER
                  | define_list ',' LETTER '[' ']'
                  ;
opt_argument_list : /* empty */
                  | argument_list
                  ;
argument_list     : expression
                  | argument_list ',' expression
                  ;
relational_expression
                  : expression
                  | expression REL_OP expression
                  ;
return_expression : /* empty */
                  | expression
                  ;
expression        : named_expression
                  | NUMBER
                  | '(' expression ')'
                  | LETTER '(' opt_argument_list ')'
                  | '-' expression
                  | expression '+' expression
                  | expression '-' expression
                  | expression MUL_OP expression
                  | expression '^' expression
                  | INCR_DECR named_expression
                  | named_expression INCR_DECR
                  | named_expression ASSIGN_OP expression
                  | Length '(' expression ')'
                  | Sqrt '(' expression ')'
                  | Scale '(' expression ')'
                  ;
named_expression  : LETTER
                  | LETTER '[' expression ']'
                  | Scale
                  | Ibase
                  | Obase
                  ;
```

Lexical Conventions

The following lexical conventions apply to the **bc** command:

1. The **bc** command recognizes the longest possible lexical token or delimiter beginning at a given point.
2. Comments begin with /* (slash, asterisk) and end with */ (asterisk, slash). Comments have no effect except to delimit lexical tokens.

3. The newline character is recognized as the **NEWLINE** token.
4. The **STRING** token represents a string constant. The string begins with " (double quotation mark) and terminates with " (double quotation mark). All characters between the quotation marks are taken literally. There is no way to specify a string that contains " (double quotation mark). The length of each string is limited to the maximum bytes set in the **BC_STRING_MAX** value, which is defined in the **limits.h** file.
5. Blank characters have no effect except as they appear in the **STRING** token or when used to delimit lexical tokens.
6. The \n (backslash, newline) character:
   - delimits lexical tokens.
   - is interpreted as a character sequence in **STRING** tokens.
   - is ignored when part of a multiline **NUMBER** token.
7. A **NUMBER** token uses the following grammar:

```
NUMBER  : integer
        | '.' integer
        | integer '.'
        |integer '.' integer
        ;
integer : digit
        | integer digit
        ;
digit   : 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7
        | 8 | 9 | A | B | C | D | E | F
        ;
```

   NUMBER token values are interpreted as numerals in the base specified by the **ibase** internal register value.
8. The value of a **NUMBER** token is interpreted as a numeral in the base specified by the value of the **ibase** internal register. Each of the digit characters has the value from 0 to 15 in the order listed here, and the period character presents the radix point. The behavior is undefined if digits greater than or equal to the value of the **ibase** register appear in the token. There is an exception for single-digit values being assigned to the **ibase** and **obase** registers themselves.
9. The following keywords are recognized as tokens:

```
auto    for     length   return sqrt
break   ibase   obase    scale  while
define  if      quit
```

10. Except within a keyword, any of the following letters are considered a **LETTER** token:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

11. The following single-character and two-character sequences are recognized as the **ASSIGN_OP** token:
    - = (equal sign)
    - += (plus, equal sign)
    - -= (minus, equal sign)
    - *= (asterisk, equal sign)
    - /= (slash, equal sign)
    - %= (percent, equal sign)
    - ^= (caret, equal sign)
12. The following single characters are recognized as the **MUL_OP** token:
    - * (asterisk)
    - / (slash)
    - % (percent)
13. The following single-character and two-character sequences are recognized as the **REL_OP** token:

- == (double equal sign)
- <= (less than, equal sign)
- >= (greater than, equal sign)
- != (exclamation point, equal sign)
- < (less than)
- > (greater than)

14. The following two-character sequences are recognized as the **INCR_DECR** token:
    - ++ (double plus sign)
    - -- (double hyphen)
15. The following single characters are recognized as tokens. The token has the same name as the character:

    <newline>

    ((left parenthesis)

) (right parenthesis)

    , (comma)

    + (plus)

    - (minus)

    ; (semicolon)

    [(left bracket)

    ] (right bracket)

    ^ (caret)

    { (left brace)

    } (right brace)
16. The **EOF** token is returned when the end of input is reached.

Identifiers and Operators

There are three kinds of identifiers recognized by the **bc** command: ordinary identifiers, array identifiers, and function identifiers. All three types consist of single, lowercase letters. Array identifiers are followed by [ ] (left and right brackets). An array subscript is required except in an argument or auto list. Arrays are singly dimensioned and can contain up to the amount specified by the **BC_DIM_MAX** value. Indexing begins at 0. Therefore an array is indexed from 0 up to the value defined by **BC_DIM_MAX -1**. Subscripts are truncated to integers. Function identifiers must be followed by ( ) (left and right parentheses) and possibly by enclosing arguments. The three types of identifiers do not conflict.

The Operators in a bc Program table summarizes the rules for precedence and associativity of all operators. Operators on the same line have the same precedence. Rows are in order of decreasing precedence.

Operators in a bc Program	
Operator	**Associativity**
++, - -	not applicable
unary -	not applicable
^	right to left
*, /, %	left to right
+, binary -	left to right

=, +=, -=, *=, /=, ^=	right to left
==, <=, >=, !=, <, >	none

Each expression or named expression has a *scale,* which is the number of decimal digits maintained as the fractional portion of the expression.

Named expressions are places where values are stored. Named expressions are valid on the left side of an assignment. The value of a named expression is the value stored in the place named. Simple identifiers and array elements are named expressions; they have an initial value of zero and an initial scale of zero.

The internal registers **scale**, **ibase**, and **obase** are all named expressions. The scale of an expression consisting of the name of one of these registers is 0. Values assigned to any of these registers are truncated to integers. The **scale** register contains a global value used in computing the scale of expressions (as described below). The value of the **scale** register is limited to 0 <= **scale** <= {**BC_SCALE_MAX**} and has a default value of 0. The **ibase** and **obase** registers are the input and output number radix, respectively. The value of **ibase** is limited to 2 <= **ibase** <= 16. The value of **obase** is limited to 2 <= **obase** = {**BC_BASE_MAX**}

When either the **ibase** or **obase** registers are assigned a single-digit value from the list described in ″Lexical Conventions″ , the value is assumed in hexadecimal. For example:

```
ibase=A
```

sets to base ten, regardless of the current **ibase** register value. Otherwise, the behavior is undefined when digits greater than or equal to the value of the **ibase** register appear in the input. Both **ibase** and **obase** registers have initial values of 10.

Internal computations are conducted as if in decimal, regardless of the input and output bases, to the specified number of decimal digits. When an exact result is not achieved, for example:

```
scale=0; 3.2/1
```

the **bc** command truncates the result.

All numerical values of the **obase** register are output according to the following rules:

1. If the value is less than 0, output a - (hyphen).
2. Output one of the following, depending on the numerical value:
   - If the absolute value of the numerical value is greater than or equal to 1, output the integer portion of the value as a series of digits appropriate to the **obase** register (described in step 3). Next output the most significant non-zero digit, followed by each successively less significant digit.
   - If the absolute value of the numerical value is less than 1 but greater than 0 and the scale of the numerical value is greater than 0, it is unspecified whether the character 0 is output.
   - If the numerical value is 0, output the character 0.
3. If the scale of the value is greater than 0, output a . (period) followed by a series of digits appropriate to the following **obase** register values. The digits represent the most significant portion of the fractional part of the value, and *s* represents the scale of the value being output:
   - If the **obase** value is 10, output *s* number of digits.
   - If the **obase** value is greater than 10, output the number less than or equal to *s*.
   - If the **obase** value is less than 10, output a number greater than or equal to *s*.
   - For **obase** values other than 10, this should be the number of digits needed to represent a precision of 10s.
   - For **obase** values from 2 to 16, valid digits are the first **obase** of the single characters:

```
0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
```

which represent the values 0 through 15, respectively.

- For bases greater than 16, each digit is written as a separate multidigit decimal number. Each digit except the most significant fractional digit is preceded by a single space character. For bases 17 to 100, the **bc** command writes two-digit decimal numbers, for bases 101 to 1000 the **bc** command writes three-digit decimal numbers. For example, the decimal number 1024 in base 25 would be written as:

  ```
  01 15 24
  ```

  in base 125, as:

  ```
  008 024
  ```

Very large numbers are split across lines, with 70 characters per line in the POSIX locale. Other locales may split at different character boundaries. Lines that are continued must end with a \ (backslash).

Expressions

A numeric constant is an expression. The scale is the number of digits that follow the radix point in the input representing the constant, or 0 if no radix point appears.

The sequence (*expression*) is an expression with the same value and scale as *expression*. The parentheses can be used to alter the normal precedence.

The unary and binary operators have the following semantics:

-expression	The result is the negative of the expression. The scale of the result is the scale of the expression.
	The unary increment and decrement operators do not modify the scale of the named expression upon which they operate. The scale of the result is the scale of that named expression.
++*named_expression*	The named expression is incremented by 1. The result is the value of the named expression after incrementing.
- -*named_expression*	The named expression is decremented by 1. The result is the value of the named expression after decrementing.
named_expression++	The named expression is incremented by 1. The result is the value of the named expression before incrementing.
named_expression- -	The named expression is decremented by 1. The result is the value of the named expression before decrementing.

The exponentiation operator, ^ (caret), binds right to left.

expression ^*expression*	The result is the first *expression* raised to the power of the second *expression.* If the second expression is not an integer, the behavior is undefined. If `a` is the scale of the left expression and `b` is the absolute value of the right expression, the scale of the result is:
	`if b >= 0 min(a * b, max(scale, a))`
	`if b <     0 scale`

The multiplicative operators * (asterisk), / (slash), and % (percent) bind left to right.

expression * *expression*	The result is the product of the two expressions. If `a` and `b` are the scales of the two expressions, then the scale of the result is: `min(a+b,max(scale,a,b))`

expression / *expression*
: The result is the quotient of the two expressions. The scale of the result is the value of **scale**.

expression % *expression*
: For expressions `a` and `b`, `a % b` is evaluated equivalent to the following steps:

1. Compute `a/b` to current scale.
2. Use the result to compute:

   ```
   a - (a / b) * b
   ```

   to scale:

   ```
   max(scale + scale(b), scale(a))
   ```

   The scale of the result will be:

   ```
   max(scale + scale(b), scale(a))
   ```

When **scale** is zero, the % operator is the mathematical remainder operator.

The additive operators + (plus) and - (minus) bind left to right.

expression + *expression*
: The result is the sum of the two expressions. The scale of the result is the maximum of the scales of the expressions.

expression - *expression*
: The result is the difference of the two expressions. The scale of the result is the maximum of the scales of the expressions.

The following assignment operators bind right to left:

- = (equal sign)
- += (plus, equal sign)
- -= (minus, equal sign)
- *= (asterisk, equal sign)
- /= (slash, equal sign)
- %= (percent, equal sign)
- ^= (caret, equal sign)

named-expression = *expression*
: This expression results in assigning the value of the expression on the right to the named expression on the left. The scale of both the named expression and the result is the scale of the expression.

The compound assignment forms:

named-expression <*operator* >= *expression*

are equivalent to:

named-expression = *named-expression* <*operator* > *expression*

except that the named expression is evaluated only once.

Unlike all other operators, the following relational operators are only valid as the object of an **if** or **while** statement or inside a **for** statement:

- < (less than)
- > (greater than)

- <= (less than, equal sign)
- >= (greater than, equal sign)
- == (double equal sign)
- != (exclamation, equal sign)

expression1 < *expression2*	The relation is true if the value of *expression1* is strictly less than the value of *expression2.*
expression1 > *expression2*	The relation is true if the value of *expression1* is strictly greater than the value of *expression2.*
expression1 <= *expression2*	The relation is true if the value of *expression1* is less than or equal to the value of *expression2.*
expression1 >= *expression2*	The relation is true if the value of *expression1* is greater than or equal to the value of *expression2.*
expression1 == *expression2*	The relation is true if the values of *expression1* and *expression2* are equal.
expression1 != *expression2*	The relation is true if the values of *expression1* and *expression2* are unequal.

Statements

When a statement is an expression, unless the main operator is an assignment, execution of the statement writes the value of the expression followed by a newline character.

When a statement is a string, execution of the statement writes the value of the string.

Statements separated by semicolons or newline characters are executed sequentially. In an interactive invocation of the **bc** command, each time a newline character is read that satisfies the grammatical production:

```
input_item : semicolon_list NEWLINE
```

the sequential list of statements making up the **semicolon_list** is executed immediately, and any output produced by that execution is written without any buffer delay.

If an **if** statement (**if** (*relation*) *statement*), the *statement* is executed if the relation is true.

The **while** statement (**while** (*relation*) *statement*) implements a loop in which the *relation* is tested. Each time the *relation* is true, the *statement* is executed and the *relation* retested. When the *relation* is false, execution resumes after *statement.*

A **for** statement (**for** (*expression; relation; expression*) *statement*) is the same as:

```
first-expression
while   (relation) {
   statement
   last-expression
}
```

All three expressions must be present.

The **break** statement causes termination for a **for** or **while** statement.

The **auto** statement (**auto** *identifier* **[**,*identifier* **]** ...) causes the values of the identifiers to be pushed down. The identifiers can be ordinary identifiers or array identifiers. Array identifiers are specified by following the array name by empty square brackets. The **auto** statement must be the first statement in a function definition.

The **define** statement:

```
define   LETTER ( opt_parameter_list )  {
   opt_auto_define_list
   statement_list
}
```

defines a function named LETTER. If the LETTER function was previously defined, the **define** statement replaces the previous definition. The expression:

```
LETTER ( opt_argument_list )
```

invokes the LETTER function. The behavior is undefined if the number of arguments in the invocation does not match the number of parameters in the definition. Functions are defined before they are invoked. A function is considered defined within its own body, so recursive calls are valid. The values of numeric constants within a function are interpreted in the base specified by the value of the **ibase** register when the function is invoked.

The **return** statements (**return** and **return**(*expression*)) cause termination of a function, popping of its **auto** variables, and specify the result of the function. The first form is equivalent to return(0). The value and scale of an invocation of the function is the value and scale of the expression in parentheses.

The **quit** statement (**quit**) stops execution of a **bc** program at the point where the statement occurs in the input, even if it occurs in a function definition or in an **if**, **for**, or **while** statement.

Function Calls

A function call consists of a function name followed by parentheses containing a comma-separated list of expressions, which are the function arguments. A whole array passed as an argument is specified by the array name followed by [ ] (left and right brackets). All function arguments are passed by value. As a result, changes made to the formal parameters have no effect on the actual arguments. If the function terminates by executing a **return** statement, the value of the function is the value of the expression in the parentheses of the **return** statement, or 0 if no expression is provided or if there is no **return** statement.

The result of **sqrt**(*expression*) is the square root of the expression. The result is truncated in the least significant decimal place. The scale of the result is the scale of the expression or the value of **scale**, whichever is larger.

The result of **length**(*expression*) is the total number of significant decimal digits in the expression. The scale of the result is 0.

The result of **scale**(*expression*) is the scale of the expression. The scale of the result is 0.

There are only two storage classes in a **bc** program, global and automatic (local). Only identifiers that are to be local to a function need be declared with the **auto** keyword. The arguments to a function are local to the function. All other identifiers are assumed to be global and available to all functions. All identifiers, global and local, have initial values of 0. Identifiers declared as **auto** are allocated on entry to the function and released on returning from the function. Therefore they do not retain values between function calls. The **auto** arrays are specified by the array name followed by [ ] (left bracket, right bracket). On entry to a function, the old values of the names that appear as parameters and as automatic variables are pushed onto a stack. Until the function returns, reference to these names refers only to the new values.

References to any of these names from other functions that are called from this function also refer to the new value until one of those functions uses the same name for a local variable.

Functions in -l Math Library

The following functions are defined when you specify the **-l** flag:

s(*expression***)**	Specifies the sine of *expressionx*, where *expression* is in radians.
c(*expression***)**	Specifies the cosine of *expressionx*, where *expression* is in radians.

a(*expression***)**	Specifies the arctangent of *expressionx*, where *expression* is in radians.
l(*expression***)**	Specifies the natural logarithm of *expression*.
e(*expression***)**	Specifies the exponential of *expression*.
j(*expression***,***expression***)**	Specifies the Bessel function of integer order.

The scale of an invocation of each of these functions is the value of the **scale** keyword when the function is invoked. The behavior is undefined if any of these functions is invoked with an argument outside the domain of the mathematical function.

Flags

-c Compiles the *File* parameter, but does not invoke the **dc** command.
-l (Lowercase L) Defines a library of math functions, and sets the **scale** variable to 20.

Exit Status

This command returns the following exit values:

0	Successful completion.
1	Encountered a syntax error or could not access the input file.
unspecified	Any other error occurred.

Examples

1. You can use the **bc** command as a calculator. Depending on whether you set the **scale** variable and with what value, the system displays fractional amounts. Entering:

   ```
   bc
   1/4
   ```

   displays only `0`. To set the **scale** variable and add a comment, enter:

   ```
   scale = 1 /* Keep 1 decimal place */
   1/4
   ```

   The screen displays `0.2`. Entering:

   ```
   scale = 3 /* Keep 3 decimal places */
   1/4
   ```

   displays `0.250`. Entering:

   ```
   16+63/5
   ```

   displays `28.600`. Entering

   ```
   (16+63)/5
   ```

   displays `15.800`. Entering

   ```
   71/6
   ```

   displays `11.833`.

   The **bc** command displays the value of each expression when you press the Enter key, except for assignments.

   When you enter the **bc** command expressions directly from the keyboard, press the End-of-File (Ctrl-D) key sequence to end the **bc** command session and return to the shell command line.

2. To write and run a C-like program, enter a command similar to the following:

```
bc  -l prog.bc
e(2) /* e squared */
ma
```

The screen displays `7.38905609893065022723`. If you enter:

```
f(5)   /* 5 factorial */
```

The screen displays `120`. If you enter:

```
f(10)   /* 10 factorial */
```

The screen displays `3628800`.

This sequence interprets the **bc** program saved in the **prog.bc** file, and reads more of the **bc** command statements from the keyboard. Starting the **bc** command with the **-l** flag makes the math library available. This example uses the **e** (exponential) function from the math library, and `f` is defined in the **prog.bc** program file as:

```
/* compute the factorial of n */
define f(n) {
 auto i, r;

 r = 1;
 for (i=2; i<=n; i++) r =* i;
 return (r);
}
```

The statement following a **for** or **while** statement must begin on the same line. When you enter the **bc** command expressions directly from the keyboard, press the End-of-File (Ctrl-D) key sequence to end the **bc** command session and return to the shell command line.

3. To convert an infix expression to Reverse Polish Notation (RPN), enter:

```
bc  -c
(a * b) % (3 + 4 * c)
```

The screen displays:

```
lalb* 3 4lc*+%ps.
```

This sequence compiles the **bc** command infix-notation expression into an expression that the **dc** command can interpret. The **dc** command evaluates extended RPN expressions. In the compiled output, the `l` before each variable name is the **dc** subcommand to load the value of the variable onto the stack. The `p` displays the value on top of the stack, and the `s.` discards the top value by storing it in register **.** (dot). You can save the RPN expression in a file for the **dc** command to evaluate later by redirecting the standard output of this command. When you enter the **bc** command expressions directly from the keyboard, press the End-of-File (Ctrl-D) key sequence to end the **bc** command session and return to the shell command line.

4. To assign in the shell an approximation of the first 10 digits of pi to the variable *x*, enter:

```
x=$(printf "%s\n" 'scale = 10; 104348/33215' | bc)
```

The following **bc** program prints the same approximation of pi, with a label, to standard output:

```
scale = 10
"pi equals "
104348 / 33215
```

5. To define a function to compute an approximate value of the exponential function (such a function is predefined if the -l (lowercase L) option is specified), enter:

```
scale = 20
define  e(x){
        auto a, b, c, i, s
        a = 1
        b = 1
        s = 1
        for (i = 1; 1 == 1; i++){
                a = a*x
                b = b*i
                c = a/b
                if (c == 0) {
                        return(s)
                }
                s = s+c
        }
}
```

To print approximate values of the exponential function of the first 10 integers, enter:

```
for (i = 1; i <= 10; ++i) {
        e(i)
}
```

Files

/usr/bin/bc	Contains the **bc** command.
/usr/lib/lib.b	Contains the mathematical library.
/usr/bin/dc	Contains the desk calculator.

Related Information

The **awk** command, **dc** command.

---

bdftopcf Command

Purpose

Converts fonts from Bitmap Distribution Format (bdf) to Portable Compiled Format (pcf).

Syntax

bdftopcf [**-i** | **-t**] [**-p** *Number*] [**-u** *Number*] [**-l** | **-m**] [**-L** | **-M**] [**-o** *PcfFile*] **font-file.bdf**

Description

The **bdftopcf** command is the font compiler which converts fonts from Bitmap Distribution Format to Portable Compiled Format. Fonts in Portable Compiled Format can be read by any architecture, although the file is structured to allow one particular architecture to read them directly without reformatting. This feature allows fast reading on the appropriate machine. In addition, the files remain portable to other machines, although they are read more slowly.

Flags

-p *Number*	Sets the font glyph padding. Each glyph in the font has each scanline padded into a multiple of bytes specified by the *Number* variable, where *Number* is the value of 1, 2, 4, or 8 bytes.
-u *Number*	Sets the font scanline unit. When the font bit order is different from the font byte order, the *Number* variable describes what units of data (in bytes) are to be swapped. The *Number* variable can be the value of 1, 2, or 4 bytes.

-m Sets the font bit order to MSB (most significant bit) first. Bits for each glyph are placed in this order. Thus, the left-most bit on the screen is the highest valued bit in each unit.

-l (lowercase L) Sets the font bit order to LSB (least significant bit) first. The left-most bit on the screen is the lowest valued bit in each unit.

-M Sets the font byte order to MSB (most significant byte) first. All multibyte data in the file, including metrics and bitmaps, are written most significant byte first.

-L Sets the font byte order to LSB (least significant byte) first. All multibyte data in the file, including metrics and bitmaps, are written least significant byte first.

-t Converts fonts into *terminal* fonts whenever possible. A terminal font has each glyph image padded to the same size. The Xserver can usually render these font types more quickly.

-i Inhibits the normal computation of ink metrics. When a font has glyph images that do not fill the bitmap image because the ``on'' pixels do not extend to the edges of the metrics, the **bdftopcf** command computes the actual ink metrics and places them in the **.pcf** file.

Note: The **-t** option inhibits the behavior of this flag.

-o *PcfFile* Specifies the name of an output file. By default, the **bdftopcf** command writes the **pcf** file to standard output.

Examples

1. To convert fonts into terminal fonts whenever possible, enter:
   ```
   bdftocpf -t font-file.bdf
   ```
2. To set the glyph padding to a multiple of 4 bytes, enter:
   ```
   bdftocpf -p 4 font-file.bdf
   ```

---

bdiff Command

Purpose

Uses the **diff** command to find differences in very large files.

Syntax

bdiff { *File1* | **-** } { *File2* | **-** } [*Number*] [**-s**]

Description

The **bdiff** command compares the files specified by the *File1* and *File2* parameters and writes information about their differing lines to standard output. If either file name is **-** (minus), the **bdiff** command reads standard input. The **bdiff** command is used like the **diff** command to find lines that must be changed in two files to make them identical. The primary purpose of this command is to permit processing of files that are too large for the **diff** command.

The **bdiff** command ignores lines common to the beginning of both files, splits the remainder of each file into segments of *Number* lines each, and calls the **diff** command to compare the corresponding segments. In some cases, the 3500 line default for the *Number* parameter is too large for the **diff** command. If the **diff** command fails, specify a smaller value for the *Number* parameter and try again.

The output of the **bdiff** command has the same format as that of the **diff** command. The **bdiff** command adjusts line numbers to account for the segmenting of the files. Note that because of the file segmenting, the **bdiff** command does not necessarily find the smallest possible set of file differences.

Flags

-s Suppresses error messages from the **bdiff** command. (Note that the **-s** flag does not suppress error messages from the **diff** command).

Examples

To display the differences between the `chap1` file and the `chap1.bak` file:

```
bdiff chap1 chap1.bak
```

Files

/usr/bin/bdiff	Contains the **bdiff** command.

Related Information

The **diff** command.

Files in *Operating system and device management*.

Input and output redirection overview in *Operating system and device management*.

---

bellmail Command

Purpose

Sends messages to system users and displays messages from system users.

Syntax

To Display Messages

bellmail [**-e**] [**-f***File*] [**-p**] [**-q**] [**-r**]

To Send Messages

bellmail [**-t**] *User* ...

Description

The **bellmail** command with no flags writes to standard output, one message at a time, all stored mail addressed to your login name. Following each message, the **bellmail** command prompts you with a ? (question mark). Press the Enter key to display the next mail message, or enter one of the **bellmail** subcommands to control the disposition of the message.

Use the *User* parameter to attach a prefix to messages you send. The **bellmail** command prefaces each message with the sender's name, date and time of the message (its postmark), and adds the message to the user's mailbox. Specify the *User* parameter by pressing End Of File (the Ctrl-D key sequence) or entering a line containing only a . (period) after your message.

The action of the **bellmail** command can be modified by manipulating the **/var/spool/mail/***UserID* mailbox file in two ways:

- The default permission assignment for *others* is `all permissions denied (660)`. You may change this permission to `read/write`. When you change permissions from the default, the system preserves the file, even when it is empty, to maintain the desired permissions. You can no longer remove the file.
- You can edit the file to contain as its first line:

  ```
  Forward to person
  ```

This instruction causes all messages sent to the *User* parameter to be sent to the *Person* parameter instead. The `Forward to` feature is useful for sending all of a person's mail to a particular machine in a network environment.

To specify a recipient on a remote system accessible through Unix-to-Unix Copy Program (UUCP), preface the *User* parameter with the system name and an ! (exclamation mark). The [**-t**] *User*. . .**uucp** command contains additional information about addressing remote systems.

Note: In order to use the remote mail function, UUCP must be completely configured.

If you are interested in writing your own third-party mail program, you may need to know the following locking mechanisms used by the **bellmail** command.

1. The **bellmail** command creates a *UserID*.**lock** file in the **/var/spool/mail** directory that is opened by passing the **O_NSHARE** and **O_DELAY** flags to the **open** subroutine. If the *UserID*.**lock** file is being held, your **bellmail** process sleeps until the lock is free.
2. The **bellmail** command locks **/var/spool/mail/***UserID* with the **lockf** subroutine.

Flags

-e	Does not display any messages. This flag causes the **bellmail** command to return an exit value of 0 if the user has mail, or an exit value of 1 if there is no mail.
-f*File*	Reads mail from the named *File* parameter instead of the default mail file, **/var/spool/mail/***UserID*.
-p	Displays mail without prompting for a disposition code. This flag does not delete, copy, or forward any messages.
-q	Causes the **bellmail** command to exit when you press Interrupt (the Ctrl-C key sequence). Pressing Interrupt (Ctrl-C) alone stops only the message being displayed. (In this case, the next message sometimes is not displayed until you enter the **p** subcommand.)
-r	Displays mail in first-in, first-out order.
-t	Prefaces each message with the names of all recipients of the mail. (Without this flag, only the individual recipient's name displays as addressee.)

The *User* parameter is a name normally recognized by the **login** command. If the system does not recognize one or more of the specified *User* parameters or if the **bellmail** command is interrupted during input, the **bellmail** command tries to save the message in the **dead.letter** file in the current directory. If the **bellmail** command cannot save the message to the **dead.letter** file, it saves the message in the **$HOME/dead.letter** file. Once in this file, the message can be edited and sent again.

Note: The **bellmail** command uses the **$MAIL** environment variable to find the user's mailbox.

Subcommands

The following subcommands control message disposition:

+	Displays the next mail message (the same as pressing the Enter key).
-	Displays the previous message.
!*Command*	Runs the specified workstation command.
*****	Displays a subcommand summary.
d	Deletes the current message and displays the next message.
m *User*	Forwards the message to the specified *User* parameter.
p	Displays the current message again.
q	Writes any mail not yet deleted to the **/var/spool/mail/***UserID* file and exits. Pressing End Of File (Ctrl-D) has the same effect.
s [*File*]	Saves the message in the named *File* parameter instead of in the default mail file, **$HOME/mbox**.
w [*File*]	Saves the message, without its postmark, in the specified *File* parameter instead of in the default mail file, **$HOME/mbox**.
x	Writes all mail unchanged to **/var/spool/mail/***UserID* and exits.

Security

RBAC Disclaimer

Examples

1. To send mail to other users, enter:

   ```
   bellmail tom rachel
   Don't forget the meeting tomorrow at 9:30 a.m.
   ```

   Press Ctrl-D at the end of the message. In this example, the system mails the message to users `tom` and `rachel`.
2. To send a file to another user, enter:

   ```
   bellmail lance <proposal
   ```

   In this example, the file `proposal` is sent to user `lance`.
3. To display your mail, enter:

   ```
   bellmail
   ```

   After the most recent message is displayed, a ? (question mark) indicates the **bellmail** command is waiting for one of the **bellmail** subcommands. Enter `help` or an * (asterisk) to list the subcommands available.
4. To save a message or a file to the default mail file, enter:

   ```
   bellmail
   ```

   This command displays each message mailed to you. Press the Enter key after the ? prompt until the desired file is displayed. When the appropriate file is displayed, enter:

   ```
   s
   ```

   In this example, the file is saved in the default mail file, **$HOME/mbox**.
5. To save a message or a file to a specific file, enter:

   ```
   bellmail
   ```

   This command displays each message mailed to you. Press the Enter key after the ? prompt until the desired file is displayed. When the appropriate file is displayed, enter:

   ```
   s mycopy
   ```

   In this example, the file is saved in a file named `mycopy`, instead of in the default mail file.

Files

$HOME/dead.letter	Unmailable text.
$HOME/mbox	Your personal mailbox.
/usr/mail/*.lock	Lock for mail directory.
/var/spool/mail/*UserID*	Default system mailbox for *UserID.*
/usr/bin/bellmail	Bellmail program.

Related Information

The **mail** command, **uucp** command.

The **lockfx, lockf, or flock** subroutine, **open, openx, or creat** subroutine.

Mail applications in *Networks and communication management*.

Organizing mail options in *Networks and communication management*.

---

bffcreate Command

Purpose

Creates installation image files in backup format.

Syntax

bffcreate [**-q**] [**-S**] [**-U**] [**-v**] [**-X**] [**-d** *Device*] [**-t** *SaveDir*] [**-w** *Directory*] [**-M** *Platform*] { [**-l** | **-L**] | **-c** [**-s** *LogFile*] | *Package* [*Level*] ... | **-f** *ListFile* | **all** }

Description

The **bffcreate** command creates an installation image file in backup file format (bff) to support software installation operations.

The **bffcreate** command creates an installation image file from an installation image file on the specified installation media. Also, it automatically creates an installation image file from hyptertext images (such as those on the operating system documentation CD-ROMs). The **installp** command can use the newly created installation file to install software onto the system. The file is created in backup format and saved to the directory specified by *SaveDir*. The **.toc** file in the directory specified by the *SaveDir* parameter is updated to include an entry for the image file.

The **bffcreate** command determines the bff name according to this information:

Neutral Packages	*package.v.r.m.f.platform.installtype*
POWER-based platform Packages	*package.v.r.m.f.installtype*

Image Type	**Target bff Name**
Installation image for the POWER-based platform	*package.v.r.m.f.I*
Installation image for Neutral	*package.v.r.m.f.N.I*
3.1 update for the POWER-based platform	*package.v.r.m.f.service#*
3.2 update for the POWER-based platform	*package.v.r.m.f.ptf*
4.X** or later updates for the POWER-based platform	*package.part.v.r.m.f.U*
Update image for Neutral	*package.v.r.m.f.N.U*

** 4.X or later updates contain one *package* only. In addition, AIX Version 4 and later updates do not contain *ptf* IDs.

package = the name of the software package as described by the *PackageName* parameter

v.r.m.f = version.release.modification.fix, the level associated with the software package. The *PackageName* is usually not the same as the *fileset* name.

ptf = program temporary fix ID (also known as FixID)

The installation image file name has the form *Package.Level.I*. The *Package* is the name of the software package, as described for the *Package Name* parameter. *Level* has the format of *v.r.m.f*, where *v* = version, *r* = release, *m* = modification, *f* = fix. The *I* extension means that the image is an installation image rather than an update image.

Update image files containing an AIX 3.1 formatted update have a service number extension following the level. The *Servicenum* parameter can be up to 4 digits in length. One example is `xlccmp.3.1.5.0.1234`.

Update image files containing an AIX 3.2 formatted update have a *ptf* extension following the level. One example is `bosnet.3.2.0.0.U412345`.

AIX Version 4 and later update image file names begin with the *fileset* name, not the *PackageName*. They also have *U* extensions to indicate that they are indeed update image files, not installation images. One example of an update image file is `bos.rte.install.4.3.2.0.U`.

The **all** keyword indicates that installation image files are created for every installable software package on the device.

You can extract a single update image with the AIX Version 4 and later **bffcreate** command. Then you must specify the *fileset* name and the *v.r.m.f.* parameter. As in example 3 in the Examples section, the *PackageName* parameter must be the entire *fileset* name, `bos.net.tcp.client`, not just `bos.net`.

Attention: Be careful when selecting the target directory for the extracted images, especially if that directory already contains installable images. If a *fileset* at a particular level exists as both an installation image and as an update image in the same directory, unexpected installation results can occur. In cases like this, **installp** selects the image it finds first in the table of contents (**.toc**) file. The image it selects may not be the one you intended and unexpected requisite failures can result. As a rule of thumb, you should extract maintenance and technology levels to clean directories.

Flags

-c Change image names to package name format.

-d *Device* Specifies the name of the device where the original image resides. The device can be a CD, tape, diskette, or a directory. If the image is contained on tape, the tape device must be specified as no-rewind-on-close and no-retension-on-open (**/dev/rmt*.1** for high-density tape and **/dev/rmt*.5** for low-density tape). The default device is /**dev**/**rfd0**.

-f *ListFile* Reads a list of *PackageNames* and *Levels* from *ListFile*. *PackageNames*, each optionally followed by a level, should appear one per line of text. Any text following the second set of spaces or tabs on a line is ignored.

-l Lists the *Package*, *Level*, *Image Type* (*I* for installation images and *U* for update images), and *Part(s)* of all packages on the media.

-M*Platform* Specifies that any of the following *Platform* values may be used to list or to create backup file format (bff) images of installable software products for a specific platform:

- **A** Specifies all packages.
- **N** Specifies platform-neutral packages.
- **R** Specifies POWER-based platform packages only.

-q Suppresses the request for media.

-s *LogFile* Save changed image names in file indicated by *LogFile*.

-t *SaveDir* Specifies the directory where the installation image files are to be created. The **bffcreate** command creates the specified directory if it does not exist. If the **-t** flag is not specified, the files are saved in the **/usr/sys/inst.images** directory.

-U	Upgrades the directory structure of the destination repository to the current standard, if necessary. The current standard requires images to be organized into subdirectories according to package type and architecture. For example, installp images reside in the *SaveDir*/**installp/ppc** directory. When copying from a source containing this structure, the destination is required to conform. Specifying the **-U** flag permits the **bffcreate** command to create the appropriate subdirectory structure in your repository and move any existing images into the appropriate locations. Unless invalid manual copying is performed thereafter, this flag should only need to be used once.
-v	Writes the name of the backup format file to standard output.
-w *Directory*	Specifies the directory where a temporary working directory can be created. The **bffcreate** command creates the specified directory if it does not exist. The default directory is **/tmp**.
-S	Suppresses multiple volume processing when the installation device is a CD-ROM. Installation from a CD-ROM is always treated as a single volume, even if the CD-ROM contains information for a multiple volume CD set. This same suppression of multiple volume processing is performed if the **INU_SINGLE_CD** environment is set.
-X	Automatically extends the file system if space is needed.

Security

Access Control

You must have root authority to run this command.

RBAC Disclaimer

Examples

1. To create an installation image file from the **bos.net** software package on the tape in the **/dev/rmt0** tape drive and use **/var/tmp** as the working directory, type:

   ```
   bffcreate  -d /dev/rmt0.1
    -w /var/tmp bos.net
   ```

2. To create an installation image file from the **package** software package on the diskette in the **/dev/rfd0** diskette drive and print the name of the installation image file without being prompted, type:

   ```
   bffcreate  -q  -v
   package
   ```

3. To create a single update image file from the **bos.net.tcp.client** software package on the CD in **/dev/cd0**, type:

   ```
   bffcreate  -d
   /dev/cd0 bos.net.tcp.client 4.2.2.1
   ```

4. To list the packages on the CD in **/dev/cd0**, type:

   ```
   bffcreate  -l
    -d /dev/cd0
   ```

5. To create installation and/or update images from a CD in **/dev/cd0** by specifying a list of *PackageNames* and *Levels* in a *ListFile* called my MyListFile, type:

   ```
   bffcreate  -d /dev/cd0
    -f MyListFile
   ```

6. To create installation or update images of all software packages on the CD-ROM media for the current platform, type:

   ```
   bffcreate -d /dev/cd0 all
   ```

7. To list fileset information for the bos.games software package from a particular device, type:

```
bffcreate -d /usr/sys/inst.images/bos.games -l
```

8. To list all the Neutral software packages on the CD-ROM media, type:

```
bffcreate -d /dev/cd0 -MN -l
```

Files

/usr/sbin/bffcreate	Contains the **bffcreate** command.
/usr/sys/inst.images	Contains files in backup format for use in installing or updating a complete set or subset of software packages.
/usr/sys/inst.images/.toc	The table of contents file for the default directory where a list of installation image files in the directory is maintained.

Related Information

The **installp** command, **inutoc** command.

---

bfs Command

Purpose

Scans files.

Syntax

bfs [**-**] *File*

Description

The **bfs** command reads a file specified by the *File* parameter, but does not process the file. You can scan the file, but you cannot edit it.

The **bfs** command is basically a read-only version of the **ed** command with two exceptions: the **bfs** command can process much larger files and has additional subcommands.

Input files can be up to 32,767 lines long, with up to 255 characters per line. The **bfs** command is usually more efficient than the **ed** command for scanning a file because the file is not copied to a buffer. The **bfs** command is most useful in identifying sections of a large file that can be divided, using the **csplit** command, into more manageable pieces for editing.

If you enter the **P** subcommand, the **bfs** command prompts you with an * (asterisk). You can turn off prompting by entering a second **P** subcommand. The **bfs** command displays error messages when prompting is turned on.

The **bfs** command runs in both single- and multi-byte environments. The language environment is determined by the setting of the **LANG** environment variable (in the **/etc/environment** file) for the shell.

Forward and Backward Searches

The **bfs** command supports all of the address expressions described under the **ed** command. In addition, you can instruct the **bfs** command to search forward or backward through the file, with or without wraparound. If you specify a forward search with wraparound, the **bfs** command continues searching from the beginning of the file after it reaches the end of the file. If you specify a backward search with wraparound, the command continues searching backwards from the end of the file after it reaches the beginning. The symbols for specifying the four types of search are as follows:

/*Pattern***/**	Searches forward with wraparound for the *Pattern*.
?*Pattern***?**	Searches backward with wraparound for the *Pattern*.

>*Pattern*>	Searches forward without wraparound for the *Pattern*.
<*Pattern*<	Searches backward without wraparound for the *Pattern*.

The pattern-matching routine of the **bfs** command differs somewhat from the one used by the **ed** command and includes additional features described in the **regcmp** subroutine. There is also a slight difference in mark names: only lowercase letters a through z may be used, and all 26 marks are remembered.

Flags

- Suppresses the display of file sizes. Normally, the **bfs** command displays the size, in bytes, of the file being scanned.

Subcommands

The **e**, **g**, **v**, **k**, **n**, **p**, **q**, **w**, **=**, **!**, and null subcommands operate as explained in the **ed** command. However, the **bfs** command does not support a space between the address and the subcommand. Subcommands such as **—**, **+++-**, **+++=**, **-12**, and **+4p** are accepted. **1**,**10p** and **1**,**10** both display the first ten lines. The **f** subcommand displays only the name of the file being scanned; there are no remembered file names. The **w** subcommand is independent of output diversion, truncation, or compression (the **xo**, **xt**, and **xc** subcommands, respectively). *Compressed Output* mode suppresses blank lines and replaces multiple spaces and tabs with a single space.

The following additional subcommands are available:

xf *File*	Reads the **bfs** subcommands from the specified file. When the **bfs** command reaches the end of file or receives an interrupt signal, or if an error occurs, the **bfs** command resumes scanning the file that contains the **xf** subcommand. These **xf** subcommands can be nested to a depth of 10.
xo [*File*]	Sends further output from the **p** and null subcommands to the named file, which is created with read and write permission granted to all users. If you do not specify a *File* parameter, the **bfs** command writes to standard output. Each redirection to a file creates the specified file, deleting an existing file if necessary.
:*Label*	Positions a label in a subcommand file. The label is ended with a newline character. Spaces between the **:** (colon) and the start of the label are ignored. This subcommand can be used to insert comments into a subcommand file, since labels need not be referenced.

[*Address1*[**,***Address2*]]
xb/*Pattern***/***Label*

Sets the current line to the line containing the specified pattern, and jumps to the specified label in the current command file if the pattern is matched within the designated range of lines. The jump fails under any of the following conditions:

- The value of either the *Address1* or *Address2* parameter is not between the first and last lines of the file.
- The *Address2* value is less than the *Address1* value.
- The pattern does not match at least one line in the specified range, including the first and last lines.

This subcommand is the only one that does not issue an error message on bad addresses, so it may be used to test whether addresses are bad before other subcommands are run. The subcommand:

```
xb/^/label
```

is an Unconditional Jump.

The **xb** subcommand is allowed only if it is read from some place other than a workstation. If it is read from a pipe, only a Downward Jump is possible.

xt [*Number*]

Truncates output from the **p** subcommand and the null subcommands to the number of characters. The default value of the *Number* parameter is 192.

xv[*Digit*] [*Value*]

Assigns the specified *Value* to the *Digit* parameter. The value of the *Digit* parameter can be 0 through 9. You can put one or more spaces between *Digit* and *Value*. For example:

```
xv5 100
xv6 1,100p
```

assigns the value `100` to the variable `5` and the value `1,100p` to the variable `6`.

To reference a variable, put a `%` (percent sign) in front of the variable name. Given the preceding assignments for variables `5` and `6`, the following three subcommands:

```
1,%5p
1,%5
%6
```

each display the first 100 lines of a file.

To escape the special meaning of `%`, precede it with a \ (backslash). For example:

```
g/".*\%[cds]/p
```

matches and lists lines containing **printf** variables (%c, %d, or %s).

You can also use the **xv** subcommand to assign the first line of command output as the value of a variable. To do this, make the first character of the *Value* parameter an ! (exclamation point), followed by the command name. For example:

```
xv5 !cat junk
```

stores the first line of the `junk` file in the variable `5`.

To escape the special meaning of `!` as the first character of *Value*, precede it with a \ (backslash). For example:

```
xv7 \!date
```

stores the value `!date` in the variable `7`.

xbz *Label*

Tests the last saved exit value from a shell command and jumps to the specified label in the current command file if the value is 0.

xbn *Label*
Tests the last saved exit value from a shell command and jumps to the specified label in the current command file if the value is not 0.

xc [*Switch*]
Turns compressed output mode on or off. (Compressed output mode suppresses blank lines and replaces multiple spaces and tabs with a single space.)

If the *Switch* parameter has a value of 1, output from the **p** subcommand and the null subcommands is compressed. If the *Switch* parameter is 0, this output is not compressed. If you do not specify a value for the *Switch* parameter, the current value of the *Switch* parameter, initially set to 0, reverses.

Exit Status

The following exit values are returned:

0	Successful completion without any file or command errors
>0	An error occurred.

Files

/usr/bin/bfs	Contains the **bfs** command.

Related Information

The **csplit** command, **ed** or **red** command.

The **environment** file.

The **regcmp** or **regex** subroutine.

File and directory access modes in *Operating system and device management* introduces file ownership and permissions to access files and directories.

Directories in *Operating system and device management*.

Files in *Operating system and device management* describes files, file types, and how to name files.

Input and output redirection overview in *Operating system and device management* describes how the operating system processes input and output.

bg Command

Purpose

Runs jobs in the background.

Syntax

bg [*JobID* ...]

Description

If job control is enabled (see ″Job Control in the Korn shell or POSIX shell″ in *Operating system and device management*), the **bg** command resumes suspended jobs in the current environment by running

them as background jobs. If the specified job is already running in the background, the **bg** command has no effect and exits successfully. If no *JobID* parameter is supplied, the **bg** command uses the most recently suspended job.

The *JobID* parameter can be a process ID number, or you can use one of the following symbol combinations:

%*Number*	Refers to a job by the job number.
%*String*	Refers to a job whose name begins with the specified string.
%?*String*	Refers to a job whose name contains the specified string.
%+ OR **%%**	Refers to the current job.
%-	Refers to the previous job.

Using the **bg** command to place a job into the background causes the job's process ID to become known in the current shell environment. The **bg** command output displays the job number and the command associated with that job. The job number can be used with the **wait**, **fg**, and **kill** commands by prefixing the job number with a % (percent sign). For example, **kill %3**.

A job is suspended by using the **Ctrl-Z** key sequence. That job can be restarted in the background using the **bg** command. This is effective if the job expects no terminal input and if job output is redirected to non-terminal files. If a background job has terminal output, the job can be forced to stop by entering the following command:

```
stty tostop
```

A background job can be stopped by entering the following command:

```
kill -s stop JobID
```

The **/usr/bin/bg** command does not work when operating in its own command execution environment, because that environment does not have suspended jobs to manipulate. This would be the case in the following example:

```
Command | xargs bg
```

Each **/usr/bin/bg** command operates in a different environment and does not share the parent shell's understanding of jobs. For this reason, the **bg** command is implemented as a Korn shell or POSIX shell regular built-in.

Exit Status

The following exit values are returned:

0	Successful completion.
>0	An error occurred.

If job control is disabled, the **bg** command exits with an error, and no job is placed in the background.

Examples

If the output of the **jobs** command displays the following stopped job:

```
[2] + Stopped (SIGSTOP)  sleep 100 &
```

use the job number to resume the `sleep 100 &` job by entering:

```
bg %2
```

The screen displays the revised status of job 2:

```
[2] sleep 100 &
```

Files

/usr/bin/ksh	Contains the Korn shell **bg** built-in command.
/usr/bin/bg	Contains the **bg** command.

Related Information

The **csh** command, **fg** command, **jobs** command, **kill** command, **wait** command.

Job Control in the Korn shell or POSIX shell in *Operating system and device management*.

---

bicheck Command

Purpose

Syntax checker for user-modified **bosinst.data** files.

Syntax

bicheck *Filename*

Description

The **bicheck** command checks for the existence of the control flow, target_disk_data, and locale stanzas in the **bosinst.data** file. The parameter *Filename* indicates the **bosinst.data** file you want to verify. The value—if not blank—for each field in a stanza is confirmed to match an allowable value, if possible, and checked for length limitations and/or other possible limitations.

If a non-prompted install is specified, the existence of values for required fields is confirmed.

If a dump stanza exists and if the value is not blank, the value is determined to match an allowable value, if possible. It is also checked for length limitations and/or other possible limitations.

The **bicheck** command does not stop after the first error, but continues to list all problems it finds with the given **bosinst.data** file. All error messages are sent to standard error.

Exit Status

This command returns the following exit values:

0	Successful completion.
1	An error occurred.

Files

/usr/lpp/bosinst/bicheck contains the **bicheck** command.

Related Information

The **mksysb** command.

---

biff Command

Purpose

Enables or disables mail notification during the current session.

Syntax

biff [**y** | **n**]

Description

The **biff** command informs the system whether you want to be notified when mail arrives. When mail notification is enabled, From and Subject header lines and the first 7 lines or 560 characters of a message are displayed on the screen when mail arrives. Notification, specified by the **biff y** command, is often included in the **$HOME/.login** or **$HOME/.profile** file to be executed each time the user logs in. The **biff n** command disables notification.

Note: In addition to **y** and **n**, you can use **yes** and **no** to enable and disable mail notification.

The **biff** command operates asynchronously. To receive notification when mail arrives, ensure:

1. The message permission setting is on in your shell (`mesg y`).
2. **comsat** is running (started by the **inetd** daemon).
3. Notification is enabled (`biff y`).

For synchronous notification, use the **MAIL** variable of either the **ksh** command, **bsh** command, or the **csh** command.

Options

y Enables mail notification.
n Disables mail notification.

Examples

1. To display the current setting, enter:
   ```
   biff
   ```
2. To be notified during the current terminal session whenever mail arrives, enter the following statement in your **$HOME/.login** or **$HOME/.profile** file:
   ```
   biff y
   ```

   The From and Subject header lines and the first seven lines or 560 characters of the message will be displayed on the screen when mail arrives.

Files

$HOME/.login	Read by **login** shell at login.
$HOME/.profile	Controls start-up processes and daemons.
/usr/bin/biff	Contains **biff** command.

Related Information

The **bsh** command, **csh** command, **ksh** command, **mail** command.

The **comsat** daemon.

Mail applications in *Networks and communication management*.

bindintcpu Command

Purpose

Assigns a bus interrupt level to be delivered only to the indicated CPUs.

Syntax

bindintcpu *Level CPU* [*CPU*...]

bindintcpu -u *Level*

bindintcpu -q *Level*

Description

The **bindintcpu** command lets system administrators direct interrupts from a specific hardware device at the specified bus interrupt *Level* to a specific *CPU* number, or sets of *CPU* numbers. Normally, on multiple CPU systems, hardware device interrupts can be delivered to any running CPU, and the distribution among the CPUs is determined by a predefined method. The **bindintcpu** command lets the system administrator bypass the predefined method, and control the interrupts distribution from a specific device to selected CPUs. This command is applicable only on selective hardware types.

If an interrupt level has been bound with certain CPUs, all interrupts coming from that level will be distributed only to specified CPUs until it is redirected by **bindintcpu** again. If the **-q** flag is used, this utility will instead list to which CPUs the interrupt Level is bound. With the **-u** flag, an administrator can unbind a specified interrupt from its CPUs, and that interrupt will once again be delivered to any running CPU through some predefined method. However, interrupts bound to **CPU0** cannot be redirected again. If an interrupt level has been bound to **CPU0**, it stays on **CPU0** until the system is booted again.

Notes:

- Not all hardware models support one-to-many bindings, specifying multiple CPUs with **bindintcpu** results in errors on certain types of machines. For consistency, it is recommended to specify one CPU per **bindintcpu** whenever possible.
- To see the bus interrupt level for a specific adapter, use the **lsattr** command and reference the **busintr** field. For example, device ent0 below has busintr value of 6.

```
 lsattr -E -l ent0
busio           0xbff400       Bus I/O address                   False
busintr         6              Bus interrupt level               False
intr_priority   3              Interrupt priority                False
tx_que_size     256            TRANSMIT queue size               True
rx_que_size     256            RECEIVE queue size                True
rxbuf_pool_size 384            RECEIVE buffer poof size          True
media_speed     10_Half_Duplex Media Speed                       True
use_alt_addr    no             Enable ALTERNATE ETHERNET address True
alt_addr        0x000000000000 ALTERNATE ETHERNET address        True
ip_gap          96             Inter-Packet Gap                  True
```

Flags

-q	List to which CPUs the interrupt Level is bound.
-u	Unbinds a specified interrupt from its CPUs.

Security

RBAC Disclaimer

Examples

1. To direct all interrupts from bus interrupt level 6 to CPU1, enter the following command:
   ```
   bindintcpu 6 1
   ```
2. To direct all interrupts from buss interrupt level 6 to CPU2 and CPU3, enter the following command:
   ```
   bindintcpu 6 2 3
   ```

Files

/usr/sbin/bindintcpu	Contains the **bindintcpu** command.

Related Information

The **lsattr** command.

---

bindprocessor Command

Purpose

Binds or unbinds the kernel threads of a process to a processor.

Syntax

bindprocessor *Process* [*ProcessorNum*] | **-q** | **-u** *Process*{*ProcessID* [*ProcessorNum*] | **-u** *ProcessID* | **-s** *SmtSetID* | **-b** *bindID ProcessorNum* | **-q** }

Description

The **bindprocessor** command binds or unbinds the kernel threads of a process, or lists available processors. The *Process* parameter is the process identifier of the process whose threads are to be bound or unbound, and the *ProcessorNum* parameter is the bind CPU identifier of the processor to be used. If the *ProcessorNum* parameter is omitted, the process is bound to a randomly selected processor.

If simultaneous multi-threading is enabled, each hardware thread of a physical processor is listed as a separate processor by the **bindprocessor** command. This allows software threads to be bound to each hardware thread separately. There are two hardware threads on a POWER5 processor, and they are referred to as the *primary hardware thread* and *secondary hardware thread*. The *SmtSetId* parameter is the simultaneous multi-thread set identifier value of a hardware thread and is defined to be `0` for primary hardware threads and `1` for secondary hardware threads. The **-s** flag can be used to list available processors that are all primary hardware threads or that are all secondary hardware threads. The **-b** flag lists all the available hardware threads on a single physical processor on which the *ProcessorNum* parameter is the bind CPU identifier of either the primary hardware thread or the secondary hardware thread on that processor. Refer to Simultaneous Multi-Threading in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* for more information.

The **bindprocessor** command will fail if the target process has a *Resource Attachment*.

Programs that use processor bindings should become Dynamic Logical Partitioning (DLPAR) aware. Refer to Dynamic Logical Partitioning in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* for more information.

It is important to understand that a process itself is not bound, but rather its kernel threads are bound. Once kernel threads are bound, they are always scheduled to run on the chosen processor, unless they are later unbound. When a new thread is created, it has the same bind properties as its creator. This

applies to the initial thread in the new process created by the **fork** subroutine: the new thread inherits the bind properties of the thread which called **fork**. When the **exec** subroutine is called, thread properties are left unchanged.

The **-q** flag of the **bindprocessor** command lists the available bind CPU identifiers: you can use the logical numbers given as values for the *ProcessorNum* parameter. The **-u** flag unbinds the threads of a process, allowing them to run on any processor.

When simultaneous multi-threading is enabled, the **-s** flag of the **bindprocessor** command allows you to bind the threads of an application to separate physical processors by listing the processors separately. The **-b** flag is useful if you want to bind all the threads of an application to the hardware threads of the same physical processor.

Notes:

1. The **bindprocessor** command is meant for multiprocessor systems. Although it will also work on uniprocessor systems, binding has no effect on such systems.
2. You need root authority to bind or unbind threads in processes you do not own.
3. If you attempt to bind kernel processes such as **swapper** and **sched** from the user space, the operation fails with the **EPERM** error code. You can determine which kernel processes will fail by looking for the **SSCHEDPROC** flag in the process structure. If the **SSCHEDPROC** flag is set, binding the kernel process will fail.

Flags

-b Binds all threads of an application to the hardware threads of the same physical processor.
-q Displays the processors which are available.
-s Binds all threads of an application to separate physical processors by listing the processors separately.
-u Unbinds the threads of the specified process.

Examples

1. To see which processors are available (possible *ProcessorNum* values*)*, type:

   ```
   bindprocessor -q
   ```

   For a four processor system, the output is similar to:

   ```
   The available processors are: 0 1 2 3
   ```

2. To bind the threads in process 19254 to processor 1, type:

   ```
   bindprocessor 19254 1
   ```

3. To see all the available processors that are primary hardware threads, type:

   ```
   bindprocessor -s 0
   ```

   For a four-processor system with simultaneous multi-threading enabled, the output is similar to:

   ```
   The available processors are: 0 2 4 5
   ```

   To see all the available processors that are secondary hardware threads, type:

   ```
   bindprocessor -s 1
   ```

   The output is similar to:

   ```
   The available processors are: 1 3 6 7
   ```

   When simultaneous multi-threading is disabled using the **smtctl** command, or on systems with processors that do not support simultaneous multi-threading, the outputs would be:

```
bindprocessor -s 0

The available processors are: 0 1 2 3

bindprocessor -s 1

SmtSetId 1 is not available
```

4. To see all the available bind CPU IDs on a physical processor that has a hardware thread with a bind CPU ID of `0`, type:

   ```
   bindprocessor -b 0
   ```

   The output is similar to:

   ```
   The available processors are: 0 1
   ```

   Again, typing the command:

   ```
   bindprocessor -b 1
   ```

   will also result in the same output.

File

/usr/sbin/bindprocessor	Contains the **bindprocessor** command.

Related Information

The **smit** command, **smtctl** command.

The **bindprocessor** subroutine. **exec** subroutine, **fork** subroutine.

Controlling Processor Use, Dynamic Logical Partitioning, and Simultaneous Multi-Threading in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

---

binld Daemon

Purpose

Implements a Preboot Execution Environment (PXE) boot server. Serves boot file transfer server addresses and determines the appropriate boot file for PXE clients.

Syntax

To serve boot file information to the PXE clients using the system resource controller:

startsrc -s **binld** [**-a**] ...

To serve boot file information to the PXE clients without using the system resource controller:

binld [**-f**] [**-i**]

Description

The BINLD server assigns boot files for PXE clients and informs the clients where they should download the boot file. The BINLD daemon runs in the background and maintains a database of boot files that it serves and the client information (client architecture, client machine identifier, major and minor version of the network identifier) that is appropriate for each boot file. The initial boot file database is specified by the configuration file. The configuration file also contains all the data needed to assign PXE clients their boot file information.

On startup, a BINLD server reads the configuration file and sets up its initial database of available boot files. The BINLD server accepts the **refresh** command or a SIGHUP signal to reread the configuration file.

Flags

-a	The argument to be supplied.
-f	ConfigurationFile. Specifies the configuration file to be used.
-i	IP address. Specifies to which DHCP server IP address the DHCPINFORM should be sent.

Exit Status

This command returns the following exit values:

0	Successful completion.
>	An error occurred.

Security

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

Files

/usr/sbin/binld	Contains the BINLD daemon.

Related Information

The pxed command, startsrc command, stopsrc command.

---

biod Daemon

Purpose

Handles client requests for files.

Syntax

/usr/sbin/biod *NumberOfBiods*

Description

The **biod** daemon is retained for backward compatibility with scripts that invoke it. It no longer plays an active role in management of the NFS client subsystem. Instead, the NFS client internally manages its resources for performing I/O operations to NFS servers.

The *NumberOfBiods* argument historically allowed control of NFS client thread resources for performing I/O operations. This no longer has any effect. The maximum number of **biod** threads for I/O operations can be set as a mount option. The **biod** daemon might be removed in future AIX releases.

Files

/etc/rc.nfs	Contains the startup script for the NFS and NIS daemons.

Related Information

The **chnfs** command, **chssys** command, **mount** command.

The **mountd** daemon, **nfsd** daemon.

How to Mount a File System Explicitly in *Networks and communication management*.

Network File System (NFS) Overview for System Management in *Networks and communication management*.

System Resource Controller in *Operating system and device management*.

List of NFS commands in *Networks and communication management*.

---

bj Command

Purpose

Starts the blackjack game.

Syntax

bj

Description

The **bj** command invokes the blackjack game. Blackjack is a card game. The object of blackjack is to be dealt cards with a value of up to but not over 21 and to beat the dealer's hand. The computer plays the role of the dealer in blackjack.

You place bets with the dealer on the likelihood that your hand will come equal or closer to 21 than will the dealer's. The following rules apply to betting.

The bet is two dollars every hand. If you draw a natural blackjack, you win three dollars. If the dealer draws a natural blackjack, you lose two dollars. If you and the dealer both have natural blackjacks, you exchange no money (a push).

If the dealer has an ace showing, you can make an insurance bet on the chance that the dealer has a natural blackjack, winning two dollars if the dealer has a natural blackjack and losing one dollar if not.

If you are dealt two cards of the same value, you can double, that is, play two hands, each of which begins with one of these cards, betting two dollars on each hand. If the value of your original hand is 10 or 11, you can double down, that is, double the bet to four dollars and receive exactly one more card in that hand.

Under normal play, you can draw a card (take a hit) as long as your cards total 21 or less. If the cards total more than 21, you bust and the dealer wins the bet. When you stand (decide not to draw another card), the dealer takes hits until a total of 17 or more is reached. If the dealer busts, you win. If both you and the dealer stand, the one with the higher total below or equal to 21 wins. A tie is a push.

The computer deals, keeps score, and asks the following questions at appropriate times: `Do you want a hit? Insurance? Double? Double down?` To answer yes, press Y; to answer no, press the Enter key.

The dealer tells you whenever the deck is being shuffled and displays the action (total bet) and standing (total won or lost). To quit the game, press the Interrupt (Ctrl-C) or End Of File (Ctrl-D) key sequence; the computer displays the final action and score and exits.

Files

/usr/games Location of the system's games.

Related Information

The **arithmetic** command, **back** command, **craps** command, **fish** command, **fortune** command, **hangman** command, **moo** command, **number** command, **quiz** command, **ttt** command, **turnoff** command, **turnon** command, **wump** command.

---

bootauth Command

Purpose

Allows only the authorized user to boot the system.

Syntax

bootauth

Description

The **bootauth** command verifies that the system is being started by an authorized user.

The **bootauth** command prompts you for a user name and a password. If the user name and the password entered are not valid, or if the user name does not have the **aix.system.boot** authorization, the **bootauth** command reissues the prompt. After three unsuccessful attempts, the system is restarted.

Security

To start the system successfully, you must have the following authorization:

aix.system.boot Required to start the system.

Files

/usr/sbin/bootauth Contains the **bootauth** command.

Related Information

Trusted AIX in *Security*.

---

bootlist Command

Purpose

Displays and alters the list of boot devices available to the system.

Syntax

bootlist [{ **-m** *Mode* } [**-r**] [**-o**] [[**-i**] [**-V**] [**-F**]| [[**-f** *File*] [*Device* [*Attr*=*Value* ...] ...]]] [**-v**]

Description

The **bootlist** command allows the user to display and alter the list of possible boot devices from which the system may be booted. When the system is booted, it will scan the devices in the list and attempt to boot from the first device it finds containing a boot image. This command supports the updating of the following:

- Normal boot list. The normal list designates possible boot devices for when the system is booted in normal mode.
- Service boot list. The service list designates possible boot devices for when the system is booted in service mode. How a system is booted in service mode is hardware-platform dependent. It may require a key switch to be turned to the Service position, a particular function key to be pressed during the boot process, or some other mechanism, as defined for the particular hardware platform.
- Previous boot device entry. This entry designates the last device from which the system booted. Some hardware platforms may attempt to boot from the previous boot device before looking for a boot device in one of the other lists.

Support of these boot lists may vary from platform to platform. A boot list can be displayed or altered only if the platform supports the specified boot list. It may even be the case that a particular hardware platform does not support any of the boot lists.

When searching for a boot device, the system selects the first device in the list and determines if it is bootable. If no boot file system is detected on the first device, the system moves on to the next device in the list. As a result, the ordering of devices in the device list is extremely important.

The **bootlist** command supports the specification of generic device types as well as specific devices for boot candidates. Possible device names are listed either on the command line or in a file. Devices in the boot device list occur in the same order as devices listed on the invocation of this command.

The devices to be entered into the boot list may be specified in a file. This makes an alterable record of the boot devices available for reference or future update. When the **-f** flag is used, the list of devices is taken from the file specified by the *file* variable. Devices from this list are then placed in the boot list in the order found in the file.

> **Attention:** Care must be taken in specifying the possible boot devices. A future reboot may fail if the devices specified in the device list become unbootable. The system must not be powered off or reset during the operation of the **bootlist** command. If the system is reset, or if power fails at a critical point in the execution of this command, the boot list may be corrupted or lost.

The selection of the boot list to display or alter is made with the **-m** *mode* option, where the *mode* variable is one of the keywords: **service**, **normal**, **both**, or **prevboot**. If the **both** keyword is specified, then both the normal boot list and the service boot list will be displayed, or if being altered, will be set to the same list of devices. If the **prevboot** keyword is specified, the only alteration allowed is with the **-i** (invalidate) flag . The **-i** flag invalidates the boot list specified by the **-m** flag.

The devices currently in the boot list may be displayed by using the **-o** flag. The list of devices that make up the specified boot list will be displayed, one device per line. If a device specified in the boot list is no longer present on the system, a `-' is displayed instead of a name. The output is in a form that can be captured in a file and used as input to the **bootlist** command with the **-f** flag. This may be useful for restoring a boot list after making a temporary change.

Note: When you add a hot plug adapter to the system, that adapter and its child devices might not be available for specification as a boot device when you use the **bootlist** command. You may be required to reboot your system to make all potential boot devices known to the operating system.

When specifying a disk device, additional information might need to be added to the disk using an *attribute=value* pair. This extra information is only needed when the target disk has multiple instances of AIX installed on it. When this condition exists, identify the boot logical volume on the target disk that is to be included in the boot list by its attribute **blv**.

The **blv** attribute can be used in all cases, but it is only required when the target disk has multiple instances of AIX installed. When **bootlist** displays information with the **-o** flag, the **blv** attribute is always included for each disk, even if there is only one instance of AIX on that disk.

Device Choices

The device name specified on the command line (or in a file) can occur in one of two different forms:

- It can indicate a specific device by its device logical name.
- It can indicate a generic or special device type by keyword. The following generic device keywords are supported:

fd	Any standard I/O-attached diskette drive
scdisk	Any SCSI-attached disk (including serial-link disk drives)
badisk	Any direct bus-attached disk
cd	Any SCSI-attached CD-ROM
rmt	Any SCSI-attached tape device
ent	Any Ethernet adapter
tok	Any Token-Ring adapter
fddi	Any Fiber Distributed Data Interface adapter

Note: Some hardware platforms do not support generic device keywords. If a generic device keyword is specified on such a platform, the update to the boot list is rejected and this command fails.

When a specific device is to be included in the device list, the device's logical name (used with system management commands) must be specified. This logical name is made up of a prefix and a suffix. The suffix is generally a number and designates the specific device. The specified device must be in the Available state. If it is not, the update to the device list is rejected and this command fails. The following devices and their associated logical names are supported (where the bold type is the prefix and the *xx* variable is the device-specific suffix):

fd*xx*	Diskette-drive device logical names
hdisk*xx*	Physical-volume device logical names
cd*xx*	SCSI CD-ROM device logical names
rmt*xx*	Magnetic-tape device logical names
ent*xx*	Ethernet-adapter logical names
tok*xx*	Token-ring adapter logical names
fddi*xx*	Fiber Distributed Data Interface adapter logical names

Attribute Choices

Attributes are extra pieces of information about a device that the user supplies on the command line. Since this information is specific to a particular device, generic devices do not have attributes. Attributes apply to the device that immediately precedes them on the command line, which allows attributes to be interspersed among devices on the command line. Currently, only network devices have attributes. These are:

bserver	The IP address of the BOOTP server
gateway	The IP address of the gateway
client	The IP address of the client
speed	Network adapter speed
duplex	The mode of the network adapter

filename The name of the file that is loaded by Trivial File Transfer Protocol (TFTP) from the BOOTP server

These attributes can be combined in the following ways:

- The **hardware** attribute cannot be specified alone; it must be specified with the **bserver** or **gateway** attribute. When specified with **bserver** or **gateway**, it applies to the server or gateway, respectively; when both **bserver** and **gateway** are specified, **hardware** will apply to **gateway**.
- The **bserver** attribute can be specified alone, with **hardware**, and/or **gateway**.
- If the **gateway** attribute is specified, **bserver** and **client** must also be specified.
- The **client** attribute can only be specified with **gateway** and **bserver**.

Some of these attributes may not be supported on some hardware platforms. Additional hardware platform restrictions may apply.

The syntax for specifying an attribute is *attr=value*, where *attr* is the attribute name, *value* is the value, and there are no spaces before or after the =.

File Format When Using the -f Flag

The file specified by the *file* variable should contain device names separated by white space:

```
hdisk0 hdisk1 cd1
```

or one device per line:

```
hdisk0
hdisk1
cd1
```

Error Handling

If this command returns with an error, the device lists are not altered. The following device list errors are possible:

- If the user attempts to display or alter a boot list that is not supported by the hardware platform, the command fails, indicating the mode is not supported.
- If the user attempts to add too many devices to the boot list, the command fails, indicating that too many devices were requested. The number of devices supported varies depending on the device selection and the hardware platform .
- If an invalid keyword, invalid flag, or unknown device is specified, the command fails with the appropriate error message.
- If a specified device is not in the Available state, the command fails with the appropriate error message.

Flags

Device	Provides the names of the specific or generic devices to include in the boot list.
-f *File*	Indicates that the device information is to be read from the specified file name.
-F	Indicates that the boot list must be modified even if the validation of the **speed** and **duplex** attributes, if specified, is not possible.
-i	Indicates that the device list specified by the **-m** flag should be invalidated.
-m *Mode*	Specifies which boot list to display or alter. Possible values for the *mode* variable are **normal**, **service**, **both**, or **prevboot**.
-o	Indicates that the specified boot list is to be displayed after any specified alteration is performed. The output is a list of device names.
-r	Indicates that the specified boot list is to be displayed after any specified alteration is performed. The output is hardware-platform dependent. It may be a hexadecimal dump of the boot list or a list of device names. (This is normally used for problem determination.)
-V	Indicates that the speed and duplex attributes, if specified, are to be verified only. The boot list is not modified.

-v Displays verbose output. This flag is for problem determination only.

Security

Privilege Control

Only the root user and members of the security group should have execute (x) access to this command.

Auditing Events

Event	Information
NVRAM_Config	File name

RBAC Disclaimer

Examples

1. To invalidate the Service mode boot list, enter:

   ```
   bootlist -m service -i
   ```

2. To make a boot list for Normal mode with devices listed on the command line, enter:

   ```
   bootlist -m normal hdisk0 hdisk1 rmt0 fd
   ```

3. To make a boot list for Normal mode with a device list from a file, enter:

   ```
   bootlist -m normal -f /bootlist.norm
   ```

   where **bootlist.norm** is a file containing device names to be placed in the boot list for Normal mode. The device names in the **bootlist.norm** file must comply with the described format.

4. To invalidate the previous boot device entry, enter:

   ```
   bootlist -m prevboot -i
   ```

5. To boot from a Token-Ring device in slot 2, enter:

   ```
   bootlist -m normal tok0
   ```

6. To attempt to boot through a gateway using Ethernet, and then try other devices, enter:

   ```
   bootlist -m normal ent0 gateway=129.35.21.1 bserver=129.12.2.10
   \ client=129.35.9.23 hdisk0 rmt0 tok0 bserver=129.35.10.19
   hdisk1
   ```

7. To specify boot logical volume hd5 on disk hdisk0 for a normal boot, type:

   ```
   bootlist -m normal hdisk0 blv=hd5
   ```

8. To view the boot list set in the preceding example, type:

   ```
   bootlist -m normal -o
   hdisk0 blv=hd5
   ```

9. To specify booting in normal mode from the only boot logical volume on hdisk0, or the mb_hd5 boot logical volume on hdisk1, type:

   ```
   bootlist -m normal hdisk0 hdisk1 blv=mb_hd5 cd0
   ```

10. To view the boot list set in the preceding example, type:

    ```
    bootlist -m normal -o
    hdisk0
    hdisk1 blv=mb_hd5
    cd0
    ```

Related Information

The **nvram** special file.

Device Configuration Subsystem Programming Introduction in *AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts*.

List of Device Configuration Commands in *AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts*.

Special Files Overview in *AIX 5L Version 5.3 Files Reference*.

---

bootparamd Daemon

Purpose

Provides information for booting to diskless clients.

Syntax

/usr/sbin/rpc.bootparamd [**-d**]

Description

The **bootparamd** daemon is a server process that provides information necessary to diskless clients for booting. It consults either the **bootparams** database or the **/etc/bootparams** file if the NIS service is not running.

Flags

-d Displays debugging information.

Files

/etc/bootparams Contains the list of client entries that diskless clients use for booting.

Related Information

Network File System (NFS) Overview for System Management in *Networks and communication management*.

List of NFS commands in *Networks and communication management*.

---

bootpd Daemon

Purpose

Sets up the Internet Boot Protocol server.

Syntax

bootpd [**-s**] [**-t** *Integer*] [**-d** [**-d** ...]] [**-g**] [C*onfigFile* [*DumpFile*]]

Description

The **bootpd** command implements an Internet Boot Protocol server.

The **bootpd** daemon is normally started by the **inetd** daemon. The default **/etc/inetd.conf** file contains the following line:

```
bootps dgram udp wait root /usr/sbin/bootpd bootpd
```

By default, this entry is commented out. One way to add the **bootpd** daemon to the **inetd** daemon's list of available subservers is to use the System Management Interface Tool (SMIT). Another way to make the

bootpd daemon available is to edit the **/etc/inetd.conf** file, uncomment the `bootps` entry, and enter `refresh -s inetd` or `kill -1 InetdPid` to inform the **inetd** daemon of the changes to its configuration file. Now, when a bootp request arrives, **inetd** starts the **bootpd** daemon. Once the daemon is started, **bootpd** continues to listen for boot requests. However, if the server does not receive a boot request within 15 minutes of the previous one, it exits to conserve system resources. This time-out value of 15 minutes can be changed using the **-t** flag.

To start the **bootpd** daemon without **inetd**, use the `-s` flag. In this mode, the **bootpd** daemon continues to listen for bootp requests until the daemon is killed.

Upon startup, the **bootpd** daemon looks in the **/etc/services** file to find the port numbers to use, and extracts the following entries:

bootps The BOOTP server listening port.
bootpc The destination port used to reply to clients.

Then, the **bootpd** daemon reads its configuration file. If a configuration file is not specified, the default file is **/etc/bootptab**. Once the configuration file is read, the **bootpd** daemon begins listening for and processing bootp requests. The **bootpd** daemon rereads its configuration file when it receives a **SIGHUP** hang-up signal, or when it receives a bootp request packet and detects that the file has been updated. Hosts may be added, deleted, or modified when the configuration file is reread.

Flags

-d Increases the level of debugging output. This flag can be used many times. The following levels of debugging are available:

Debug Level	Syntax	Message
1	-d	Only error messages.
2	-d -d	Level 1 messages and messages indicating potential errors.
3	-d -d -d ...	Level 1 and level 2 and general information messages.

If the debug level is set to >0 and if the **syslogd** daemon is running, then all debug messages are printed in the **syslogd** log file.

-g Keeps the same gateway IP address that is in **bootp** request in **bootp** reply.

-s Runs the **bootpd** command in a stand-alone configuration. This mode is used for large network installations with many hosts. In this case, the **-t** flag has no effect since the **bootpd** command never exits.

-t Specifies a different time-out value in minutes, such as `-t20`. A time-out value of `0` means forever. The default time-out value is 15 minutes.

ConfigFile Specifies the configuration file. The default configuration file is **/etc/bootptab**.

DumpFile Specifies the file into which the **bootpd** daemon dumps a copy of the bootp server database. The default dump file is **/etc/bootpd.dump**.

Examples

1. To start the bootpd daemon in a stand-alone mode, enter the following:

   ```
   /usr/sbin/bootpd -s
   ```

2. To start the **bootpd** daemon in a stand-alone mode with a debug level of 3, with a configuration file of **/etc/newconfig**, and a dump file of **/etc/newdumpfile**, enter the following:

   ```
   /usr/sbin/bootpd -s -d -d -d /etc/newconfig /etc/newdumpfile
   ```

Files

/etc/bootpd.dump The default **bootpd** dumpfile
/etc/bootptab The default **bootpd** configuration file.
/etc/services Defines sockets and protocols used for Internet services.

/etc/inetd.conf Contains the configuration information for the **inetd** daemon.

Related Information

The **x_add_nfs_fpe** command, **x_rm_fpe** command.

---

bootptodhcp Command

Purpose

To convert a BOOTP configuration file into a DHCP configuration file or to remove BOOTP configuration information for a particular host from the DHCP configuration file.

Syntax

To Convert a BOOTP Configuration File into a DHCP Configuration File

/usr/sbin/bootptodhcp [**-d** *DHCPFile*] [**-b** *BOOTPFile*]

To Remove a BOOTP Configuration Information From a DHCP Configuration File

/usr/sbin/bootptodhcp [**-d** *DHCPFile*] **-r** *HostName*]

Description

The **bootptodhcp** command has two functions. The first is to translate a BOOTP configuration file into a DHCP configuration. The default command with no arguments translates the **/etc/bootptab** file. The filenames may be changed by using the **-b** or **-d** flags to specify a different file names.

The second function of the **bootptodhcp** command is the removal of a BOOTP client's information from a DHCP configuration file. The **-r** flag specifies which client to remove from the file. If the **-d** flag is not used.

Flags

-b *BOOTPFile*	Specifies the BOOTP configuration file. The default is **/etc/bootptab**.
-d *DHCPFile*	Specifies the DHCP configuration file.
-r *HostName*	Specifies the hostname of a BOOTP section to delete from the DHCP configuration file.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Security

Access Control: Any User

Files Accessed: Need appropriate access permissions for files

Files

/usr/sbin/bootptodhcp	Contains the **bootptodhcp** command.
/etc/bootptab	Contains the default configuration file for bootpd.

Related Information

The **dhcpsconf** command

DHCP Client Configuration File

DHCP Server Configuration File

bootp Configuration File

TCP/IP address and parameter Assignment - Dynamic Host Configuration Protocol in *Networks and communication management*

TCP/IP reference in *Networks and communication management*.

---

bosboot Command

Purpose

Creates boot image.

Syntax

For General Use:

bosboot -*Action* [**-d** *Device*] [**-***Options* ...]

To Create a Device Boot Image:

bosboot -a [**-d** *Device*] [**-p** *Proto*] [**-k** *Kernel*] [**-I** | **-D**] [**-I** *LVdev*] [**-L**] [**-M** { *Norm* | *Serv* | *Both* }] [**-T** *Type*] [**-b** *FileName*] [**-q**]

Description

The **bosboot** command creates the boot image that interfaces with the machine boot ROS (Read-Only Storage) EPROM (Erasable Programmable Read-Only Memory).

The **bosboot** command creates a boot file (boot image) from a RAM (Random Access Memory) disk file system and a kernel. This boot image is transferred to a particular media that the ROS boot code recognizes. When the machine is powered on or rebooted, the ROS boot code loads the boot image from the media into memory. ROS then transfers control to the loaded images kernel.

The associated RAM disk file system contains device configuration routines that make the machine's devices and file systems available. The RAM disk file system contains differing configuration files depending upon the boot device. A **mkfs** prototype file is supplied for each type of device. (See note 6 below.) Currently supported devices are:

- CD-ROM
- Disk
- Tape
- Network

A network device may be a token ring, Ethernet, or Fiber-Distributed Data Interface (FDDI) used to boot from a network boot server over a local area network (LAN).

The boot image varies for each type of device booted and is compressed to fit on certain media and to lessen real memory requirements. The boot logical volume must be large enough for the boot image.

In addition to creating a boot image, the **bosboot** command always saves device configuration data for disk. It does not update the list of boot devices in the NVRAM (nonvolatile random access memory). You can modify the list with the **bootlist** command.

The **bosboot** command is usually called during the Base Operating System installation and by the **updatep** command when the operating system is upgraded.

Notes:

1. You must have root user authority to use the **bosboot** command.
2. Do not reboot the machine if the **bosboot** command is unsuccessful with a message not to do so while creating a boot disk. The problem should be resolved and the **bosboot** command run to successful completion.
3. The **bosboot** command requires some space in the **/tmp** file system and the file system where the target image is to reside, if there is such an image.
4. The **bosboot** command requires that the specified physical disk contain the boot logical volume. To determine which disk device to specify, issue the following command:

   ```
   lsvg    -M    rootvg
   ```

   This command displays a map of all logical volumes. The default boot logical volume is **hd5**. Use the disk device that contains the boot logical volume.
5. When the device is not specified with the **-d** flag, the **bosboot** command assumes the default device is the disk the system is booted from. However, if the prototype file is specified with a **-p** flag, the device must also be specified with a **-d** flag.
6. The prototype file used by the **bosboot** command to build the RAM disk file system depends on the boot device and the hardware platform (**sys0**) type of the machine the boot image will run on.

   The hardware platform type is an abstraction which allows machines to be grouped according to fundamental configuration characteristics such as number of processors or I/O bus structure or both. Machines with different hardware platform types will have basic differences in the way their devices are dynamically configured at boot time. The hardware platform type **rs6k** in AIX 5.1 and earlier applies to all Micro Channel-based uni-processor models through AIX 5.1 only. The type **rs6ksmp** applies to all Micro Channel-based symmetric multi-processor models through AIX 5.1 only. The type **rspc** in AIX 5.1 and earlier applies to all ISA-bus models. As new models are developed, their hardware platform types will either be one of the aforementioned types or, if fundamental configuration differences exist, new types will be defined. Boot images for a given boot device type will generally be different for machines with different hardware platform types.

   ″The prototype file used by **bosboot** is constructed by starting with a copy of the base prototype file for the platform type and boot device (for example, **/usr/lib/boot/chrp.disk.proto**). Next the **bosboot** command looks at the **pcfg** file for the platform type being used (for example, **/usr/lib/boot/chrp.pcfg**). The **pcfg** file contains entries which bosboot uses in a template to search for proto extension files. These files, located in the directory **/usr/lib/boot/protoext**, provide extensions to the prototype file under construction. For example, if the platform type is **chrp** and the boot device is **disk**, and the file **/usr/lib/boot/protoext/chrp.pcfg** contains the following:

   ```
   scsi.
   chrp.
   chrp_lpar.
   fcp.
   graphics.
   ide.
   isa_sio.
   pci.
   ssa.
   sys.pci.
   tty.
   usbif.
   ```

The **bosboot** command will start with the base prototype file **/usr/lib/boot/chrp.disk.proto**, and search the directory **/usr/lib/boot/protoext** for any files that match the template **disk.proto.ext.scsi.***. The contents of these files are added to the prototype file under construction. Next, the contents of files matching the template **/usr/lib/boot/protoext/disk.proto.ext.scsi.*** are added to the prototype file under construction. This continues until all lines in the **pcfg** file have been processed. At this point the prototype file under construction is complete. The **bosboot** command passes this prototype file to the **mkfs** command which builds the RAM disk file system.

7. The prototype files used by the BOSBOOT command to build boot images are dependent on the boot device. In addition, the prototype files are dependent on the system device type (sys0) of the machine for which the boot image is built.

   This is reflected in the names of these prototype files:

   /usr/lib/boot/chrp.disk.proto

   /usr/lib/boot/chrp.cd.proto

   /usr/lib/boot/chrp.tape.proto

   /usr/lib/boot/network/chrp.ent.proto

   /usr/lib/boot/network/chrp.tok.proto

   /usr/lib/boot/network/chrp.atm.proto

   /usr/lib/boot/network/chrp.fddi.proto

   The system device type is an abstraction that allows machines to be grouped according to fundamental configuration characteristics, such as number of processors and I/O bus structure. The system device is the highest-level device in the system node, which consists of all physical devices in the system.

   Machines with different system device types have basic differences in the way their devices are dynamically configured at boot time.

   The **bosboot** command, by default, uses the prototype file that matches the system device type of the machine executing the command. The -p option allows you to specify the system device type of the prototype file.

Flags

-d *device* Specifies the boot device. This flag is optional for hard disk.

The following flags are action flags. One and only one flag must be specified.

-a Creates complete boot image and device.

The following flags are option flags:

-b *FileName* Uses specified file name as the boot image name. This flag is optional.

-D Loads the low level debugger. This flag is optional.

-I (upper case i) Loads and invokes the low-level debugger. This flag is optional.

-k *Kernel* Uses the specified kernel file for the boot image. This flag is optional, and if not specified, **/unix** is the default.

-L Enables lock instrumentation for MP systems. This flag has no effect on systems that are not using the MP kernel.

-l (lower case L) *LVDev* Uses target boot logical volume for boot image. This flag is optional.

-M *Norm|Serv|Both* Specifies the boot mode. The options are:

- **norm** Indicates normal mode.
- **serv** Indicates service mode.
- **both** Indicates both modes.

-p *Proto*	Uses the specified prototype file for the RAM disk file system. This flag is optional.
-q	Determines how much disk space is required in which file system to create the boot image. Boot image is not created. This flag is optional.
-T *Type*	Specifies the hardware platform type (see note 6). This causes the **bosboot** command to create a boot image for the hardware platform type specified. If the type is not specified, the **bosboot** command creates a boot image whose hardware platform type matches that of the currently running machine. This flag is optional.

Security

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

Examples

1. To create a boot image on the default boot logical volume on the fixed disk from which the system is booted, type:

   ```
   bosboot -a
   ```

2. To create a bootable image called **/tmp/tape.bootimage** for a tape device, type:

   ```
   bosboot -ad /dev/rmt0 -b /tmp/tape.bootimage
   ```

3. To create a boot image file for an Ethernet boot, type:

   ```
   bosboot -ad /dev/ent0
   ```

4. To create a token ring boot image for a machine whose hardware platform type is **chrp** while you are running on a machine whose hardware platform type is **chrp**, type:

   ```
   bosboot -ad /dev/tok -T chrp
   ```

Files

/usr/sbin/mkboot	Specifies boot creation routine.
/usr/lib/boot/chrp.disk.proto	Specifies the disk RAM file system template.
/usr/lib/boot/chrp.cd.proto	Specifies the CD-ROM RAM file system template.
/usr/lib/boot/chrp.tape.proto	Specifies the tape RAM file system template.
/usr/lib/boot/network/chrp.ent.proto	Specifies the Ethernet RAM file system template.
/usr/lib/boot/network/chrp.tok.proto	Specifies the token-ring RAM file system template.
/usr/lib/boot/network/chrp.atm.proto	Specifies the ATM file system template.
/usr/lib/boot/network/chrp.fddi.proto	Specifies the FDDI RAM file system template.

Related Information

The **bootlist** command, **mkboot** command, and **locktrace** command.

.Boot process in *Operating system and device management*.

---

bosdebug Command

Purpose

Enables, disables and/or displays the status information of the system.

Syntax

bosdebug [**-l** | **-D**] [**-l** <*file*>] [**-f**]

bosdebug [**-K on | off**]

bosdebug [**-o**]

bosdebug [**-L**]

bosdebug [**-b**]

bosdebug [**-R on | off**]

bosdebug [**-M**]

bosdebug [**-s sizelist**]

bosdebug [**-S**]

bosdebug [**-n sizelist**]

bosdebug [**-h**]

Description

The **bosdebug** command enables, disables, and/or displays the status of debugging features of the system.

-b Disables data collection of state information for backtracking faults. This information is useful for debugging certain kinds of kernel errors. Disabling state information data collection for backtracking faults can provide a slight performance improvement under certain rare workloads, but that disablement does not allow the preservation of data that might be critical for problem analysis.

-D Causes the kernel debug program to be loaded on each subsequent reboot.

-I Causes the kernel debug program to be loaded and invoked on each subsequent reboot.

-L Displays the current settings for the kernel debug program and the memory overlay detection system. Note that the settings shown will not take effect until after the next time that the **bosboot -a** and **shutdown -r** commands are run. This is the default.

-K on | off Sets the state of kernel extension allocation tracking.

-o Turns off all debugging features of the system.

-R on | off Activates or deactivates the real-time kernel option. When `-R on` is specified, the kernel proactively generates an extra interrupt to ensure rapid response to a cross-CPU preemption request when the preempting thread is considered a real-time thread. Without this extra interrupt (called an *MPC*), the preempted thread might continue to run uninterrupted until the next regularly scheduled timer tick, or generally up to 10 ms.

Threads running with a fixed priority policy are considered real time by default. If `RT_MPC=ON` is exported in the environment before a process is started, that process's threads are also considered real time. Note that while the extra MPC interrupts reduce preemption latency, they also add overhead. Consider this additional overhead before exporting `RT_MPC=ON` in the default environment.

-l <file> Loads a symbol file into kernel for the **kdb** debugger print facility. Loads the symbols immediately. Do not reboot. A symbol file to print LFS stuctures may be created as follows:

```
# echo '#include <sys/vnode.h>' > sym.c
# echo 'main() { ; }' >> sym.c
# cc -g -o sym sym.c -qdbxextra /* for 32 bit kernel */
# cc -g -q64 -o sym sym.c -qdbxextra /* for 64 bit kernel */
```

-f Flushes all the symbols (loaded through **-l** option) from kernel memory. Flushed immediately. Does not require a reboot.

-M Causes the memory overlay detection system to be enabled. Memory overlays in kernel extensions and device drivers will cause a system crash.

-s sizelist	Causes the memory overlay detection system to promote each of the specified allocation sizes to a full page, and allocate and hide the next subsequent page after each allocation. This causes references beyond the end of the allocated memory to cause a system crash. sizelist is a list of memory sizes separated by commas. Each size must be in the range from 16 to 2048, and must be a power of 2.
-S	Causes the memory overlay detection system to promote all allocation sizes to the next higher multiple of page size (4096), but does not hide subsequent pages. This improves the chances that references to freed memory will result in a crash, but it does not detect reads or writes beyond the end of allocated memory until that memory is freed.
-n sizelist	Has the same effect as the -s option, but works instead for network memory. Each size must be in the range from 32 to 2048, and must be a power of 2. This causes the **net_malloc_frag_mask** variable of the **no** command to be turned on during boot.
-h	Displays the usage message for this command.

Any changes made by this command will not take effect until the **bosboot** and **shutdown -r** commands have been run (except **-l** and **-f** option).

Security

RBAC Disclaimer

Related Information

The **bosboot** command and **shutdown** command.

---

bs Command

Purpose

Compiles and interprets modest-sized programs.

Syntax

bs [*File* [*Arguments*]]

Description

The **bs** command is a compiler and interpreter for interactive program development and debugging. To simplify program testing, it minimizes formal data declaration and file manipulation, allows line-at-a-time debugging, and provides trace and dump facilities and run-time error messages.

The optional parameter *File* specifies a file of program statements that you create and that the compiler reads before it reads from standard input. Statements entered from standard input are normally executed immediately (see **compile** and **execute** statement syntax). By default, statements read from *File* are compiled for later execution.

Unless the final operator is assignment to a variable, the result of an immediate expression statement is displayed.

Additional command line *Arguments* can be passed to the program using the built-in functions **arg** and **narg**.

Program lines must conform to one of the following formats:

```
statement
label statement
```

The interpreter accepts labeled statements only when it is compiling statements. A *label* is a name immediately followed by a colon. A label and a variable can have the same name. If the last character of a line is a **** (backslash), the statement continues on the following physical line.

A statement consists of either an expression or a keyword followed by zero or more expressions.

> **Note:** To avoid unpredictable results when using a range expression in the international environment, use a character class expression rather than a standard range expression.

Statement Syntax

break Exits the innermost **for** or **while** loop.

clear Clears the symbol table and removes compiled statements from memory. A **clear** is always executed immediately.

compile [*Expression*] Causes succeeding statements to be compiled (overrides the immediate execution default). The optional *Expression* is evaluated and used as a file name for further input. In this latter case, the symbol table and memory are cleared first. **compile** is always executed immediately.

continue Transfers control to the loop-continuation test of the current **for** or **while** loop.

dump [*Name*] Displays the name and current value of every global variable or, optionally, of the *Named* variable. After an error or interrupt, **dump** displays the number of the last statement and (possibly) the user-function trace.

exit [*Expression*] Returns to the system level. The *Expression* is returned as process status.

execute Changes to immediate execution mode (pressing the INTERRUPT key has the same effect). This statement does not cause stored statements to execute (see **run**).

for Performs repeatedly, under the control of a named variable, a statement or a group of statements using one of the following syntaxes:

```
for name=Expression Expression statement
next
```

OR

```
for name=Expression Expression
statement . . .
next
```

OR

```
for Expression, Expression, Expression statement
next
```

OR

```
for Expression, Expression, Expression
statement . . .
next
```

The first format specifes a single statement where the variable takes on the value of the first expression and then is increased by one on each loop until it exceeds the value of the second expression. You can use the second format to do the same thing , but you can specify a group of statements.

The third format requires an initialization expression followed by a test expression (such as true to continue) and a loop-continuation action expression. You can use the fourth format to do the same thing, but you can specify a group of statements. Use commas to separate the expressions in the third and fourth formats.

fun Defines a user-written function using the following syntax:

```
fun f ([a, . . . ]) [v, . . . ]
statement . . .
nuf
```

f specifies the function name, *a* specifies any parameters, and *v* identifies any local variables for the user-written function. You can specify up to 10 parameters and local variables; however, they cannot be arrays or associated with I/O functions. You cannot nest function definitions.

freturn Signals the failure of a user-written function. Without interrogation, **freturn** returns zero. (See the unary interrogation operator (**?**).) With interrogation, **freturn** transfers to the interrogated expression, possibly bypassing intermediate function returns.

goto *Name* Passes control to the compiled statement with the matching label of *Name*.

ibase *n* Sets the input base to *n*. The only supported values for *n* are 8**,** 10 (the default), and 16**.** Hexadecimal values 10-15 are entered as alphabetic characters a-f**.** A leading digit is required when a hexadecimal number begins with an alphabetic character (for example, f0a must be entered as 0f0a). **ibase** is always executed immediately.

if Performs a statement in one of the following syntaxes:

```
if Expression statement
[else
statement . . . ]
fi
```

OR

```
if Expression
statement . . .
 [else
statement . . . ]
fi
```

The first format specifies a single statement and the second format specifies a group of statements to continue using if the expression evaluates to nonzero. The strings 0 and "" (null) evaluate as zero.

In the second format, an optional **else** allows a group of statements to be performed when the first group is not. The only statement permitted on the same line with an **else** is an **if**. You can put **fis** only on the same line as another **fi**. You can combine **else** and **if** into **elif**. You can close an **if** . . . **elif** . . . [**else** . . .] sequence with a single **fi**.

include *Expression* Evaluates an *Expression* to the name of a file containing program statements. Such statements become part of the program being compiled. The **include** statements are always executed immediately. Do not nest **include** statements.

obase *n* Sets the output base to *n*. The only supported values for *n* are 8, 10 (the default), and 16**.** Hexadecimal values 10 through15 are entered as alphabetic characters a-f**.** A leading digit is required when a hexadecimal number begins with an alphabetic character (that is, f0a must be entered as 0f0a). Like **ibase**, **obase** is always executed immediately.

onintr Provides program control of interrupts using one of the following syntaxes:

```
onintr Label
```

OR

```
onintr
```

In the first format, control passes to the *Label* given, just as if a **goto** had been performed when **onintr** was executed. The effect of the **onintr** statement is cleared after each interrupt. In the second format, pressing INTERRUPT ends the **bs** program.

return [*Expression*] Evaluates the *Expression* and passes the result back as the value of a function call. If you do not provide an expression, the function returns zero.

run Passes control to the first compiled statement. The random number generator is reset. If a file contains a **run** statement, it should be the last statement; **run** is always executed immediately.

stop Stops execution of compiled statements and returns to immediate mode.

trace [*Expression*] Controls function tracing. If you do not provide an *Expression* or if it evaluates to zero, tracing is turned off. Otherwise, a record of user-function calls/returns will be written. Each return decreases by one the **trace** expression value.

while Performs repeatedly, under the control of a named variable, a statement or a group of statements using one of the following syntaxes:

```
while Expression statement
next
```

OR

```
while Expression
statement . . .
next
```

The **while** statement is similar to the **for** statement except that only the conditional expression for loop continuation is given.

!*cmd* Runs a command and then returns control to the **bs** program.

*Comment* Inserts a comment line.

Expression Syntax

Name Specifies a variable or, when followed immediately by a colon, a label. Names are composed of a letter (uppercase or lowercase) optionally followed by letters and digits. Only the first six characters of a name are significant. Except for names declared locally in **fun** statements, all names are global. Names can take on numeric (double float) values or string values or be associated with input/output (see the built-in function **open**).

*Name***(**[*Expression*[, *Expression*] . . .]**)** Calls function *Name* and passes to it the parameters in parentheses. Except for built-in functions, *Name* must be defined in a **fun** statement. Function parameters are passed by value.

*Name***[***Expression*[, *Expression*] . . . **]** References either arrays or tables (see built-in function table). For arrays, each expression is truncated to an integer and used as a specifier for the name. The resulting array reference is syntactically identical to a name; a [1,2] is the same as a [1] [2]. The truncated expressions must be values between 0 and 32,767.

Number Represents a constant numerical value. This number can be expressed in integer, decimal, or scientific notation (it can contain digits, an optional decimal point, and an optional **e** followed by a possibly signed exponent).

String Represents a character string delimited by ″ ″ (double quotation marks). Within the string, you can use the \ (backslash) as an escape character that allows the double quotation mark (\″), new-line character (\n), carriage return(\r), backspace (\b), and tab (\t) characters to appear in a string. When not immediately followed by these special characters, \ stands for itself.

(*Expression*) Alters the normal order of evaluation.

(*Expression***,** *Expression*[, *Expression*] . . . **) [***Expression***]** Specifies to use the bracketed expression outside the parentheses as a subscript to the list of expressions within the parentheses. List elements are numbered from the left, starting at zero. The following expression has the value of True if the comparison is true:

```
(False, True) [a == b]
```

Expression Operator Expression Converts the operands to numeric form before the operator is applied unless the operator is an assignment, concatenation, or relational operator.

Unary Operators

? *Expression*
Tests for the success of *Expression* rather than its value. This interrogation operator is useful for testing:

- The end of file
- Result of the **eval** built-in function
- Return from user-written functions (see **freturn**)

An interrogation trap (end of file, for example), causes an immediate transfer to the most recent interrogation, possibly skipping assignment statements or intervening function levels.

- *Expression*
Negates *Expression*.

++ *Name*
Increases by one the value of the variable (or array reference).

— *Name*
Decreases by one the value of the variable.

! *Expression*
Specifies the logical negation of *Expression*.

Note: Unary operators treat a null string as a zero.

Binary Operators (in increasing precedence)

=
Specifies the assignment operator. The left operand must be a name or array element. It acquires the value of the right operand. Assignment binds right to left; all other operators bind left to right.

_
Specifies the concatenation operator. (It is the underline character).

& |
Specifies logical AND, logical OR.

The result of:

Expression & Expression

is 1 (true) only if both of its parameters are non-zero (true); it is 0 (false) if one or both of its parameters are 0 (false).

The result of:

Expression | *Expression*

is 1 (true) if one or both of its expressions are non-zero (true); it is 0 (false) only if both of its expressions are 0 (false). Both operators treat a null string as a zero.

< <= > >= == !=
Specifies the relational operators:

- < for less than
- <= for less than or equal to
- > for greater than
- >= for greater than or equal to
- == for equal to
- != for not equal to

The relational operators return 1 if the specified relation is True; otherwise they return 0 (false). Relational operators at the same level extend as follows: **a>b>c** is the same as **a>b& b>c**. A string comparison is made if both operands are strings. The comparison is based on the collating sequence specified in the environment variable **LC_COLLATE**. The National Language Support Overview contains more information on this environment variable.

+ -
Specifies addition and subtraction.

*** / %**
Specifies multiplication, division, and remainder.

^
Specifies exponentiation.

Note: Binary operators treat a null string as a zero.

Functions Dealing With Arguments

arg(*i*) Returns the value of the *i*-th actual argument at the current function call level. At level zero, **arg** returns the *i*-th command-line argument. For example, **arg(0)** returns **bs**.

narg( ) Returns the number of arguments passed. At level zero, it returns the command line argument count.

Mathematical Functions

abs(*x*) Returns the absolute value of *x*.

atan(*x*) Returns the arc tangent of *x*.

ceil(*x*) Returns the smallest integer not less than *x*.

cos(*x*) Returns the cosine of *x*.

exp(*x*) Returns e raised to the power *x*.

floor(*x*) Returns the largest integer not greater than *x*.

log(*x*) Returns the natural logarithm of *x*.

rand( ) Returns a uniformly distributed random number between zero and one.

sin(*x*) Returns the sine of *x*.

sqrt(*x*) Returns the square root of *x*.

String Functions

size(*s*) Returns the size (length in characters) of *s*.

bsize(*s*) Returns the size (length in bytes) of *s*.

format(*f*, *a*) Returns the formatted value of *a*, *f* being a format specification string in the style of the **printf** subroutine. Use only the **%...f**,**%...e**, and **%...s** formats.

index(*x*, *y*) Returns a number that is the first position in *x* containing a character that any of the characters in *y* matches. 0 return if no match is found. For 2-byte extended characters, the location of the first byte is returned.

trans(*s*, *f*, *t*) Translates characters in the source string *s* which match characters in *f* into characters having the same position in *t*. Source characters that do not appear in *f* are copied unchanged into the translated string. If string *f* is longer than *t*, source characters that match characters found in the excess portion of *f* do not appear in the translated string.

substr(*s*, *Start*, *Length*) Returns the substring of *s* defined by *Start* position in characters and *Length* in characters.

match(*String*, *Pattern*)
mstring(*n*)

Returns the number of characters in *string* that match *pattern*. The characters **.**, *****, **$**, **[**, **]**, **^** (when inside square brackets), **** (and **\)** have the following special meanings:

Note: See **ed** for a more detailed discussion of this special notation.

. Matches any character except the new-line character.

***** Matches zero or more occurrences of the pattern element that it follows. For example, .* matches zero or more occurrences of any character except the new-line character.

$ Specifies the end of the line.

[.-.] Matches any one character in the specified range ([.-.]) or list ([. . .]), including the first and last characters.

[^ .-.]

[^ . . .]
Matches any character except the new-line character and any remaining characters in a range or list. A circumflex (^) has this special meaning only when it immediately follows the left bracket.

[].-.]

[] . . .]
Matches] or any character in the list. The right square bracket does not terminate such a list when it is the first character within it (after an initial ^ , if any).

\(. . . \)
Marks a substring and matches it exactly.The pattern must match from the beginning of the string and the longest possible string. Consider, for example:

```
match ('a123ab123',".*\([a-z]\)") = 6
```

In this instance, .* matches `a 123a` (the longest string that precedes a character in the range a-z); `\ ([a-z]\)` matches `b`, giving a total of six characters matched in the string. In an expression such as `[a-z]`, the minus means ″through,″ according to the current collating sequence.

A collating sequence may define equivalence classes for use in character ranges. See the ″International Character Support Overview″ for more information on collating sequences and equivalence classes.

The **mstring** function returns the *n*th substring in the last call to **match** (*n* must be between 1 and 10 inclusive).

File-Handling Functions

open*(Name, File, Mode)*

close*(Name*) Specifies the name, file type and file mode. *Name* must be a legal variable name (passed as a string). After a close, the name becomes an ordinary variable. For open, the *File* can be one of the following:

- 0 for standard input
- 1 for standard output
- 2 for error output
- A string representing a file name
- A string beginning with an !, which represents a command to be run (using ″sh -c″)

Mode must be specified with an r for read, w for write, W for write without the new line character, or a for append. The initial associations are:

- open (″get″, 0, ″r″)
- open (″put″, 1, ″w″)
- open (″puterr″, 2, ″w″)

access*(p, m)* Performs the access subroutine. Parameter *p* is the path name of a file; *m* is a bit pattern representing the requested mode of access. This function returns a 0 if the system request is permitted, -1 if it is denied.

ftype(*s)* Returns a single character indicating file type: f for regular file, p for FIFO (named pipe), d for directory, b for block special, or c for character special.

Table Functions

table*(Name, Size)* Specifies an associatively accessed, one-dimensional array. ″Subscripts″ (called keys) are strings (numbers are converted). *Name* must be a **bs** variable name (passed as a string). *Size* sets the minimum number of elements to be allocated. On table overflow, **bs** writes an error message.

item*(Name, i)*

key( ) Accesses table elements sequentially instead of in an orderly progression of key values. Where the item function accesses values, the key function accesses the ″subscript″ of the previous item call. Do not quote *Name*.

Since exact table sizes are not defined, the interrogation operator should be used to detect end-of-table; for example:

```
table("t",100)


.
.
.
#If word contains "parity", the following expression
#adds one to the count of that word:
++t[word]
.
.
.
#To display the key/value pairs:
for i = 0, ? (s = item (t, i)), ++i if key( ) put = key
( )_":"_s
```

iskey*(Name, Word)* Tests whether the key word exists in the table name and returns one for true, zero for false.

Miscellaneous Functions

eval*(string)*

Specifies to evaluate the string parameter as an expression. The function is handy for converting numeric strings to numbers. **eval** can also be used as a crude form of indirection, as in:

```
name = "x,y,z"
eval("++"_name)
```

which increments the variable ″x,y,z″. In addition, when **eval** is preceded by ? (interrogation operator), you can control **bs** error conditions. For example:

```
?eval ("open(\"X\",\"XXX\", \"r\")")
```

returns the value zero if there is no file named ″XXX″ (instead of halting your program). The following performs a **goto** to the label ″L:″ (if it exists):

```
label = "L:"
if! (?eval ("goto"_label))puterr="no label"
```

plot(*request, args)*

Produces output on devices recognized by the **tplot** command. Some requests do not apply to all plotters. All requests except 0 and 12 are implemented by piping characters to **tplot**.

The call requests are as follows:

plot(0, *term***)**
Causes further plot output to be piped into **tplot** with a flag of **-T***term.*

plot(1) Erases the plotter.

plot(2, *string***)**
Labels the current point with *string*

plot(3, *x1, y1, x2, y2***)**
Draws the line between *(x1, y1*) and (*x2, y2*).

plot(4, *x, y, r***)**
Draws a circle with center(*x, y*) and radius *r*.

plot(5, x*1, y1, x2, y2, x3, y3***)**
Draws an arc (counterclockwise) with center *(x1, y1)*, and end points *(x2,y2)* and *(x3, y3).*

plot(6) Not implemented.

plot(7, *x, y***)**
Makes the current point at *(x, y).*

plot(8, *x***, y)**
Draws a line from the current point to *(x ,y).*

plot(9, *x, y***)**
Draws a point at *(x, y).*

plot(10, string)
Sets the line mode to string

plot(11, *x1, y1, x2, y2***)**
Makes *(x1, y1)* the lower left corner of the plotting area and (x2, y2) the upper right corner of the plotting area.

plot(12, *x1, y1, x2, y2***)**
Causes subsequent *x(y*) coordinates to be multiplied by *x1 (y1)* and then added to *x2 (y2)* before they are plotted. The initial scaling is **plot(12, 1.0, 1.0, 0.0, 0.0).**

last ( )

Returns, in immediate mode, the most recently computed value.

Example

To execute the bs command and direct the result to a file called output, enter:

```
bs < input.n > output
```

OR

```
bs input.n > output
```

Related Information

The **ksh** command.

The **access** subroutine, **printf** subroutine.

The National Language Support Overview for Programming in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

---

bsh Command

Purpose

The **bsh** command invokes the Bourne shell.

Syntax

bsh [**-i**] [**-r**] [{ **+** | **-** } { [**a**] [**e**] [**f**] [**h**] [**k**] [**n**] [**t**] [**u**] [**v**] [**x**] }] [**-c** *String* | **-s** | *File* [*Parameter*]]

> **Note:** Preceding a flag with a **+** (plus sign) rather than a **-** (minus sign) turns it off.

Description

The **bsh** command invokes the Bourne shell, an interactive command interpreter and command-programming language. The shell carries out commands either interactively from a terminal keyboard or from a file.

For more information about the Bourne shell, see ″Bourne shell″ in *Operating system and device management*.

Flags

The Bourne shell interprets the following flags only when the shell is invoked at the command line.

> **Note:** Unless you specify either the **-c** or **-s** flag, the shell assumes that the next parameter is a command file (shell script). It passes anything else on the command line to that command file. See the discussion of positional parameters in ″Variable substitution in the Bourne shell″ in *Operating system and device management* for more information.

-a Marks for export all variables to which an assignment is performed. If the assignment precedes a command name, the export attribute is effective only for that command's execution environment, except when the assignment precedes one of the special built-in commands. In this case, the export attribute persists after the built-in command has completed. If the assignment does not precede a command name, or if the assignment is a result of the operation of the **getopts** or **read** command, the export attribute persists until the variable is unset.

-c *String* Runs commands read from the *String* variable. Sets the value of special parameter `0` from the value of the *String* variable and the positional parameters (`$1`, `$2`, and so on) in sequence from the remaining *Parameter* operands. The shell does not read additional commands from standard input when you specify this flag.

-e Exits immediately if all of the following conditions exist for a command:
- It exits with a return value greater than 0.
- It is not part of the compound list of a **while**, **until**, or **if** command.
- It is not being tested using AND or OR lists.
- It is not a pipeline preceded by the ! (exclamation point) reserved word.

-f Disables file name substitution.

-h Locates and remembers the commands called within functions as the functions are defined. (Normally these commands are located when the function is executed; see the **hash** command.)

-i Makes the shell interactive, even if input and output are not from a workstation. In this case the shell ignores the **TERMINATE** signal, so that the **kill 0** command does not stop an interactive shell, and traps an **INTERRUPT** signal, so you can interrupt the function of the **wait** command. In all cases, the shell ignores the **QUIT** signal.

-k Places all keyword parameters in the environment for a command, not just those preceding the command name.

-n Reads commands but does not execute them. The **-n** flag can be used to check for shell-script syntax errors. An interactive shell may ignore this option.

-r Invokes the restricted shell. Using this flag is the same as issuing the **Rsh** command, except the shell enforces restrictions when reading the **.profile** files.

-s Reads commands from standard input. Any remaining parameters specified are passed as positional parameters to the new shell. Shell output is written to standard error, except for the output of built-in commands.

-t Exits after reading and executing one command.

-u Treats an unset variable as an error and immediately exits when performing variable substitution. An interactive shell does not exit.

-v Displays shell input lines as they are read.

-x Displays commands and their arguments before they are executed.

Note: Using a + (plus sign) rather than a - (minus sign) unsets flags. The $- special variable contains the current set of flags.

Files

/usr/bin/bsh	Specifies the path name to the Bourne shell.
/usr/bin/Rsh	Specifies the path name to the restricted Bourne shell, a subset of the Bourne shell.
/tmp/sh*	Contains temporary files that are created when a shell is opened.

Related Information

The **env** command, **sh** command, **Rsh** command.

The **/etc/passwd** file, **null** special file, **environment** file.

The **profile** file format.

Bourne shell in *Operating system and device management*.

Bourne shell built-in commands in *Operating system and device management*.

Variable substitution in the Bourne shell in *Operating system and device management*.

bterm command

Purpose

Emulates terminals in bidirectional (BIDI) mode.

Syntax

bterm [**-maps** *Map*] [**-help**] [**-keywords**] [**-nobidi**] [**-symmetric**] [**-autopush**] [**-or** *Orientation*] [**-text** *TextType*] [**-nss** *NumShape*] [**-csd** *CharShape*] [**-tail**] [**-nonulls**]

Description

The **bterm** command emulates the IBM 3151, VT220, HFT and other terminals. It operates in BIDI mode on ASCII terminals. This command creates a BIDI shell that can run any BIDI application. You cannot initiate the **bterm** command recursively from within itself.

The maps that determine the keyboard mapping and the symmetric swapping of characters are specified by the **-maps** flag. You can specify other BIDI behaviors using the flags available to the **bterm** command or by setting them in the defaults files. Such behaviors include the default text mode, the default screen orientation, the default mode of Arabic character shaping, the default shape of numerals, whether the Symmetric Swapping mode is enabled and whether the Autopush mode is enabled or not. The behaviors specified with flags take precedence over the behaviors set in the defaults files.

The default files are searched in the following order:

1. The **.Bidi-defaults** file is searched for in your home directory.
2. If the file is not found, the **bterm** command searches for the **BTerm** resource file in the **/usr/lib/nls/bidi/$LANG/app-defaults** file.

Flags

-autopush
Enables the Autopush mode in visual text mode.

-csd *CharShape*
Specifies the shape of Arabic characters. The *CharShape* variable can be one of the following options:

- **automatic**
- **isolated** (visual text mode only)
- **initial** (visual text mode only)
- **middle** (visual text mode only)
- **final** (visual text mode only)
- **passthru**

The default is automatic shaping.

-help
Lists the available parameters and their syntax.

-keywords
Lists the keywords available in defaults file.

-maps *Map*
Specifies the map used for keyboard mapping and symmetric swapping of characters. Each language has a different map, and the available options for the *Map* variable are in the **/usr/lib/nls/bidi/maps** directory. You must specify the environment variable **BIDIPATH** as follows:

```
export BIDIPATH=/usr/lib/nls/bidi
```

-nobidi
Disables the BIDI mode.

-nonulls
Initializes the screen with spaces instead of nulls.

-nss *NumShape* Specifies the shape of the numerals. Specify one of the following options for the *NumShape* variable:

- **bilingual**
- **hindi**
- **arabic**
- **passthru**

The default is **bilingual**.

-or *Orientation* Specifies screen orientation. The *Orientation* variable can be either **LTR** or **RTL**. The default is **LTR**.

-symmetric Enables the Symmetric Swapping mode.

-tail Writes the ″seen,″ ″sheen,″ ″sad,″ and ″dad″ characters of the Arabic language in two cells instead of one cell.

-text *TextType* Specifies the type of data stream. The *TextType* variable can be either **implicit** or **visual**. The default is **implicit**.

Key Combinations

To change the BIDI settings using key combinations, press the Ctrl+X key sequence to enter a BIDI command mode. Any key you type after this key sequence is interpreted as a BIDI command. Invalid keys sound a beep and exit the BIDI command mode. The following keys are valid BIDI commands:

Key	Purpose
r	Reverses the screen orientation.
n	Sets the language keyboard layer to National.
l	Sets the language keyboard layer to Latin.
a	Toggles the automatic shaping variable option of the Arabic characters (valid also for Implicit mode).
t	Displays the status.
space	Enters a required space (RSP).

For implicit mode only:

Key	Purpose
c	Toggles the column heading mode.

For visual mode only:

Key	Purpose
s	Initiates the Push mode.
e	Terminates the End Push mode.
p	Toggles the Autopush mode.
f	Shapes Arabic characters in their final forms.
i	Shapes Arabic characters in their initial forms.
b	Shapes Arabic characters in the Passthru mode.
o	Shapes Arabic characters in their isolated forms.
m	Shapes Arabic characters in their middle forms.

.Bidi-defaults Keywords

Use the following keywords to set the defaults for the **bterm** command.

.Bidi-defaults Keywords

Keywords	Value/Effect	
fScrRev	**on**	Screen reverse function key is enabled.
	off	Screen reverse function key is disabled.
fRTL	**on**	RTL keyboard layer function key is enabled.
	off	RTL keyboard layer function key is disabled.
fLTR	**on**	LTR keyboard layer function key is enabled.
	off	LTR keyboard layer function key is disabled.
fPush	**on**	Push function key is enabled.
	off	Push function key is disabled.
fEndPush	**on**	End Push function key is enabled.
	off	End Push function key is disabled.
fAutoPush	**on**	AutoPush function key is enabled.
	off	AutoPush function key is disabled.
fASD	**on**	Automatic Shape Determination function key is enabled.
	off	Automatic Shape Determination function key is disabled.
fShapeIS	**on**	Isolated Shape function key is enabled.
	off	Isolated Shape function key is disabled.
fShapeIN	**on**	Initial Shape function key is enabled.
	off	Initial Shape function key is disabled.
fShapeM	**on**	Middle Shape function key is enabled.
	off	Middle Shape function key is disabled.
fShapeF	**on**	Final Shape function key is enabled.
	off	Final Shape function key is disabled.
textType	**implicit**	Type of data stream is set to Implicit.
	visual	Type of data stream is set to Visual.
orientation	**LTR**	Left-to-right default screen orientation.
	RTL	Right-to-left default screen orientation.
symmetric	**on**	Symmetric Swapping enabled.
	off	Symmetric Swapping disabled.

.Bidi-defaults Keywords

Keywords	Value/Effect
numShape	**bilingual** Numeral shaping is set to bilingual. **hindi** Numerals are represented in Hindi. **arabic** Numeral shaping is set in Arabic/Hebrew. **passthru** Numerals are represented in passthru.
charShape	**automatic** Arabic characters are shaped automatically. **passthru** Arabic characters are displayed in passthru mode. **isolated** Arabic characters are displayed in isolated mode. **initial** Arabic characters are displayed in initial mode. **final** Arabic characters are displayed in final mode. **middle** Arabic characters are displayed in middle mode.
maps	Specifies the page code directory to be used for Keyboard. layering, input, output and symmetric character swapping.
expandTail	**on** Writes ″seen″-like characters and their tails on two cells. **off** Writes ″seen″-like characters and their tails on one cell.
nobidi	**on** Activates BIDI mode. **off** Turn off BIDI mode.
noNulls	**on** Replaces nulls by spaces. **off** Leaves nulls as null, no replacement of spaces.

Related Information

The **aixterm** command, the **telnet, tn, or tn3270** command.

Bidirectionality and Character Shaping in *AIX 5L Version 5.3 National Language Support Guide and Reference*.

bugfiler Command

Purpose

Automatically stores bug reports in specified mail directories.

Syntax

bugfiler [**-d**] [**-m** *MessageMode*] [**-b** *BugUserName*] [*MailDirectory*]

Description

The **bugfiler** command automatically intercepts bug reports, summarizes them, and stores them in the appropriate folders in the directory specified by the *MailDirectory* variable.

The mail delivery program starts the **bugfiler** command through a line in the **/etc/aliases** file. The line has the following format:

```
bugs:"|/usr/lib/bugfiler $HOME/bugstuff"
```

In the example, the bug reports are placed in the **$HOME/bugstuff** directory. If no directory is specified, the **bugfiler** command places the bug reports in the **$HOME/mail** default directory.

> **Note:** The **$HOME**/**mail** directory must be created for the **bugfiler** command to use as a default directory.

If the *BugUserName* is other than `bugs`, the entry in the **/etc/aliases** file should contain a **-b** *BugUserName* flag, as in the following example:

```
hadley:"|/usr/lib/bugfiler -b hadley"
```

In this example, `hadley` is declared the *BugUserName* and all bug reports are placed in the **/home/hadley/mail** default directory. All directories used by the **bugfiler** command must be owned by `hadley`.

The **bugfiler** command reads bug reports from standard input, checks the format of each report, then either sends a message acknowledging receipt ($HOME/*MailDirectory*/**.ack** file) or indicates improper format ($HOME/*MailDirectory*/**.format** file).

Improperly formatted bug reports are filed in the **errors** directory, which the **bugfiler** command creates as a subdirectory of the *MailDirectory* variable. Bug reports must be in the format found in the **/usr/lib/bugformat** file. Use the **sendbug** command to start the **/usr/lib/bugformat** file. The **bugfiler** command summarizes valid bug reports and files them in the folder specified in the `Index:` line of the report. The source directory name in the `Index:` line must match one of the directory names in the mail directory. The **bugfiler** command appends a line in the following format to the *MailDirectory*/**summary** file:

```
DirectoryName/MessageNumber IndexInformation SubjectInformation
```

> **Note:** The **bugfiler** command does not recognize forwarded mail. It notifies the forwarder, not the sender, unless a `Reply-To:` line is included in the header of the report.

Format of Bug Reports

Bug reports must be submitted in ARPA RFC 822 format. The **sendbug** command contains information to compose and mail bug reports in the correct format.

The reports require the following header lines for proper indexing:

`Date:`	Followed by the date the **bugfiler** command receives the report.
`From:`	Followed by the valid return address of the sender.
`Subject:`	Followed by a short summary of the problem.
`Index:`	Followed by the path of the source directory and source file, the version number, and optionally, the **Fix** keyword.

The body of the bug report requires the following lines:

`Description:`	Followed by a detailed description of the problem, suggestion, or complaint.
`Repeat-By:`	Followed by a procedure to repeat the problem.
`Fix:`	Followed by a **diff** command comparing the old and new source files or a description of how to solve the problem. Include the `Fix:` line only if the **Fix** keyword is specified in the `Index:` line.

Redistribution of Bug Reports

Bug reports can be redistributed according to index information in the *MailDirectory*/**.redist** file. The *MailDirectory*/**.redist** file is examined for a line beginning with an index name followed by a tab. Following the index name and tab is a comma-separated list of mail addresses to receive copies of bug reports. If the list continues on multiple lines, each line but the last must end with a \ (backslash). The following is an example of index information in the **.redist** file:

```
myindex   joe@hal,mary@mercutio,martha@banquo,sarah@mephisto,\
dee@hamlet,dewayne@ceasar
```

Flags

-b *BugUserName*	Specifies a new user ID. If the **-b** *BugUserName* flag is not specified, the **bugfiler** command defaults to the login name.
-d	Sets debugging on. When the **-d** flag is specified, the **bugfiler** command sends error messages to standard output.
-m *MessageMode*	Sets message protection. The **-m** *MessageMode* flag specifies file permissions, using hexadecimal format, for all files that the **bugfiler** command creates.

Examples

1. The syntax of the **bugfiler** command when used with all three flags and a specified *MailDirectory* variable is as follows:

   ```
   hadley:"|/usr/lib/bugfiler -d -m 755 -b hadley
   /home/hadley/bugdir"
   ```

   When placed in the **/etc/aliases** file, this line starts debugging, sets file permissions to `rwxr-xr-x`, declares `hadley` as the *BugUserName*, and specifies the **/home/hadley/bugdir** directory.

2. The following is an example of a bug report:

   ```
   Date: Mon, 27 Nov 89 11:26:15 -600
   From: a@B
   Subject: Read not setting errno correctly
   Index: LFS/rdwr.c workstation 3.1

   Description: Read not setting errno correctly

   Repeat-By: Start an NFS daemon and it receives errors. Errno is
   zero.
   ```

Files

/etc/ aliases	Contains system-wide aliases for the mail transport system.
usr/sbin/sendmail	Contains the mail delivery program.
MailDirectory/**summary**	Contains the bug report summaries.
BugUserName/*MailDirectory*/**.ack**	Contains the message sent in acknowledgment.
BugUserName/*MailDirectory*/**.format**	Contains the message sent when format errors are detected.
MailDirectory/**.redist**	Contains the redistribution list for bug reports.

Related Information

The **sendbug** command.

Mail management in *Networks and communication management*.

burst Command

Purpose

Divides a message into separate, new messages.

Syntax

burst [**+***Folder*] [*Messages*] [**-inplace**] [**-noinplace**] [**-quiet**] [**-noquiet**] [**-verbose**] [**-noverbose**]

Description

The **burst** command allows you to divide a message into multiple, new messages. The **burst** command operates on digests, messages forwarded by the **forw** command, and blind carbon copies sent by the **forw** and **send** commands. Messages created using the **burst** command are numbered consecutively, beginning with the next highest number in the specified folder.

The **burst** command can create about 1000 messages from a single message. However, the **burst** command generally does not place a specific limit on the number of messages in a folder after bursting is complete.

The **burst** command uses encapsulation boundaries to determine where to separate the encapsulated messages. If an encapsulation boundary is located within a message, the **burst** command may split that message into two or more messages.

By default, the first message extracted from the first digest becomes the current message. If the **-inplace** flag is specified, the first new message becomes the current message.

Flags

+*Folder*
Specifies the folder containing the message to divide. By default, the system uses the current folder.

-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.

-inplace
Replaces each digest with a table of contents for the digest, places the messages contained in each digest directly after the digest's table of contents, and renumbers all subsequent messages in the folder to make room for the messages from the divided digest.

Attention: The **burst** command does not place text displayed after the last encapsulated message in a separate message. When you specify the **-inplace** flag, the **burst** command loses this trailing text. In digests, this text is usually an End-of-Digest string. However, if the sender appended remarks after the last encapsulated message, the **burst** command loses these remarks.

Messages Specifies the messages that you want to divide. This parameter may specify several messages, a range of messages, or a single message. Use the following references to specify messages:

Number
Number of the message. When specifying several messages, separate each number with a comma. When specifying a range, separate the first and last number in the range with a hyphen.

Sequence
A group of messages specified by the user. Recognized values include:

all All messages in the folder.

cur or . (period)
Current message. This is the default.

first First message in a folder.

last Last message in a folder.

next Message following the current message.

prev Message preceding the current message.

-noinplace Preserves each digest. This is the default.

-noquiet Reports information about messages not in digest format. This flag is the default.

-noverbose Prevents reporting of the actions the **burst** command performs while dividing the digests. This flag is the default.

-quiet Prevents reporting of information about messages not in digest format.

-verbose Reports the actions the **burst** command performs while dividing a digest.

Security

RBAC Disclaimer

Profile Entries

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

`Current-Folder:`	Sets the default current folder.
`Msg-Protect:`	Sets the protection level for your new message files.
`Path:`	Specifies a user's MH directory.

Examples

1. The user receives message `5` from `mickey@mouse` containing several messages in digest form:

```
5+ 03/02 mickey@mouse
6+ 03/02 disney@world
```

   To burst message `5` into several, separate messages, enter:

```
burst 5
5+ 03/02 mickey@mouse
6 03/02 disney@world
7 first message in digest
8 second message in digest
9 third message in digest
```

   The resulting new messages are appended to the end of the folder. Message `5` remains intact and still contains all four messages.

2. To burst message `5` using the **-inplace** flag, enter:

```
burst 5 -inplace
5+ 03/02 mickey@mouse
6 first message in digest
7 second message in digest
8 third message in digest
9 03/02 disney@world
```

The resulting new messages are placed immediately after the digest, and the **burst** command renumbers all the messages that follow. Message 5 now contains only the header and text of the forwarded message.

Files

$HOME/.mh_profile	Contains the MH user profile.
/usr/bin/burst	Contains the executable form of the **burst** command.

Related Information

The **forw** command, **inc** command, **msh** command, **packf** command, **send** command, **show** command.

The **mh_alias** file format, **mh_profile** file format.

Mail applications in *Networks and communication management*.

---

cachefslog Command

Purpose

Controls the logging of a cache file system.

Syntax

cachefslog [**-f***LogFile* |**-h**] *Cachefs_Mount_Point*

Description

The **cachefslog** command displays where CacheFS statistics are being logged. Optionally, it sets where CacheFS statistics are being logged, or it halts logging for a cache specified by *Cachefs_Mount_Point*. The *Cachefs_Mount_Point* argument is a mount point of a cache file system. All file systems cached under the same cache as *Cachefs_Mount_Point* are logged.

Flags

-f *LogFile*	Specifies the log file to be used. **Note:** You must have root authority in order to use this flag.
-h	Halts logging. **Note:** You must have root authority in order to use this flag.

Exit Status

The following exit values are returned:

0	success
non-zero	an error has occurred.

Examples

1. To checks if the directory **/home/sam** is being logged, type:

   ```
   cachefslog /home/sam
   ```

   The system displays the following:

   ```
   not logged: /home/sam
   ```

2. To change the *logfile* of **/home/sam** to **/var/tmp/samlog**, type:

   ```
   cachefslog -f /var/tmp/samlog /home/sam
   ```

   The system displays the following:

   ```
   /var/tmp/samlog: /home/sam
   ```

3. To halt logging for the **/home/sam directory**, type:

   ```
   cachefslog -h /home/sam
   ```

   The system displays the following:

   ```
   not logged: /home/sam
   ```

Files

/usr/sbin/cachefslog	Contains the **cachefslog** command.

Related Information

The **cachefsstat** command, **cachefswssize** command, and **cfsadmin** command.

---

cachefsstat Command

Purpose

Displays information about a cache file system.

Syntax

cachefsstat [**-z**] [*path...*]

Description

The **cachefsstat** command displays statistical information about the cache file system mounted on *path*. The statistical information includes cache hits and misses, consistency checking, and modification operations. If *path* is not specified, all mounted cache file systems are used. **cachefsstat** can also be used to reinitialize this information (see **-z** flag).

The statistical information includes the following:

`hit rate`	The percentage of cache hits over the total number of attempts, followed by the actual numbers of hits and misses.
`consistency checks`	The number of consistency checks performed, followed by the number that passed, and the number that failed.
`modifies`	The number of modify operations, including, for example, writes and creates.

Flags

-z	Reinitializes, zeros, statistics. Execute **cachefsstat -z** before running **cachefsstat** again to gather statistics on the cache performance. This flag can only be use by the superuser. The statistics printed reflect those just before the statistics are reinitialized.

Exit Status

The following exit values are returned:

0	success
non-zero	an error has occurred.

Examples

1. To display the cache file system statistics of the **/home/sam** directory, type:

   ```
   cachefsstat /home/sam
   ```

   The system displays the following:

   ```
   cache hit rate: 73% (1234 hits, 450 misses) consistency checks: 700 (650 pass, 50 fail) modifies: 321
   ```

Files

/usr/sbin/cachefsstat	Contains the **cachefsstat** command.

Related Information

The **cachefslog** command, **cachefswssize** command, and **cfsadmin** command.

---

cachefswssize Command

Purpose

Displays the work space size for a cache file system.

Syntax

cachefswssize *LogFile*

Description

The **cachefswssize** command displays the work space size determined from *LogFile*. This includes the amount of cache space needed for each filesystem that was mounted under the cache, as well as a total.

Exit Status

The following exit values are returned:

0	success
non-zero	an error has occurred.

Examples

1. To display the work space size of the cache filesystems being logged in the file **/var/tmp/samlog**, type:

   ```
   cachefswssize /var/tmp/samlog
   ```

   The system displays similar to the following:

   ```
   /home/sam
                                  end size:    10688k
                             high water size:    10704k

   /foo
   ```

```
                                end size:      128k
                           high water size:      128k

    /usr/dist

                                end size:     1472k
                           high water size:     1472k

    total for cache

                             initial size:   110960k
                                end size:    12288k
                           high water size:    12304k
```

Files

/usr/sbin/cachefswssize	Contains the **cachefswssize** command.

Related Information

The **cachefslog** command, **cachefsstat** command, and **cfsadmin** command.

---

cal Command

Purpose

Displays a calendar.

Syntax

cal [[*Month*] *Year*]

Description

The **cal** command displays a calendar of the specified year or month.

The *Year* parameter names the year for which you want a calendar. Since the **cal** command can display a calendar for any year from 1 through 9999, you must enter the full year rather than just the last two digits. The *Month* parameter identifies the month for which you want the calendar. It can be a number from 1 (indicating January) to 12 (indicating December). If you specify neither the *Year* nor the *Month* parameter, the **cal** command displays the current month. If you specify only one parameter, the **cal** command assumes the parameter is the *Year* parameter and displays the calendar for the indicated year.

> **Note:** The **cal** command does not accept standard input.

The **cal** command uses the appropriate month and day names according to the locale settings. The ″National Language Support Overview for Programming″ in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* contains more information on the **LANG**, **LC_TIME**, **LC_ALL**, and **TZ** environment variables.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Examples

1. To display a calendar for February, 1994, at your workstation, enter:
   ```
   cal 2 1994
   ```
2. To print a calendar for 1994, enter:
   ```
   cal 1994 | qprt
   ```
3. To display a calendar for the year 84, enter:
   ```
   cal 84
   ```

Files

/usr/bin/cal	Contains the **cal** command.

Related Information

The **calendar** command.

National Language Support Overview in *AIX 5L Version 5.3 National Language Support Guide and Reference*.

Input and output redirection in *Operating system and device management*.

National Language Support Overview for Programming in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

---

calendar Command

Purpose

Writes reminder messages to standard output.

Syntax

calendar [**-**]

Description

The **calendar** command reads the **calendar** file and displays any line in the file that contains today's or tomorrow's date. The **calendar** file is user-created and must be in the same directory from which you run the **calendar** command. Typically, the **calendar** file resides in your home directory.

If you run the **calendar** command on a Friday, the **calendar** command displays all lines containing the dates for that Friday as well as the subsequent Saturday, Sunday, and Monday. The command does not recognize holidays.

The **calendar** command recognizes date formats such as *Month Day*, *Abbreviation Date*, and *MonthNumeral/Date.* Examples of these formats include December 7, Dec. 7 and 12/7. The **calendar** command also recognizes the special character * (asterisk) when followed by a / (slash). It interprets */7, for example, as signifying the seventh day of every month. The **calendar** command does not recognize formats such as 7/*, 7 December, 7/12, * 7 or DEC. 7.

If the system administrator has created a **calendar** file for all users, you can access this file by placing the following line at the beginning of your local **calendar** file:

```
#include <FileName>
```

The actual value of the *FileName* variable is determined by the system administrator. The name of this file does not have to be **calendar**. When you run the **calendar** command, it displays reminders that were stored in your local **calendar** file as well as those stored in the file specified by the *FileName* variable.

> **Note:** When the **calendar** file contains include statements, the **calendar** command runs the **calendar** file through the C preprocessor. To use include statements with the **calendar** file, the C preprocessor, which is contained in the **/usr/ccs/lib/cpp** file, must be installed on the operating system.

For you to get reminder service, your **calendar** file must have read permission for others. See the **chmod** command for information on setting permissions.

Flag

- Calls the **calendar** command for everyone having a **calendar** file in the home directory. The **calendar** command sends reminders using the **mail** command instead of writing the results to standard output.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

1. A typical **calendar** file might look like the following:

   ```
   */25 - Prepare monthly report
   Aug. 12 - Fly to Denver
   aug 23 - board meeting
   Martha out of town - 8/23, 8/24, 8/25
   8/24 - Mail car payment
   sat aug/25 - beach trip
   August 27 - Meet with Simmons
   August 28 - Meet with Wilson
   ```

   To run the **calendar** command, enter:

   ```
   calendar
   ```

   If today is Friday, August 24, then the **calendar** command displays the following:

   ```
   */25 - Prepare monthly report
   Martha out of town - 8/23, 8/24, 8/25
   8/24 - Mail car payment
   sat aug/25 - beach trip
   August 27 - Meet with Simmons
   ```

2. A **calendar** file that contains an include statement might look like the following:

   ```
   #include </tmp/out>
   1/21 -Annual review
   1/21 -Weekly project meeting
   1/22 *Meet with Harrison in Dallas*
   Doctor's appointment - 1/23
   1/23 -Vinh's wedding
   ```

   To run the **calendar** command, enter:

   ```
   calendar
   ```

   If today is Wednesday, January 21, then the **calendar** command displays the following:

```
Jan.21 Goodbye party for David
Jan.22 Stockholder meeting in New York
1/21 -Annual review
1/21 -Weekly project meeting
1/22 *Meet with Harrison in Dallas*
```

The results of the **calendar** command indicate the `/tmp/out` file contained the following lines:

```
Jan.21 Goodbye party for David
Jan.22 Stockholder meeting in New York
```

Files

$HOME/calendar	Contains the **calendar** command.
/usr/lib/calprog	Contains the program that determines dates.
/usr/ccs/lib/cpp	Contains the C preprocessor.
/etc/passwd	Contains basic user attributes.

Related Information

The **cal** command, **chmod** command, **mail** or **Mail** command.

File and directory access modes in *Operating system and device management* introduces file ownership and permissions to access files and directories.

The Input and output redirection in *Operating system and device management*.

---

cancel Command

Purpose

Cancels requests to a line printer.

Syntax

cancel { *JobID ...* | *PrinterName* }

or

cancel *JobID QueueName*

Description

The **cancel** command cancels line printer requests that were made by the **lp** command.

Specifying the following cancels the local print jobs:

- *JobID* cancels the print request, even if it is currently printing.
- *PrinterName* cancels the printing of your jobs on the specified queue. (If you have root user authority, all jobs on the queue are deleted.)

In AIX 4.3.2 and above, **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 `cancel 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.

And for remote print jobs, both the *JobID* and remote *QueueName* must be specified in order to explicitly cancel a job on a remote queue.

Notes:

1. You must have root-user authority, or be a member of the **print** group, to cancel print requests that were not submitted by your current ID.
2. The *JobID* must be a number.
3. If you enter `cancel -?`, the system displays the following error message:

   ```
   enq: (FATAL ERROR): 0781-048: Bad queue or device name: -?
   ```

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Security

RBAC Disclaimer

Files

/var/spool/qdaemon/*	Contains temporary copies of enqueued files.
/var/spool/lpd/qdir/*	Contains job description files for print jobs.
/usr/bin/cancel	Contains the command file.

Related Information

The **enable** command, **enq** command, **lp** command, **lpstat** command, **qcan** command.

Canceling a print job (qcan command) in *Operating system and device management*.

Printers, print jobs, and queues in *Operating system and device management*.

System V Print Subsystem cancel Command

Purpose

Cancel print requests

Syntax

cancel [*request-IDs*] [*printers*]

cancel -u *login-IDs* [*printers*]

Description

The **cancel** command allows users to cancel print requests previously sent with the **lp** command. The first form of **cancel** permits cancellation of requests based on their *request-ID*. The second form of **cancel** permits cancellation of requests based on the *login-ID* of their owner.

Canceling a print request

The **cancel** command cancels requests for print jobs made with the **lp** command. The first form allows a user to specify one or more *request-IDs* of print jobs to be canceled. Alternatively, the user can specify one or more *printers*, on which only the currently printing job will be canceled if it is the user's job.

The second form of **cancel** cancels all jobs for users specified in *login-IDs*. In this form the *printers* option can be used to restrict the printers on which the users' jobs will be canceled. Note that in this form, when the *printers* option is used, all jobs queued by the users for those printers will be canceled. A printer class is not a valid argument.

A user without special privileges can cancel only requests that are associated with his or her own login ID; To cancel a request, a user issues the command:

cancel -u *login-ID* [*printer*]

This command cancels all print requests associated with the *login-ID* of the user making the request, either on all printers (by default) or on the printer specified.

Administrative users with the appropriate privileges can cancel jobs submitted by any user by issuing the following types of commands:

cancel -u "*login-ID-list***"**
: Cancels all requests (on all relevant printers) by the specified users, including those jobs currently being printed. Double quotes must be used around *login-ID-list* if the list contains blanks. The argument *login-ID-list* 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

A remote job can be canceled only if it originated on the client system; that is, a server system can cancel jobs that came from a client, and a client system can cancel jobs it sent to a server.

cancel -u "*login-ID-list***"** *printer-1 printer-2 printer-n*
: Cancels all requests by the specified users for the specified printers, including those jobs currently being printed. (For a complete list of printers available on your system, execute the **lpstat -p** command.)

In any of these cases, the cancellation of a request that is currently printing frees the printer to print the next request.

Security

RBAC Environment and Trusted AIX

This command implements and can perform privileged operations. Only privileged users can run such privileged operations. For more information about authorizations and privileges, refer to the RBAC section: Privileged Command Database in *Security*. To review the list of privileges and the authorizations associated with this command, refer to the **/etc/security/privcmds** database.

Related Information

The **enable** command, **enq** command, **lp** command, **lpstat** command, **qcan** command.

Canceling a print job (qcan command) in *Operating system and device management*.

Printers, print jobs, and queues in *Operating system and device management*.

---

canonls Command

Purpose

Processes **troff** command output for the Canon LASER SHOT in LIPS III mode.

Syntax

canonls [**-eg***File*] [**-em***File*] [**-F***Directory*] [**-quietly**] [**-ug***File*] [**-um***File*] [*File* ...]

Description

The **canonls** command processes **troff** command output for the Canon LASER SHOT in LIPS III mode. This command is provided exclusively for Japanese language support.

The **canonls** command processes one or more files specified by the *File* parameter. If no file is specified, the **canonls** command reads from standard input.

The **canonls** command uses font files in the **/usr/lib/font/devcanonls** directory that have command names ending with **.out**. The **canonls** command does not produce correct output unless these files are provided.

Flags

-eg*File*	Specifies the Gothic font for the IBM Japanese extended character set. By default, the **canonls** command uses the Gothic font found in the **/usr/lib/X11/fonts/JP/IBM_JPN23G.snf** file.
-em*File*	Specifies the Mincho font for the IBM Japanese extended character set. By default, the **canonls** command uses the Mincho font found in the **/usr/lib/X11/fonts/JP/IBM_JPN23.snf** file.
-F*Directory*	Specifies a directory name as the place to find font files. By default, the **canonls** command looks for font files in the **/usr/lib/font/devvcanonls** directory.
-quietly	Suppresses all nonfatal error messages.
-ug*File*	Specifies the Gothic font for the user-defined characters of Japanese. By default, the **canonls** command uses the Gothic font found in the **/usr/lib/X11/fonts/JP/IBM_JPN23G.snf** file.
-um*File*	Specifies the Mincho font for the user-defined characters of Japanese. By default, the **canonls** command uses the Gothic font found in the **/usr/lib/X11/fonts/JP/IBM_JPN23.snf** file.

Example

To process the `reports` file for the Canon LASER SHOT printer, enter:

```
troff reports |canonls | qprt -dp
```

The **canonls** command first processes the output of the **troff** command, then sends the file to a print queue.

File

/usr/lib/font/devcanonls/*.out	Contains font files.

Related Information

The **troff** command.

The **troff** font file format.

---

captoinfo Command

Purpose

Converts a **termcap** file to a **terminfo** descriptor file.

Syntax

captoinfo [**-w***Number*] [**-v**] [**-V**] [**-1**] [*FileName*...]

Description

The **captoinfo** command converts a **termcap** source file to a **terminfo** source file and displays it on the screen. The **termcap** file format is an older format. The **termcap** and **terminfo** files differ mainly in the capability names and the entry syntax. Therefore, the **captoinfo** command only makes the syntactical transformations and vocabulary substitutions. The command also strips obsolete **termcap** capabilities such as nc, and 2-character **termcap** names like D3.

By default, the **captoinfo** command converts the **termcap** description for the terminal specified by the **TERM** environment variable. The command reads the description of the terminal from the **/etc/termcap** file and outputs a **terminfo**-style description. If you specify the *Filename* parameter, the command converts all the descriptions in the file to **terminfo** format.

You can redirect the output of the **captoinfo** command to a file.

Flags

-v Turns on the verbose mode.

-V Displays the version number.

-w*Number* Defines the line width of the **terminfo** entry. The **captoinfo** command fits as many **terminfo** fields in this width as is possible on the output line. A **terminfo** field consists of a capability name and a corresponding value. If you specify the **-w** flag, you must specify a *Number* parameter. By default, the line width is 60.

Notes:

1. If the width you specify is too small to contain even one field, the command displays one field per line.
2. If the width you specify is zero or negative, the line width will be set to 60.

-1 Displays one **terminfo** field per line.

Examples

1. To convert the **termcap** file **Wyse50.tc** to a **terminfo** file and see the results on the display, enter:
   ```
   captoinfo   Wyse50.tc
   ```
2. To convert the **termcap** file **Wyse50.tc** to a **terminfo** file and save the results, enter:
   ```
   captoinfo   Wyse50.tc   >   Wyse50.ti
   ```
3. To display one **terminfo** field per line and see more information, enter:
   ```
   captoinfo   -1   -v   Wyse50.tc
   ```

4. To produce a **terminfo** description of an ibm3101 terminal defined by the **TERM** environment variable, enter:

   ```
   captoinfo -w 40
   ```

   The **captoinfo** command converts the ibm3101 description in the **/etc/termcap** file into a **terminfo** description and produces a description with a 40 character width. The output of the command is similar to the following:

   ```
   ibm|ibm3101|3101|i3101|IBM 3101-10,
            am, xon,
            cols#80, lines#24,
            bel=^G, clear=\EK, cr=\r, cub1=\b,
            cud1=\n, cuf1=\EC,
            cup=\EY%p1%'\s'%+%c%p2%'\s'%+%c,
            cuu1=\EA, ed=\EJ, el=\EI,
            home=\EH, ht=\t, ind=\n,
            kcub1=\ED, kcud1=\EB, kcuf1=\EC,
            kcuu1=\EA,
   ```

Related Information

The **terminfo** file format.

The Curses Overview for Programming in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

---

capture Command

Purpose

Allows terminal screens to be dumped to a file.

Syntax

capture [**-a**] [*File*]

Description

The **capture** command allows a user to dump everything printed on the user's terminal to a file. The screen is printed to the file specified by the *File* parameter or to the **screen.out** file if no file is specified. If the **-a** flag is specified, the **capture** command appends the contents of the screen to the file.

In order to dump the screen to a file, the **capture** command creates a shell that emulates a VT100 terminal and maintains a record of what is being displayed on the screen. The **SHELL** environment variable determines the shell created. If the **SHELL** environment variable is not set, the **/usr/bin/bsh** shell is the default. The **TERM** environment variable is set to `TERM=vt100`. If, while running the **capture** command, the program asks for the terminal type in use, the user must enter `vt100`.

The Ctrl-P key sequence is the default keystroke to cause a screen dump to be performed. This can be changed by setting the **SCREENDUMP** environment variable to the 3-digit octal value of the desired screen dump key. For example, setting:

```
SCREENDUMP=014
```

changes the screen dump keystroke to Ctrl-L. Trying to set the **SCREENDUMP** environment variable by entering `^L` or `'\014'` results in an error message.

To stop the screen capture process, use the Ctrl-D key sequence or type `exit`. The system displays the message, `You are NO LONGER emulating a vt100 terminal`.

Flags

-a Appends the screen contents to the specified file or, if no file is specified, to the **screen.out** file.

Files

/usr/bin/capture Contains the **capture** command.

Related Information

The **bsh** command, **csh** command, **ksh** command, **script** command.

The Input and output redirection overview in *Operating system and device management* describes how the operating system processes input and output and how to use redirect and pipe symbols.

---

cat Command

Purpose

Concatenates or displays files.

Syntax

cat [**-q**] [**-r**] [**-s**] [**-S**] [**-u**] [**-n** [**-b**]] [**-v** [**-e**] [**-t**]] [**-** | *File* ...]

Description

The **cat** command reads each *File* parameter in sequence and writes it to standard output. If you do not specify a file name, the **cat** command reads from standard input. You can also specify a file name of **-** (dash) for standard input.

Attention: Do not redirect output to one of the input files using the redirection symbol, > (greater than symbol). If you do this, you lose the original data in the input file because the shell truncates the file before the **cat** command can read it. See ″Input and output redirection in the Korn shell or POSIX shell″ in *Operating system and device management* for more information.

Flags

-b Omits line numbers from blank lines, when specified with the **-n** flag.

-e Displays a $ (dollar sign) at the end of each line, when specified with the **-v** flag.

-n Displays output lines preceded by line numbers, numbered sequentially from 1.

-q Does not display a message if the **cat** command cannot find an input file. This flag is identical to the **-s** flag.

-r Replaces multiple consecutive empty lines with one empty line. This flag is identical to the **-S** flag.

-s Does not display a message if the **cat** command cannot find an input file. This flag is identical to the **-q** flag.
Note: Previously, the **-s** flag handled tasks now assigned to the **-S** flag.

-S Replaces multiple consecutive empty lines with one empty line. This flag is identical to the **-r** flag.

-t Displays tab characters as `^I` if specified with the **-v** flag.

-u Does not buffer output. The default is buffered output.

-v Displays nonprinting characters as visible characters, with the exception of tabs, new-lines, and form-feeds. ASCII control characters (octal 000–037) are printed as ^*n*, where *n* is the corresponding ASCII character in the octal range 100–137 (@, A, B, C,..., X, Y, Z, [, \,], ^, and _); the DEL character (octal 0177) is printed as ^?. Other non-printable characters are printed as M-*x*, where *x* is the ASCII character specified by the low-order seven bits.

When used with the -v option, the following options may be used:

-e A $ character will be printed at the end of each line prior to a new line.

-t Tabs will be printed as ^I and form feeds will be printed as ^L

The -e and -t options are ignored if the -v option is not specified.

- Allows standard input to the **cat** command.

Exit Status

This command returns the following exit values:

0 All input files were output successfully.

>0 An error occurred.

Examples

Attention: Do not redirect output to one of the input files using the redirection symbol, > (caret).

1. To display a file at the workstation, enter:

   ```
   cat notes
   ```

   This command displays the data in the notes file. If the file is more than one less than the number of available display lines, some of the file scrolls off the screen. To list a file one page at a time, use the **pg** command.

2. To concatenate several files, enter:

   ```
   cat section1.1 section1.2 section1.3 >section1
   ```

   This command creates a file named section1 that is a copy of section1.1 followed by section1.2 and section1.3.

3. To suppress error messages about files that do not exist, enter:

   ```
   cat  -q section2.1 section2.2 section2.3 >section2
   ```

   If section2.1 does not exist, this command concatenates section2.2 and section2.3. The result is the same if you do not use the **-q** flag, except that the **cat** command displays the error message:

   ```
   cat: cannot open section2.1
   ```

   You may want to suppress this message with the **-q** flag when you use the **cat** command in shell procedures.

4. To append one file to the end of another, enter:

   ```
   cat section1.4 >> section1
   ```

   The >> (two carets) appends a copy of section1.4 to the end of section1. If you want to replace the file, use the > (caret).

5. To add text to the end of a file, enter:

   ```
   cat >>notes
   Get milk on the way home
   Ctrl-D
   ```

This command adds `Get milk on the way home` to the end of the file called `notes`. The **cat** command does not prompt; it waits for you to enter text. Press the Ctrl-D key sequence to indicate you are finished.

6. To concatenate several files with text entered from the keyboard, enter:

```
cat section3.1 - section3.3 >section3
```

This command concatenates the file `section3.1` with text from the keyboard (indicated by the minus sign), and the file `section3.3`, then directs the output into the file called `section3`.

Files

/usr/bin/cat	Contains the **cat** command.

Related Information

The **cp** command, **ksh** command, **pcat** command, **pg** command, **pr** command.

Files in *Operating system and device management*.

Directories in *Operating system and device management*.

The Input and output redirection overview in *Operating system and device management*.

Input and output redirection in the Korn shell or POSIX shell in *Operating system and device management*.

Shells in *Operating system and device management*.

File and directory access modes in *Operating system and device management*.

---

catman Command

Purpose

Creates the cat files for the manual.

Syntax

catman [**-n** | **-p** | **-w**] [**-M** *Path*] [*Section...*]

Description

The **catman** command creates the preformatted versions of the online manual from the **nroff** command input files. The **catman** command examines each manual page and recreates those pages whose preformatted versions are missing or out of date. If any changes are made, the **catman** command recreates the **whatis** command database.

Flags

-M *Path* Updates manual pages located in the set of directories specified by the *Path* variable (the **/usr/share/man** directory by default).The *Path* variable has the form of a colon (:) separated by a list of directory names. For example:

```
'/usr/local/man:/usr/share/man'
```

If the environment variable **MANPATH** is set, its value is used for the default path. If the **nroff** command source file contains a line such as:

```
' .so manx/yyy.x'
```

a symbolic link is made in the **catx** directory to the appropriate preformatted manual page. This allows easy distribution of the preformatted manual pages among a group of associated machines using the **rdist** command.

The **nroff** command sources need not be distributed to all machines, thus saving the associated disk space.

For example, a local network of five machines (called mach1 through mach5) has mach3 with the manual page **nroff** command sources. Every night, mach3 runs the **catman** command by using the **cron** daemon and later runs the **rdist** command with a **distfile** file that looks like the following:

```
MANSLAVES = (mach1 mach2 mach4 mach5)
MANUALS = (/usr/share/man/cat[1-8no] /usr/share/man/whatis)
${MANUALS}    ->    ${MANSLAVES}
install -R;
notify root;
```

-n Prevents creation of the **whatis** command database.

-p Prints the names of the manual pages that need to be recreated or updated without recreating or updating them.

-w Reads the Berkeley Software Distribution (BSD) style manual pages in the **/usr/share/man/cat?/*.*** and **/usr/share/man/man?/*.*** files, and then reads the hypertext information bases and creates the **/usr/share/man/whatis** database.

Tip: If the base EN_US documentation fileset is installed on the system, set the ilocale to en_US to build a complete **whatis** database.

Examples

To update manual sections 1, 2, and 3 only, enter:

```
catman 123
```

Files

/usr/lbin/getNAME	Contains the command to create the **whatis** database.
/usr/share/man	Specifies the default manual directory location.
/usr/share/man/man?/*.*	Contains the raw (the **nroff** command input) manual sections.
/usr/share/man/cat?/*.*	Contains preformatted manual pages.
/usr/share/man/whatis	Contains the **whatis** command database.
/usr/lbin/mkwhatis	Contains the command script to make the **whatis** command database.

Related Information

The **man** command, **nroff** command and **rdist** command.

The **cron** daemon.

The **distfile** file.

cb Command

Purpose
Puts C source code into a form that is easily read.

Syntax
cb [**-s**] [**-l** *Length* | **-j**] [*File* ...]

Description
The **cb** command reads C programs from standard input or from specified files and writes them to standard output in a form that shows, through indentations and spacing, the structure of the code. When called without flags, the **cb** command does not split or join lines. Note that punctuation in preprocessor statements can cause indentation errors.

For best results, use this command on source code that is syntactically correct.

Flags

-j	Joins lines that are split. Ignored if **-l** flag is given.
-l *Length*	Splits lines that are longer than *Length* characters.
-s	Formats the source code according to the style of Kernighan and Ritchie in *The C Programming Language* (Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1978).

Example
To create a version of `pgm.c` called `pgm.pretty.c` that is easy to read, enter:

```
cb pgm.c > pgm.pretty.c
```

Files

/usr/ccs/bin/cb	Contains the **cb** command.
/usr/bin/cb	Symbolic link to the **cb** command.

Related Information
The **indent** command.

cd Command

Purpose
Changes the current directory.

Syntax
cd [*directory*]

or

cd [*directorya directoryb*]

Description

The **cd** command sets the current working directory of a process. The user must have execute (search) permission in the specified directory.

If a directory parameter is not specified, the **cd** command sets the current working directory to the login directory (**$HOME** in the **ksh** and **bsh** environments, or **$home** in the **csh** environment). If the specified directory name is a full path name, it becomes the current working directory. A full path name begins with a / (slash) indicating root directory, a **.** (dot) indicating current directory, or a **..** (dot-dot) indicating parent directory. If the directory name is not a full path name, the **cd** command searches for it relative to one of the paths specified by the **$CDPATH** shell variable (or **$cdpath csh** variable). This variable has the same syntax as, and similar semantics to, the **$PATH** shell variable (or **$path csh** variable).

Note: Running **/usr/bin/cd** from a shell does not change the shell's working directory. The shell's built in **cd** command must be used.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

1. To change the current working directory to the login (home) directory, type:
   ```
   cd
   ```
2. To change to an arbitrary directory, type:
   ```
   cd /usr/include
   ```
   This changes the current directory to `/usr/include`.
3. To go down one level of the directory tree, type:
   ```
   cd sys
   ```
   If the current directory is `/usr/include` and it contains a subdirectory named `sys`, then `/usr/include/sys` becomes the current directory.
4. To go up one level of the directory tree, type:
   ```
   cd ..
   ```
   The special file name, **..** (dot-dot), refers to the directory immediately above the current directory.
5. Specifying two directory parameters substitutes the string **directoryb** for the string **directorya** in the current working directory, then makes the new path the current directory. For example, if the current working directory is
   ```
   /home/directorya/sub1/sub2/sub3/sub4
   ```
   the command
   ```
   cd directorya directoryb
   ```
   will set the current working directory to
   ```
   /home/directoryb/sub1/sub2/sub3/sub4
   ```
   if that directory exists. Additionally, if the current working directory is:
   ```
   home/directorya/sub1/sub2/sub3/sub4
   ```

the command

```
cd directorya directoryb/test
```

will set the current working directory to

```
home/directoryb/test/sub1/sub2/sub3/sub4
```

if that directory exists. Likewise, if the current working directory is

```
/home/directoryb/test/sub1/sub2/sub3/sub4
```

the command

```
cd directoryb/test directorya
```

will set the current working directory to

```
home/directorya/sub1/sub2/sub3/sub4
```

if that directory exists.

Subdirectories must all have the same name.

Related Information

The **bsh** command, **csh** command, **ksh** command, **pwd** command.

The **chdir** subroutine.

Directories in *Operating system and device management* describes the structure and characteristics of directories in the file system.

Directories in *Operating system and device management*.

Shells in *Operating system and device management* describes shells, the different types, and how they affect the way commands are interpreted.

---

cdc Command

Purpose

Changes the comments in a SCCS delta.

Syntax

cdc -r*SID* [**-m** [*ModificationRequestList*]] [**-y** [*Comment*]] *File ...*

Description

The **cdc** command changes the Modification Requests (MRs) and comments for the specified SCCS delta (the *SID* variable) for each named Source Code Control System (SCCS) file. If you specify a directory name, the **cdc** command performs the requested actions on all SCCS files in that directory (that is, all files with names that have the **s.** prefix). If you specify a **-** (minus) in place of *File*, the **cdc** command reads standard input and interprets each line as the name of an SCCS file.

You can change the comments and MRs for an SID only if you made the SID or you own the file and the directory.

Flags

-m[*ModificationRequestList*]

Supplies a list of MR numbers for the **cdc** program to add or delete in the SID specified by the **-r** flag. You can only use this flag if the specified file has the **v** header flag set. A null MR list has no effect.

In the actual *ModificationRequestList* parameter, MRs are separated by blanks, tab characters, or both. To delete an MR, precede the MR number with an ! (exclamation point). If the MR you want to delete is currently in the list of MRs, it is changed into a comment line. The **cdc** command places a list of all deleted MRs in the comment section of the delta and precedes them with a comment line indicating that the MRs were deleted.

If you do not specify the **-m** flag, and the **v** header flag is set, MRs are read from standard input. If standard input is a workstation, the **cdc** command prompts you for the MRs. The first new-line character not preceded by a backslash ends the list on the command line. The **cdc** command continues to take input until it reads an end-of-line character or a blank line. MRs are always read before comments (see the **-y** flag).

If the **v** header flag has a value, the **cdc** command interprets the value as the name of a program that validates MR numbers. If the MR number validation program returns a nonzero exit value, the **cdc** command stops and does not change the MRs.

-r*SID*

Specifies the SCCS identification number of the delta for which the **cdc** command will change the comments or MRs.

-y[*Comment*]

Specifies comment text to replace an existing comment for the delta specified by the **-r** flag. The **cdc** command keeps the existing comments but precedes them by a comment line stating that they were changed. A null *Comment* value has no effect.

If you do not specify the **-y** flag, the **cdc** command reads comments from standard input until it reads an end-of-file character. If the standard input is a workstation, the **cdc** command prompts for the comments and also allows a blank line to end input. If the last character of a line is a \ (backslash), the **cdc** command ignores it and continues to read standard input.

> **Note:** If the **cdc** command reads standard input for file names (that is, when you specify a file name of **-**), you must use the **-y** and **-m** flags.

Example

To change the comment for SID 1.3 of SCCS file `s.text.c` to `"new comment"`, enter:

```
cdc -r1.3 -y"new comment" s.test.c
```

Files

/usr/bin/cdc	Contains the path to SCCS **cdc** command**.**

Related Information

The **admin** command, **delta** command, **get** command, **prs** command, **sccshelp** command.

The **sccsfile** file format.

Source Code Control System (SCCS) Overview in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

List of SCCS Commands in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

---

cdcheck Command

Purpose

Asks **cdromd** daemon information about a device.

Syntax

cdcheck { **-a** | **-m** | **-u** | **-e** } [**-q**] [**-h** | **-?**] *DeviceName*

Description

The **cdcheck** command sends an appropriate command to the **cdromd** daemon to get information on a media or a device depending on the flag used.

The **cdcheck** command returns a zero (True) exit value and prints a message on **stdout** if the specified condition is true. Otherwise, the **cdcheck** command returns a nonzero (False) exit value and prints an error message on **stderr**.

To check if a device is managed by **cdromd** daemon, use the **cdcheck** command with the **-a** flag. If the **cdromd** daemon is running and the specified device is in its device list, the **cdcheck -a** command will return with a zero (True) exit value after printing the following message on **stdout**:

```
cd<x> is managed by cdromd.
```

Note: An exit value of zero (True) with the **-a** flag means that a media will be automatically mounted when it is inserted. It does not mean that a media is currently mounted.

To check if a media is present and was mounted by **cdromd** daemon, use the **cdcheck** command with the **-m** flag. When a media is inserted in a drive, it can take several seconds or tens of seconds before it become ready and mounted. The **cdcheck -m** command waits until the end of the mount operation by the **cdromd** daemon. If this operation is successfull, the **cdcheck -m** command returns with a zero (True) exit value after printing the mount point on **stdout**.

Note: If the media is damaged and can't be mounted by the **cdromd** daemon, the **cdcheck -m** command returns a nonzero (False) exit value and prints an error message on **stderr**.

To check if a media is present but was unmounted by the **cdumount** command, use the **cdcheck** command with the **-u** flag. If the **cdromd** daemon is running and the specified device is in in unmounted state, the **cdcheck -u** command will return with a zero (True) exit value after printing the following message on **stdout**:

```
cd<x> is not mounted.
```

To check that there is no media present in the specified device, use the **cdcheck** command with the **-e** flag. If the **cdromd** daemon is running and there is no media present in the drive, the **cdcheck -e** command will return with a zero (True) exit value after printing the following message on **stdout**:

```
No media present in cd<x>.
```

When using **cdcheck** in shell scripts, the **-q** flag can be added to the **cdcheck** command so that no messages are printed on **stdout** and **stderr**. The only exception is the **cdcheck** command with the **-m** flag, which always prints the mount point on **stdout** so that the shell script can get this mount point.

Flags

-a	Checks if a device is managed by **cdromd**.
-e	Checks if a media has been ejected from a device.
-h or **-?**	Displays the command usage message.
-m	Checks if a media is mounted on a device.
-q	Specifies silent mode: Doesn't print any information or error message. **Note:** If **-q** is used with the **-m** flag, the mount point will be printed to **stdout**.
-u	Checks if a media is not mounted on a device.
DeviceName	Specifies the name of the device.

Exit Status

This command returns the following exit values:

0 answer = yes.

>0 answer = no or error.

Examples

1. To ask cdromd if cd0 is managed enter:
   ```
   cdcheck -a cd0
   ```
2. To ask **cdromd** if a media is mounted on **cd1** without any printed error messages, enter:
   ```
   cdcheck -m -q cd1
   ```
3. To ask **cdromd** if a media is not mounted on **cd1** enter:
   ```
   cdcheck -u cd1
   ```
4. To ask **cdromd** if a media is not present on **cd0** enter:
   ```
   cdcheck -e cd0
   ```
5. Shell script example:
   ```
   DEVICE=$1

   if [ cdcheck -a -q "$DEVICE" ]; then
       AUTO_MOUNT="ON"
   else
       AUTO_MOUNT="OFF"
   fi

   # Other initializations
   # ...

   if [ "$AUTO_MOUNT" = "ON" ]; then
       MOUNT_POINT=`cdcheck -m -q $DEVICE`
   else
       MOUNT_POINT="/tmp/MyProg_$$"
       mount -rv cdrfs $DEVICE $MOUNT_POINT
   fi
   if [ $? -ne 0 ]; then
       echo "mount $DEVICE failed"
       exit 1
   fi

   # Now extract data from $MOUNT_POINT...
   ```

```
# ...


# End of processing. Umount the media
if [ "$AUTO_MOUNT" = "ON" ]; then
    cdeject -q $DEVICE
else
    unmount $DEVICE
fi
if [ $? -ne 0 ]; then
    echo "unmount $DEVICE failed"
    exit 1
fi
```

Related Information

The **cdeject** command, **cdmount** command, **cdromd** daemon, **cdumount** command, **cdutil** command.

The **cdromd.conf** file format.

---

cdeject Command

Purpose

Ejects a media from a CD drive managed by the **cdromd** daemon.

Syntax

cdeject [**-q**] [**-h** | **-?**] *DeviceName*

Description

The **cdeject** command sends an appropriate command to the **cdromd** daemon which unmounts (if necessary) the file system corresponding to the specified device and ejects the media from the drive specified by *DeviceName*.

Flags

-h or **-?**	Displays the command usage message.
-q	Specifies silent mode: Doesn't print any information or error message.
DeviceName	Specifies the name of the device.

Exit Status

This command returns the following exit values:

0 No error.

>0 An error occurred.

Examples

1. To eject a media from **cd0** enter:
   ```
   cdeject cd0
   ```
2. To eject a media from **cd1** without any printed error messages,enter:
   ```
   cdeject -q cd1
   ```

Related Information

The **cdcheck** command, **cdmount** command, **cdromd** daemon, **cdumount** command, **cdutil** command.

The **cdromd.conf** file format.

---

cdmount Command

Purpose

Makes a file system available for use on a device managed by **cdromd**.

Syntax

cdmount [**-q**] [**-h** | **-?**] *DeviceName*

Description

The **cdmount** command sends an appropriate command to the **cdromd** daemon which mounts the file system on the device specified by *DeviceName* if it is not already mounted. This command can be used to mount a file system that was previously unmounted by the **cdumount** command.

The mount point used is either the one found in **/etc/cdromd.conf** file for the specified *DeviceName* or the default one (**/cdrom/cd0** for cd0, **/cdrom/cd1** for cd1, etc...).

The file system type and options used (**-o** and **-V** flag for **mount** command) are those found in **/etc/cdromd.conf** file or the default ones: ″**-Vcdrfs -oro**″ for a CD-ROM and ″**-Vudfs -oro**″ or ″**-Vcdrfs -oro**″ for DVD-ROM.

Flags

-h or **-?**	Displays the command usage message.
-q	Specifies silent mode: Doesn't print any information or error message.
DeviceName	Specifies the name of the device.

Exit Status

This command returns the following exit values:

0 No error.

>0 An error occurred.

Examples

1. To mount a file system on **cd0** enter:
   ```
   cdmount cd0
   ```
2. To mount a file system on **cd1** without any printed error messages, enter:
   ```
   cdmount -q cd1
   ```

Related Information

The **cdcheck** command, **cdeject** command, **cdromd** daemon, **cdumount** command, **cdutil** command, **mount** command.

The **cdromd.conf** file format.

cdromd Command

Purpose

Automatically mounts a CD-ROM or DVD-ROM when it is inserted in a device, and provides the server function for the **cdutil**, **cdcheck**, **cdmount**, **cdumount**, and **cdeject** commands.

Syntax

cdromd [**-d**]

Note: Use System Resource Controller (SRC) commands to control the **cdromd** daemon from the command line. To have the **cdromd** daemon enabled on each system startup, add the following line to **/etc/inittab**:

```
cdromd:23456789:wait:/usr/bin/startsrc -s cdromd
```

Description

The **cdromd** daemon finds the device list it has to manage and their respective mount points in **/etc/cdromd.conf** file. If this file does not exist or is empty, **cdromd** manages all the CD-ROM and DVD-ROM devices available on the system, and the mount points are **/cdrom/cd0** for **cd0**, **/cdrom/cd1** for **cd1**, etc.

After its init phase **cdromd** periodically checks if a media is present in one of the managed drives (for devices that are not already mounted) and mounts it if there is a media.

cdromd also periodically checks its socket for requests comming from **cdutil**, **cdcheck**, **cdmount**, **cdumount** or **cdeject** commands.

The **cdromd** daemon should be controlled using the System Resource Controller (SRC). Entering **cdromd** at the command line is not recommended.

The cdromd daemon sends its error messages to the **syslogd** daemon.

The **cdromd** daemon can interfere with scripts, applications, or instructions that attempt to mount the CD or DVD device without first checking to see if the device is already enabled. A resource or device busy error will occur in such a condition. Use the **cdumount** or **cdeject** command to unmount the device so that you can mount the device as specified in the program or instructions. Alternatively, use the **cdcheck -m** or **mount** command to determine the current mount point of the device.

Manipulating the cdromd daemon with the System Resource Controller:

The **cdromd** daemon is a subsystem controlled by the System Resource Controller (SRC). Its subsystem name is **cdromd**. The **cdromd** daemon can be manipulated by the following SRC commands:

stopsrc
: Stops a subsystem, group of subsystems, or a subserver.

startsrc
: Starts a subsystem, a group of subsystems, or a subserver.

refresh
: Requests a refresh of a subsystem or group of subsystems.

traceson
: Turns on tracing of a subsystem, group of subsystems, or a subserver.

tracesoff
: Turns off tracing of a subsystem, group of subsystems, or a subserver.

lssrc Gets the status of a subsystem, group of subsystems, or a subserver.

In addition, the **cdromd** daemon can be controlled by issuing signals using the **kill** command. Sending a **SIGHUP** signal to cdromd is equivalent to the "`refresh -s cdromd`" command, and sending a **SIGTERM** signal to **cdromd** is equivalent to the "`stopsrc -s cdromd`" command.

Flags

-d Sends debugging messages to **syslogd** daemon.

Exit Status

This daemon returns the following exit values:

0 The **cdromd** daemon was stopped by SRC or **SIGTERM** signal.

>0 An error occurred.

Examples

1. To stop the cdromd daemon normally, enter the following:

   ```
   stopsrc -s cdromd
   ```

   This command stops the daemon. The **-s** flag indicates that the specified subsystem is to be stopped.
2. To start the **cdromd** daemon, enter the following:

   ```
   startsrc -s cdromd
   ```

   This command starts the daemon. This command is in the **/etc/inittab** file and can be used on the command line. The -s flag indicates that the specified subsystem is to be started.
3. To get a short status report from the **cdromd** daemon, enter the following:

   ```
   lssrc -s cdromd
   ```

   This command returns the name of the daemon, the process ID of the daemon, and the state of the daemon (active or inactive).
4. To tell **cdromd** daemon its configuration file has changed, enter the following:

   ```
   refresh -s cdromd
   ```

   This command tells the **cdromd** daemon to read its configuration file again.

Files

/etc/cdromd.conf Describes managed devices and supported file systems.

Related Information

cdcheck command, **cdeject** command, **cdmount** command, **cdumount** command, **cdutil** command, **syslogd** daemon.

cdromd.conf file format

---

cdumount Command

Purpose

Unmounts a previously mounted file system on a device managed by **cdromd**.

Syntax

cdumount [**-q**] [**-h** | **-?**] *DeviceName*

Description

The **cdumount** command sends an appropriate command to the **cdromd** daemon which tries to unmount the file system on the device specified by *DeviceName*.

The **cdumount** command doesn't eject the media.

Flags

-h or **-?**	Displays the command usage message.
-q	Specifies silent mode: Doesn't print any information or error messages.
DeviceName	Specifies the name of the device.

Exit Status

This command returns the following exit values:

0 No error.

>0 An error occurred.

Examples

1. To unmount a file system on **cd0** enter:
   ```
   cdumount cd0
   ```
2. To unmount a file system on **cd1** without any printed error messages, enter:
   ```
   cdumount -q cd1
   ```

Related Information

The **cdcheck** command, **cdeject** command, **cdmount** command, **cdromd** daemon, **cdutil** command.

The **cdromd.conf** file format.

---

cdutil Command

Purpose

Tells the **cdromd** daemon to suspend or resume management of a device.

Syntax

cdutil { **-l** | **-r** | **-s** [**-k**] } [**-q**] [**-h** | **-?**] *DeviceName*

Description

The **cdutil** command sends an appropriate command to the **cdromd** daemon which suspends (**-s** flag) or resumes (**-r** flag) the management of the device specified by *DeviceName*.

A device managed by **cdromd** must be set in suspend state if it needs to be unconfigured (for example for a hotswap of the parent adapter).

The resume flag (**-r**) asks **cdromd** to restart polling the device.

Flags

-h or **-?**	Displays the command usage message.
-k	Do not eject the media when suspending a device.
-l	Load the media if one is present in the drive.
-q	Specifies silent mode: Doesn't print any information or error messages.
-r	Resumes device management by **cdromd**.
-s	Suspends device management by **cdromd**.
DeviceName	Specifies the name of the device.

Exit Status

This command returns the following exit values:

0 No error

>0 An error occurred

Examples

1. To suspend management of **cd0** by **cdromd**, type:
   ```
   cdutil -s cd0
   ```
2. To suspend management of **cd0** by **cdromd** without ejecting the media, type:
   ```
   cdutil -s -k cd0
   ```
3. To resume management of **cd1** by **cdromd** without any printed error messages, type:
   ```
   cdutil -r -q cd1
   ```

Related Information

The **cdcheck** command, **cdeject** command, **cdmount** command, **cdromd** daemon, **cdumount** command.

The **cdromd.conf** file format.

---

certadd Command

Purpose

certadd stores a certificate into the local LDAP repository.

Syntax

certadd [**-c**|**-r**] [**-p** *privatekeystore*] [**-f** *file*] **-l** *label tag* [*username*]

Description

The **certadd** command stores a user-supplied certificate in the local LDAP repository.

If the **-c** (create only) option is used, it will return an error if the username and tag pair already exists as a named certificate. Otherwise, the existing certificate shall be replaced by the new certificate. If the **-r** (replace only) option is used, an error is returned if the username and tag pair does not already exist as a named certificate. These two options are mutually exclusive. The default behavior is to create the entry if it does not exist and to replace the existing certificate if it exists.

If the **-f** option is not given, the certificate shall be read from stdin. The certificate is in DER format. The **certadd** command is limited to `root` users, or users with the appropriate administrative roles, when the username parameter is other than the current user.

The **-l** option must always be specified. The label is a variable length text string that will be used to map a key in the keystore to the certificate which contains the matching public key. Make sure this label is the same as the one specified when the **certcreate** command is invoked.

If the **-p** option is not given, the default will be **file:/var/pki/security/keys/<***username***>**. If no protocol is specified, **file:** is assumed. Currently only URIs of type **file:** are supported. It is the responsibility of the invoker of this command to ensure that the private keystore contains the private key matching the public key in the certificate. If the certificate to be added is created using the **certcreate** command, then the private key is already in the private keystore. Alternatively, if the certificate is externally created, the user can later add the private key associated with the public key to the private keystore using the **keyadd** command.

The *tag* parameter is a variable length text string from the same character set as user names which is used to uniquely identify the certificate amongst all of the certificates owned by username. The *tag* `ALL` shall be reserved for the **certlist** command so that all certificates owned by a user may be viewed, therefore can not be used with the **certadd** command. It shall be also an error to replace a certificate named by the **auth_cert** attribute for a user. When an existing certificate is replaced with another one, the keys corresponding to the replaced certificate remain in the keystore until deleted by the user. These keys could be removed from the keystore using key management commands. Similarly, the keys for the new certificate could also be added to the keystore again using the key management commands. Only a certificate that is not revoked can be added, unless the system policy specifies otherwise.

The system revocation check policy is specified in the policy file, **/usr/lib/security/ pki/policy.cfg** under the stanza **crl**. When the **check** attribute is set to `yes`, the certificates are checked against a CRL. The certificate revocation list will be obtained using the Certificate Revocation Distribution Point information from the certificate and from the **/usr/lib/security/pki/ca.cfg file**. This file has an entry called **crl**, which one can use to specify the method of CRL retrieval. **ldap:**, **http:** and **file:** retrieval methods are supported. If more than one URI is specified, they must be delimited with a space. The certificate will not be added if the certificate revocation list could not be retrieved.

Flags

-c	Adds a new certificate.
-r	Replaces an existing certificate.
-l *label*	Specifies a label for the private key that matches the public key in certificate.
-p *privatekeystore*	Specifies the location of the private keystore.
-f *file*	Specifies a file that contains the DER-encoded certificate.

Exit Status

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

Security

This is a privileged (set-UID root) command.

`Root` and invokers belonging to group security can add certificates for anybody. A non-privileged user can only add certificates for themself.

Audit
This command records the following event information:

CERT_Add <*username*>

Examples
To add a certificate stored in **cert.der** to the local LDAP repository and associate it with user Bob, enter:
```
$ certadd -c -f cert.der -l signcert cert1 bob
```

or,
```
$ certadd -c -l signcert cert1 bob < cert.der
```

This will read the DER encoded certificate from file **cert.der** and assign **signcert** as the label and **cert1** as the tag and store it in LDAP as Bob's certificate. The default private keystore location will be **/var/pki/security/keys/bob**.

To replace Bob's **cert1** certificate with another certificate enter:
```
$ certadd -r -f newcert1.der -l newsigncert cert1 bob
```

Files
/usr/lib/security/pki/acct.cfg

/usr/lib/security/pki/ca.cfg

/usr/lib/security/pki/policy.cfg

Related Information
The **certcreate**, **certdelete**, **certget**, **certlink**, **certlist**, **certrevoke**, **certverify**, **keyadd**, **keydelete**, **keylist**, **keypasswd**, and **mksecpki** commands.

---

certcreate Command

Purpose
certcreate requests a new certificate for the specified user.

Syntax
certcreate [**-S** *servicename*] [**-s** *startdate*] [**-e** *enddate*] { **-f** *file* | [**-b** | **-t**] } [**-p** *privatekeystore*] **-l** *label* [**-a** *subject_alt_name*] *subject_distinguished_name* [*user-name*]

Description
The **certcreate** command invokes the end-entity services and libraries and requests that a new certificate be created with the identifying information contained on the command line. Which service to use is specified by the **-S** option. Available services are defined in **/usr/lib/security/pki/ca.cfg**. Certificate requests without the **-S** option are created using the local service. It is an error to specify a servicename which does not have an entry in the **/usr/lib/security/pki/ca.cfg** file. The service entry in the **ca.cfg** file specifies which CA to send the request.

If the **-s** option is not given, the current day's date shall be used. If the **-e** option is not given, the validity value from the **policy.cfg** file will be used. If this value does not exist, then one year from the starting date shall be used as the validity period. Both `startdate` and `enddate` shall have the same format as the `expires` attribute used by the **chuser** command. The format is 10-character string in the `MMDDhhmmyy` form,

where MM refers to month, DD refers to day, hh refers to hour, mm refers to minute, and yy refers to last 2 digits of the years 1939 through 2038. All characters are numeric.

If the **-f** option is given, the new certificate shall be DER encoded and stored in the named file in a binary format. Otherwise, it shall be DER encoded and output to **stdout**, either in binary or in hexadecimal format. If **-b** option is given then the output will be displayed to **stdout** in binary, otherwise it will be hexadecimal. If neither **-b** nor **-t** is given, a binary format will be used.

The corresponding private key shall be stored in a private keystore or device, as required by the underlying commands or libraries. If **-p** option is given, the private key will be stored in private keystore specified. If **-p** option is not given the default will be **/var/pki/ security/keys/**<*username*>.

The **-l** option must be specified. The label is a variable length text string that will be used as an alias for the private key in the keystore.

The value of *subject_alt_name* will be an Internet electronic mail address (RFC2459 defines this to be a rfc822Name). This value is optional. If no value is provided, the certificate will not have an rfc822Name subject alternative name extension. *Subject_distinguished_name* shall be restricted to the valid set of values for PKI certificates. This is defined to be an X.501 type Name by RFC2459.

The **certcreate** command issues one or more prompts and request a password in order to generate the certificate and store it in the user's private keystore. If the user has an existing keystore, the user will be prompted once for the password. If the keystore does not exist, then it will be created and the user will be asked to re-enter the password again for confirmation. The command will fail if it is unable to open **/dev/tty** for the current process.

Flags

-S *servicename*	Specifies which service module to use.
-s *startdate*	Specifies the date on which the certificate will become valid.
-e *enddate*	Specifies the date on which the certificate will become invalid.
-f *file*	Specifies the file that certificate will be stored.
-p *privatekeystore*	Specifies the location of the private keystore.
-l *label*	Specifies the label of the private key in the keystore.
-a *subject_alt_name*	Specifies the subject alternative name of the certificate owner.
-b	Specifies the format of the certificate data to be binary.
-t	Specifies the format of the certificate data to be hexadecimal.

Exit Status

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

Security

This is a **setuid** command.

Root and invokers belonging to group security can create certificates for anyone. A non-privileged user can only create certificates for himself with the following rules while specifying a private keystore location:

- The invoker can specify the default private keystore: **/var/pki/security/keys/**<*user-name*>

- The invoker can specify a private keystore that they have access to write.

A non-privileged user can not request a certificate for others.

Audit
This command records the following event information:

CERT_Create <*username*>

Examples
```
$ certcreate -S local -s 0831112702 -e 1231235902 -f cert.der -p file:/home/bob/bob.priv -l signcert bob@ibm.com ou=finance,cn=Bob%20James bob
```

In the above example, the certificate will be valid from August 31, 2002 11:27 AM until December 31, 2002, 11:59 PM. The certificate will be placed in file **cert.der** and the private key will be stored in **bob.priv** with an alias **signcert**.

The following example uses the defaults for the start date, end date, and the private keystore.
```
$ certcreate -l signcert bob@ibm.com ou=finance,cn=Bob James > cert.der
```

Files
/usr/lib/security/pki/ca.cfg

/usr/lib/security/pki/policy.cfg

Related Information
The **certadd**, **certdelete**, **certget**, **certlink**, **certlist**, **certrevoke**, **certverify**, **keyadd**, **keydelete**, **keylist**, **keypasswd**, and **mksecpki** commands.

---

certdelete Command

Purpose
certdelete removes a certificate from the list of certificates associated with a user account and deletes the certificate from the local LDAP repository.

Syntax
certdelete tag [*username*]

Description
The **certdelete** command removes certificates associated with a user from the local LDAP repository. A deleted certificate could be added again using the **certadd** command. Note that the **certdelete** operation does not affect the certificates in CA's LDAP store where they are published.

The **tag** parameter uniquely identifies the certificate in the list of certificates owned by a user. It shall be an error to remove the certificate named by the **auth_cert** attribute for a user. Only a privileged (`root`) user, or a user belonging to group security may specify a user name other than their own.

If invoked without the username parameter, the **certdelete** command uses the name of the current user.

Specifying `ALL` as the value of tag will cause all of the certificates owned by a user to be removed. The command terminates on the first delete error it encounters while processing an `ALL` request. This leaves the rest of the certificates owned by the user undeleted. If the error is due to some temporary condition

(such as local LDAP repository is inaccessible), the next **certdelete** will delete the remaining certificates. The user might query about the certificates that did not get deleted by using **certlist** command with a tag value of `ALL`.

Exit Status

0	Successful completion.
>0	An error occured.

Security

This is a privileged (set-UID root) command.

`Root` and invoker belonging to group security can delete certificates for anybody. A non-privileged user can only delete certificates for himself/herself.

Audit

This command records the following event information:

CERT_Create <*username*>

Examples

1. To remove a certificate with a tag value **signcert** belonging to Bob, enter:
   ```
   $ certdelete signcert bob
   ```
2. To remove all the certificates from the local LDAP repository belonging to the current user, enter:
   ```
   $ certdelete ALL
   ```

Files

/usr/lib/security/pki/acct.cfg

Related Information

The **certadd**, **certcreate**, **certget**, **certlink**, **certlist**, **certrevoke**, **certverify**, **keyadd**, **keydelete**, **keylist**, **keypasswd**, and **mksecpki** commands.

---

certget Command

Purpose

certget retrieves a single certificate from local LDAP repository.

Syntax

certget {**-f** *file* | [**-b** | **-t**]}*tag* [*username*]

Description

The **certget** command retrieves a single certificate from the local LDAP repository. This command retrieves a single certificate at a time. If the invoker wishes to retrieve all the certificates for a user, the **certlist** command may be used to first to obtain a list of the certificates and then perform the **certget** operation on the certificate list.

If the **-f** option is used, the certificate shall be written in binary format to the named file. Otherwise the certificate is output to **stdout** either in binary or hexadecimal. If the **-b** option is given, binary output is used (default). If the **-t** option is given, hexadecimal output is used. Certificates are output in DER format.

The **tag** parameter uniquely selects one of the user's certificates. The **username** parameter specifies which AIX user is to be queried. If invoked without the **username** parameter, the **certdelete** command uses the name of the current user.

Flags

-f	Specifies the file that the DER encoded certificate will be stored.
-b	Specifies the format of the certificate data to be binary.
-t	Specifies the format of the certificate data to be hexadecimal.

Exit Status

0	If successful.
EINVAL	If the command is ill-formed or the arguments are invalid.
ENOENT	If a) the user doesn't exist, b) the tag does not exist c) the file does not exist.
EIO	If unable to create/modify LDAP entry.
ENOCONNECT	If the service is not available.
errno	If system error.

Security

This command can be executed by anyone to retrieve a certificate belonging to a user from the local repository.

Audit

This command records the following event information:

CERT_Get <*username*>

Examples

1. To retrieve Bob's certificate tagged as **signcert** and store in **cert.der**, enter:
   ```
   $ certget -f cert.der signcert bob
   ```
2. To store Bob's certificate **signcert** in hexadecimal in **cert.der**, enter:
   ```
   $ certget -t signcert > cert.der
   ```

Files

/usr/lib/security/pki/acct.cfg

Related Information

The **certadd**, **certcreate**, **certdelete**, **certlink**, **certlist**, **certrevoke**, **certverify**, **keyadd**, **keydelete**, **keylist**, **keypasswd**, and **mksecpki** commands.

---

certlink Command

Purpose

certlink links a certificate in a remote repository to a user account.

Syntax

certlink [**-c**|**-r**] [**-p** *privatekeystore*] **-l** *label* **-o** *option tag* [*username*]

Description

The **certlink** command links a certificate in a remote repository to a user account. **certlink** is very similar to **certadd** except that the user provides a link to the certificate rather than providing the certificate itself.

If the **-c** (create only) option is given, it is an error if the {username, tag} pair already exists as a named certificate. Otherwise, an existing certificate shall be replaced by the new certificate. If the **-r** (replace only) option is given, it is an error if the {username, tag} pair does not already exist as a named certificate. These two options are mutually exclusive. The default behavior is to create the entry if it does not exist and to replace the existing certificate if it exists.

The **-l** option must be specified. The label is a variable length text string that will be used to map a key in the keystore to the certificate which contains the matching public key.

If the **-p** option is not given, the default will be **/var/pki/security/keys/<***username***>**. It is the responsibility of the invoker of this command to add the private key associated with the public key by using the **keyadd** command. Refer to the **certadd** command for more details on the use of the **-l** and **-p** flags. This information also applies to the **certlink** command.

The **-o** option is the URI where the certificate is stored. Currently only LDAP URIs are supported. The URI of the repository must be given in the format as specified in RFC 2255.

The **tag** parameter is a variable length text string from the same character set as user names which is used to uniquely identify the certificate among all of the certificates owned by **username**. The **ALL** tag shall be reserved for the **certlist** command so that all certificates owned by a user may be viewed. An error is also returned if a certificate named by the **auth_cert** attribute for a user is replaced.

When an existing certificate is replaced with another one, the keys corresponding to the replaced certificate remain in the keystore until deleted by the user. These keys can be removed from the keystore using key management commands. Similarly, the private key matching to a certificate can also be added to the keystore using the key management commands.

Only a certificate that is not revoked can be added unless the system policy specifies otherwise. The system revocation check policy is specified in the policy file **/usr/lib/security/ pki/policy.cfg**. The certificate revocation list will be obtained using the Certificate Revocation Distribution Point information in the certificate. If one is not given, the certificate distribution point information will be retrieved from the **/usr/lib/security/ pki/ca.cfg** file. The certificate will not be added, if the certificate revocation list could not be retrieved.

Flags

-c	Links a new certificate.
-r	Replaces an existing certificate.
-p	Specifies the location of the private keystore.
-l *label*	Specifies a label for the private key corresponding to the public key in certificate.
-o *option*	Specifies the URL where the certificate to be linked stored.

Exit Status

0	If successful.
>0	An error occured.

Security

This is a privileged (set-UID root) command.

Root and *invokers* belonging to group security can add certificates for anybody. A non-privileged user can only add certificates for themself.

Examples

To link a certificate stored in an external certificate repository and associate it with user Bob, enter:

```
$ certlink -c -l signcert -p /home/bob/keystore.p12 -o ldap://
cert.austin.ibm.com/o=ibm,ou=Finance,c=us?usercertificate??(
cn=Bob James)?X-serial=1A:EF:54 cert1 bob
```

Files

/usr/lib/security/pki/ca.cfg

/usr/lib/security/pki/policy.cfg

Related Information

The **certadd**, **certcreate**, **certdelete**, **certget**, **certlist**, **certrevoke**, **certverify**, **keyadd**, **keydelete**, **keylist**, **keypasswd**, and **mksecpki** commands.

---

certlist Command

Purpose

certlist lists the contents of one or more certificates.

Syntax

certlist [**-c**] [**-a** attr [attr....]]*tag* [*username*]

Description

The **certlist** command lists the contents of one or more certificates. Using the **-c** option causes the output to be formatted as colon-separated data with the attribute names associated with each field on the previous line as follows:

```
# name: attribute1: attribute2: ...
User: value1: value2: ...
```

The **-f** option causes the output to be formatted in stanza file format with the username attribute given as the stanza name. Each `attribute=value` pair is listed on a separate line:

```
user:
     attribute1=value
     attribute2=value
     attribute3=value
```

When neither of these command line options are selected, the attributes are output as `attribute=value` pairs.

The **-a** option selects a list of one or more certificate attributes to output. In addition to the attributes supported by the load module, several pseudo-attributes shall also be provided for each certificate.

Those attributes are:

auth_user User's authentication certificate.

distinguished_name	User's subject distinguished name in the certificate.
alternate_name	User's subject alternate name in the certificate.
validafter	The date the user's certificate becomes valid.
validuntil	The date the user's certificate becomes invalid.
tag	The name that uniquely identifies this certificate.
issuer	The distinguished name of the certificate issuer.
label	The label that identifies this certificate in the private keystore.
keystore	The location of the private keystore for the private key of the certificate.
serialnumber	The serial number of the certificate.
verified	`true` indicates that the user poved that he is in possession of the private key.

Flags

-c	Displays the output in colon-separated records.
-f	Displays the output in stanzas.
-a *attr*	Selects one or more attributes to be displayed.

The **tag** parameter selects which of the user's certificates is to be output. The reserved value `ALL` indicates that all certificates for the user are to be listed.

The **username** parameter specifies the name of the AIX user to be queried. If invoked without the **username** parameter, the **certdelete** command uses the name of the current user.

Exit Status

0	If successful.
EINVAL	If the command is ill-formed or the arguments are invalid.
ENOENT	If a) the user doesn't exist, b) the tag does not exist c) the file does not exist.
EACCES	If the attribute cannot be listed, for example, if the invoker does not have **read_access** to the user data-base.
EPERM	If the user identification and authentication fails.
errno	If system error.

Security

This command can be executed by any user in order to list the attributes of a certificate. Certificates listed may be owned by another user.

Audit

This command records the following event information:

CERT_List <*username*>

Examples

```
$ certlist -f -a verified keystore label signcert bob
bob:
      verified=false
      keystore=file:/var/pki/security/keys/bob
      label=signcert
```

```
$ certlist -c -a validafter validbefore issuer signcert bob
#name:validafter:validuntil:issuer
bob:1018091201:1018091301:c=US,o=xyz
$ certlist -f ALL bob
bob:
      auth_cert=logincert
      distinguished_name=c=US,o=xyz,cn=bob
      alternate_name=bob@xyz.com
      validafter=0921154701
      validuntil=0921154801
      issuer=c=US,o=xyz
      tag=logincert
      verified=true
      label=loginkey
      keystore=file:/var/pki/security/keys/bob
      serialnumber=03
bob:
      auth_cert=logincert
      distinguished_name=c=US,o=xyz,cn=bob
      alternate_name=bob@ibm.com
      validafter=1018091201
      validuntil=1018091301
      issuer=c=US,o=xyz
      tag=signcert
      verified=false
      label=signkey
      keystore=file:/var/pki/security/keys/bob
      serialnumber=02
```

Files

/usr/lib/security/pki/acct.cfg

/usr/lib/security/pki/policy.cfg

Related Information

The **certadd**, **certcreate**, **certdelete**, **certget**, **certlink**, **certrevoke**, **certverify**, **keyadd**, **keydelete**, **keylist**, **keypasswd**, and **mksecpki** commands.

---

certrevoke Command

Purpose

certrevoke revokes a user certificate.

Syntax

certrevoke [**-S** *servicename*] { **-f** *file* **-l** *label* [**-p** *privatekeystore*] | tag [*user-name*]}

Description

The **certrevoke** command is used to revoke certificates issued by a certificate authority which is part of the system's domain. The **-S** option specifies which service to use while revoking a certificate. Available services are defined in **/usr/lib/security/pki/ca.cfg**. Certificate requests without the **-S** option are created using the local service. An error is returned if you specify a servicename which does not have an entry in the **/usr/lib/security/pki/ ca.cfg** file.

If the **-f** option is selected, the certificate shall be read from the named file, or **stdin** if the name is ″**-**″. Certificates must be in DER format. Whenever the user specifies the **-f** option, the label of the private key matching the public key must also be specified. If the user does not provide the location of the private keystore, the default location will be used.

If the **-f** option is not specified, the invoker must provide the tag value and optional username for the certificate to be revoked. If invoked without the username parameter, the **certrevoke** command will use the name of the current user.

The **-l** option will be used to retrieve the private key matching the public key in the certificate that is to be revoked. The **certrevoke** command will fail if the user is unable to demonstrate the ownership of the private key matching the public key that is to be revoked. The **certrevoke** command will ask the user a password before actually performing the certificate revocation. The command may fail if it is unable to open **/dev/tty** for the current process.

Flags

-S *servicename*	Specifies which service module to use.
-f *file*	Specifies that the certificate to be revoked will be read form file.
-l *label*	Specifies the label associated with the private key of the certificate to be revoked.
-p *privatekeystore*	Specifies the location of the private keystore.

Exit Status

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

Security

This is a **setuid** command.

`Root` and invokers belonging to group security can revoke anybody's certificate. `Root` will revoke the the certificate using the revocation passphrase. Revocation passphrase is specified in the **/usr/lib/security/pki/acct.cfg** file.

A non-privileged user can only revoke certificates that they own. They have to demonstrate that they own the private key matching to the public key in the certificate to be revoked.

Audit

This command records the following event information:

CERT_Revoke <*username*>

Examples

To revoke the certificate `signcert` owned by Bob, enter:

```
$ certrevoke signcert bob
```

To revoke a certificate in file **cert.der**, enter:

```
$ certrevoke cert.der
```

Files

/usr/lib/security/pki/ca.cfg

Related Information

The **certadd**, **certcreate**, **certdelete**, **certget**, **certlink**, **certlist**, **certverify**, **keyadd**, **keydelete**, **keylist**, **keypasswd**, and **mksecpki** commands.

certverify Command

Purpose
certverify verifies that the invoker is in possession of the private key for the specified certificate.

Syntax
certverify [**-S** *servicename*] tag [*user-name*]}

Description
The **certverify** command verifies that the user is in possession of the private key for the specified certificate. Once the system verifies that the user is in possession of the private key, a signature is created for this certificate and associated with the certificate. A certificate that has not gone through this verification process is considered untrustworthy by AIX.

The **-S** option specifies which end-entity services and libraries to use while verifying the certificate. Available services are defined in **/usr/lib/security/pki/ca.cfg**. When invoked without **-S** flag, **certverify** will use the default service, **local**. It is an error to specify a service name which does not have an entry in the **/usr/lib/security/pki/ca.cfg** file. The `tag` parameter uniquely selects one of the user's certificates. The `username` parameter specifies which AIX user is to be queried. The **certverify** command will issue a password prompt and request the user to enter the password of the keystore. The command may fail if it is unable to open **/dev/tty** for the current process.

Flags

-S *servicename* Specifies which service module to use.

Exit Status

0 Successful completion.

>0 An error occured.

Security
This is a **setuid** command.

A user must prove that he has the possession of the private key matching the certificate he owns by knowing the password of the private keystore and the label that identifies the private key in the keystore.

`Root` and invokers belonging to group security are allowed to perform the verification operation, however, they can only successfully complete this operation if they have the knowledge of the label and the password to the keystore.

A non-privileged user is allowed to verify the possession of the private key only for the certificates they own.

Audit
This command records the following event information:

CERT_Verify <*username*>

Examples

To verify Bob's `cert1` certificate, enter:

```
$ certverify cert1 bob
```

Files

/usr/lib/security/pki/acct.cfg

Related Information

The **certadd**, **certcreate**, **certdelete**, **certget**, **certlink**, **certlist**, **certrevoke**, **keyadd**, **keydelete**, **keylist**, **keypasswd**, and **mksecpki** commands.

---

cfgif Method

Purpose

Configures or activates one or all network interface (IF) instance(s) defined in the system configuration database.

Syntax

cfgif [**-l** *InterfaceInstance*]

Description

The **cfgif** method configures or activates one or all IF instance(s) of TCP/IP defined in the system configuration database. The **cfgif** method performs the following steps:

1. Retrieves the attributes associated with the Interface Program from the customized database. The attributes may include network address, network mask, security level and other related information.
2. Invokes the **ifconfig** command to load the IF instance using the customized attributes. The **ifconfig** command will load the appropriate interface program if it has not already been loaded.
3. Calls the **ifconfig** command to attach a routine to establish a path between the interface instance and the adapter.
4. Sets the status of a particular IF instance to ″AVAILABLE″ in the customized database. All the IF instances are set to ″DEFINED″ at system reboot. When the **cfgif** method is invoked during boot time or from the command line, the IF instance(s) are then made available.

Flags

-l *InterfaceInstance*	Specifies the interface instance to configure. If the instance name is specified, only that Interface instance is configured. If this flag is not used, all Interface instances in the defined state are configured.
-2	Indicates that **ifconfig** will be invoked from the second phase of IPL so that a hex value will be shown on the front panel display. This flag should not be used during runtime.

Examples

1. To configure a particular token-ring IF instance, enter the following command. Note that `tr0` is the logical name for the token-ring IF instance. It should be defined using the **defif** method.

   ```
   cfgif -l tr0
   ```

2. To configure all IF instances, use the following command:

   ```
   cfgif
   ```

Related Information

The **chdev** command, **defif** method, **definet** method, **ifconfig** command, **mkdev** command.

The **cfginet** method.

The **odm_run_method** subroutine.

Object Data Manager (ODM) Overview for Programmers in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

Writing a Device Method in *AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts*.

TCP/IP network interfaces, TCP/IP addressing in *Networks and communication management*.

---

cfginet Method

Purpose

Loads and configures an Internet instance and its associated IF instances.

Syntax

cfginet [**-2**]

Description

The **cfginet** method loads and configures an instance of TCP/IP (an Internet instance) by performing the following steps:

1. Loads the protocol code.
2. Initializes entries in the Address Family Domain switch table and in the Network Input switch table.
3. Sets the status flag of the Internet instance to AVAILABLE.
4. Invokes the **hostname** command and the **route** command to set the hostname and static routes. The hostname and routing data are retrieved from the configuration database.

Note: The **cfginet** method is a programming tool and should not be executed from the command line.

Flag

-2 Specifies the second phase of IPL device configuration. A predetermined hex value will be displayed on the front panel. This option should not be used during regular run-time operation.

Example

To configure an Internet instance on a host, enter the method in the following format:

```
cfginet
```

Related Information

The **mkdev** command.

The **odm_run_method** subroutine.

TCP/IP network interfaces in *Networks and communication management*.

Object Data Manager (ODM) Overview for Programmers in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

Writing a Device Method in *AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts*.

---

cfgmgr Command

Purpose

Configures devices and optionally installs device software by running the programs specified in the Configuration Rules object class.

Syntax

cfgmgr [**-f** | **-s** | **-p** *Phase*] [**-i** *Device*] [**-l** *Name*] [**-v**]

Description

The **cfgmgr** command configures devices and optionally installs device software into the system. The configurable devices are controlled by the Configuration Rules object class, which is part of the Device Configuration database. Each configuration rule specifies the following:

- The full path name of an executable program to run
- When to run the program (in relation to the other rules)
- In which phase to run the program

During system boot, the **cfgmgr** command configures all the devices that are necessary to use the system. System boot is a two-step process:

1. Called phase 1, this step begins when the kernel is brought into the system and the boot file system is initialized. During this phase, the **cfgmgr** command is invoked, specifying this as phase 1 by using the **-f** flag. The **cfgmgr** command runs all of the phase 1 configuration rules, which results in the base devices being configured.
2. Phase 2 execution begins, and the **cfgmgr** command is called with the **-s** flag.

The **cfgmgr** command recognizes three phases of configuration rules:

- Phase 1
- Phase 2 (second boot phase for normal boot)
- Phase 3 (second boot phase for service boot)

The **cfgmgr** command runs all of the rules for the phase specified during invocation (for example, phase 1 rules for the **-f** flag). However, if the **-l** flag is used, the **cfgmgr** command configures only the named device and its children.

If the **cfgmgr** command is invoked without a phase option (for example, without the **-f**,**-s**, or **-p** flags), then the command runs the phase 2 rules. The only way to run the phase 3 rules is with the **-p** flag.

The configuration rules for each phase are ordered based on the values specified in the `seq` field. This field is an integer that specifies the priority in which to run this rule, relative to the other rules for this phase. The higher the number specified by the `seq` field, the lower the priority. For example, a value of 1 specified in the `seq` field is executed before a rule with a value of 10. There is one exception: a `seq` field value of 0 implies a ″don't care″ condition, and runs last. Therefore, a `seq` field value of 1 is the highest priority and runs first.

If there are any devices detected that have no device software installed when configuring devices, the **cfgmgr** command returns a warning message with the name or a list of possible names for the device

package that must be installed. If the specific name of the device package is determined, it is displayed as the only package name on a line below the warning message. If the specific name cannot be determined, a colon-separated list of possible package names is displayed on a single line. A package name or list of possible package names is displayed for each of the devices, if more than one device is detected without its device software.

The system displays the following warning message when devices without their device software are detected:

```
cfgmgr: 0514-621 WARNING: The following  device packages are
        required for device support but are not currently
        installed.
devices.pci.22100020
devices.pci.14101800
devices.pci.scsi:devices.pci.00100300:devices.pci.NCR.53C825
```

In this example, two devices missing software were found, and the **cfgmgr** command displays the names of the device packages that must be installed. A third device that is also missing software was found, but in this case, the **cfgmr** command displays several possible device package names.

When more than one possible package name is identified for a device, only one of the names will actually correspond to a device package on the installation medium. This is the package you must install. However, in some cases, more than one of the names will correspond to actual device packages on the installation medium. In this case, the first package name in the list for which there is a device package on the install medium is the package that must be installed. If the **cfgmgr** command is used with the **-i** flag, then the correct packages will be installed.

If you invoke the **cfgmgr** command with the **-i** flag, the command attempts to install device software automatically for each new detected device. The *Device* variable of the **-i** flag specifies where to find the installation medium. The installation medium can be a hardware device (such as a tape or diskette drive), a directory that contains installation images, or the installation image file itself.

> **Attention:** To protect the Configuration database, the **cfgmgr** command is not interruptible. Stopping this command before it is complete could result in a corrupted database.

Flags

-f	Specifies that the **cfgmgr** command runs the phase 1 configuration rules. This flag is not valid at run time (after system start).
-i *Device*	Specifies the location of the installation medium.
-l *Name*	Specifies the named device to configure along with the children of the device.
-p *Phase*	Specifies that the **cfgmgr** command runs the specified phase.
-s	Specifies that the **cfgmgr** command runs the phase 2 configuration rules.
-v	Specifies verbose output. The **cfgmgr** command writes information about what it is doing to standard output.

Configuration Rules

phase	Specifies whether this rule belongs to phase 1, phase 2, or phase 3 (second boot phase for service mode).
seq	Specifies the relative priority of this rule as an integer.
rule	A string containing the full path name of a program to execute. It can also contain any flags, but they must follow the program name as the whole string run as if it was typed on the command line.

Security

Access Control: Only the root user and members of the system group should have execute (x) access to this command.

Auditing Event:

Event	Information
DEV_Configure	Device name

Examples

These examples are based on the configuration rules containing the following information:

```
phase   seq        rule


1     10      /usr/lib/methods/defsys
1     12      /usr/lib/methods/deflvm


2     10      /usr/lib/methods/defsys
2     12      /usr/lib/methods/deflvm
2     13      /etc/methods/startusb
2     17      /etc/methods/cfgvlan -2
2     18      /usr/lib/methods/cfgrcnet
2     19      /usr/lib/methods/ptynode
2     20      /etc/methods/vconnode
2     20      /usr/lib/methods/startlft
2     22      /etc/methods/startrcm
2     25      /usr/lib/methods/starttty
2     27      /etc/methods/startsgio
2      0      /usr/lib/methods/defaio
2      0      /usr/lib/methods/def_posix_aio
2      0      /usr/lib/perf/cfg_perfstat load
2      0      /usr/lib/perf/load_blockset_ext


3     10      /usr/lib/methods/defsys
3     12      /usr/lib/methods/deflvm
3     13      /etc/methods/startusb
3     15      /usr/lib/methods/starttty
3     19      /usr/lib/methods/ptynode
3     20      /usr/lib/methods/startlft
3     20      /etc/methods/vconnode
3     22      /etc/methods/startrcm
3     27      /etc/methods/startsgio
```

1. When the **cfgmgr** command is invoked with the **-f** flag, the command gets all of the configuration rules with phase = 1 and runs them in the following order:

   ```
   /usr/lib/methods/defsys
   /usr/lib/methods/deflvm
   ```

   Note: The **-f** flag cannot be used during run time.

2. When the **cfgmgr** command is run with the **-s** flag, the command gets all of the configuration rules with phase = 2 and runs them in the following order:

   ```
   /usr/lib/methods/defsys
   /usr/lib/methods/deflvm
   /etc/methods/cfgvlan -2
   /usr/lib/methods/cfgrcnet
   /usr/lib/methods/ptynode
   /etc/methods/vconnode
   /usr/lib/methods/startlft
   /etc/methods/startrcm
   /usr/lib/methods/starttty
   /etc/methods/startsgio
   /usr/lib/methods/defaio
   /usr/lib/methods/def_posix_aio
   /usr/lib/perf/cfg_perfstat load
   /usr/lib/perf/load_blockset_ext
   ```

3. When the **cfgmgr** command is run with the **-p 3** flag, the command gets all of the configuration rules with phase = 3 and runs them in the following order:

```
/usr/lib/methods/defsys
/usr/lib/methods/deflvm
/etc/methods/startusb
/usr/lib/methods/starttty
/usr/lib/methods/ptynode
/usr/lib/methods/startlft
/etc/methods/vconnode
/etc/methods/startrcm
/etc/methods/startsgio
```

4. If the **cfgmgr** command is run without a flag, the command functions the same as when used with the -s flag. Thus, the phase 2 rules are run in the the following order:

```
/usr/lib/methods/defsys
/usr/lib/methods/deflvm
/etc/methods/cfgvlan -2
/usr/lib/methods/cfgrcnet
/usr/lib/methods/ptynode
/etc/methods/vconnode
/usr/lib/methods/startlft
/etc/methods/startrcm
/usr/lib/methods/starttty
/etc/methods/startsgio
/usr/lib/methods/defaio
/usr/lib/methods/def_posix_aio
/usr/lib/perf/cfg_perfstat load
/usr/lib/perf/load_blockset_ext
```

5. To configure detected devices attached to the `scsi0` adapter, type the following:

```
cfgmgr -l scsi0
```

6. To install device software automatically during configuration with the software contained in the `/usr/sys/inst.images` directory, type the following:

```
cfgmgr -i /usr/sys/inst.images
```

Files

/usr/sbin/cfgmgr	Specifies the command file.
/usr/include/sys/cfgdb.h	Contains numeric representations for fields in the Configuration Rules object class.

Related Information

The **chdev** command, **lsattr** command, **lsdev** command, **mkdev** command, **rmdev** command.

---

cfgqos Method

Purpose

Loads, configures, and activates the Quality of Service (QoS) instance.

Syntax

cfgqos

Description

The **cfgqos** method enables Quality of Service (QoS) for the TCP/IP protocol suite on a host by performing the following steps:

1. Loads the QoS kernel exension

2. Initializes the QoS instance
3. Attaches to the TCP/IP instance

Note: The **cfgqos** method is a programming tool and is not intended to be invoked from the command line.

Security

RBAC Disclaimer

Example

To configure QoS on a host, use the following format:

```
cfgqos
```

Related Information

The **cfginet** command, and **ucfgqos** method.

TCP/IP Quality of Service in the *Networks and communication management*.

---

cfgvsd Command

Purpose

cfgvsd – Configures a virtual shared disk.

Syntax

cfgvsd {**–a** | *vsd_name* ...}

Description

Use this command to configure the already defined virtual shared disks and bring them to the stopped state. This command does not make the virtual shared disk available.

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.

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

```
smit vsd_mgmt
```

and select the **Configure a virtual shared disk** option.

Flags

–a Specifies all virtual shared disks that have been defined.

Parameters

vsd_name Specifies a defined virtual shared disk.

Security

You must have root authority to run this command.

Restrictions

Under normal circumstances, you should not issue this command. The RVSD subsystem uses this command to manage shared disks in a controlled manner. If you issue this command, the results may be unpredictable.

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

Examples

To bring the virtual shared disk **vsd1vg1n1** from the defined state to the stopped state, enter:

```
cfgvsd vsd1vg1n1
```

Location

/opt/rsct/vsd/bin/cfgvsd

Related Information

Commands: **ctlvsd**, **lsvsd**, **preparevsd**, **resumevsd**, **startvsd**, **stopvsd**, **suspendvsd**, **ucfgvsd**

---

cflow Command

Purpose

Generates a C and C++ flow graph of external references.

Syntax

cflow [**-d** *Number*] [**-I** *Directory*] [**-i** _] [**-i p**] [**-i x**] [**-q***Option*] [**-r**] [**-MA**] [**-U** *Name*] [**-Nd***Number*] [**-Nl***Number*] [**-Nn***Number*] [**-Nt***Number*] [**-D** *Name*[**=***Definition*]] *File ...*

Description

The **cflow** command analyzes the C, C++, **yacc**, **lex,** assembler, and object files and writes a chart of their external references to standard output.

> **Note:** Processing of C++ language files by the **cflow** command requires the presence of the IBM C Set++ Compiler/6000 package.

The **cflow** command sends files with the **.y**, **.l**, and **.c** suffixes to the **yacc** command, **lex** command, and **cpp** command for processing. A modified first pass of the **lint** command then processes the **yacc**, **lex**, and **cpp** output, or any **.i** files. The **cflow** command sends files with a **.C** suffix to the C Set++ compiler.

The **cflow** command assembles files with the **.s** suffix, extracting information from the symbol table (as it does with **.o** files). From this output, the **cflow** command produces a graph of external references and writes it to standard output.

Each line of output provides the following information (in order from left to right):

- A line number followed by sufficient tabs to indicate the level of nesting
- The name of the global, a colon, and its definition.

The name is normally a function not defined as external and not beginning with an underline character (see the **-i**_ and **-i** inclusion flags).

For information extracted from C and C++ source files, the definition consists of an abstract type declaration (for example, char *), the name of the source file surrounded by angle brackets, and the line number on which the definition was found. Definitions extracted from object files contain the file name and location counter under which the symbol appeared, such as **.text** or **.data**. The **cflow** command deletes leading underline characters in C-style external names.

Once the **cflow** command displays a name, later references to the name contain only the **cflow** line number where the definition can be found. For undefined references, **cflow** displays only < > (angled brackets).

If the nesting level becomes too deep to display in available space, pipe the output from the **cflow** command to the **pr** command, using the **-e** flag to compress the tab expansion to less than eight spaces per tab stop.

> **Note:** To ensure that the line numbers produced by the **cflow** command match your **lex** and **yacc** files, you must send the **.l** or **.y** file to the **cflow** command.

Flags

-d *Number*	Sets to a decimal integer the depth at which the flow graph is cut off. By default this is a large number. Do not set the cutoff depth to a nonpositive integer.
-i _	Includes names that begin with an underline character. The default excludes these functions (and corresponding data if the **-ix** flag is used).
-i p	Disables ANSI function prototypes. The default option is to fill in undefined function information with available prototype declarations.
-i x	Includes external and static data symbols. The default includes only functions.
-r	Produces an inverted listing that shows the callers of each function, sorted by called function.
-MA	Specifies ANSI mode. The **cflow** command expects ANSI C code in this mode. The default mode of operation is extended mode.
-Nd*Number*	Changes the dimension table size to the *Number* parameter. The default value of *Number* is 2000.
-Nl*Number*	Changes the number of type nodes to the *Number* parameter. The default value of *Number* is 8000.
-Nn*Number*	Changes the symbol table size to the *Number* parameter. The default value of *Number* is 1500.
-Nt*Number*	Changes the number of tree nodes to the *Number* parameter. The default value of *Number* is 1000.

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

-D *Name*[**=***Definition*]	Defines the *Name* parameter, as if by the **#define** statement. The default *Definition* is 1.
-q*Option*	Passes the **-q***Option* to the preprocessor. For example, `-qmbcs` sets multibyte mode specified by the current locale and `-qidirfirst` modifies the search order for files included with the **#include file_name** directive.
-I *Directory*	Adds the specified *Directory* to the list of directories in which the **cflow** program searches for **#include** files.
-U *Name*	Removes any initial definition of the *Name* parameter, where *Name* is a reserved symbol that is predefined by the particular preprocessor.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Examples

1. To generate a default flow graph of these C files that compose a program, enter:
   ```
   cflow timeout.c kill.c error.c
   ```
2. To produce a **cflow** graph with a single level of nesting of functions, enter:
   ```
   cflow -d1 resam.c pptp.c ptpt.c rrr.c whn.c
   ```
3. To generate a **cflow** graph of a **lex** program, enter:
   ```
   cflow scan.l
   ```
4. To generate a **cflow** graph of the **yacc** program, enter:
   ```
   cflow yaccfile.y
   ```
5. To generate an inverted listing showing the callers of each of the functions in the C files used in example 2, enter:
   ```
   cflow -r resam.c pptp.c ptpt.c rrr.c whn.c
   ```

Files

/usr/ccs/bin/cflow	Driver for the **cflow** command
/usr/ccs/lib/cflow1	Executable for the **cflow** command
/usr/ccs/lib/dag	Executable for the **cflow** command
/usr/ccs/lib/flip	Executable for the **cflow** command
/usr/ccs/lib/lpfx	Executable for the **cflow** command
/usr/ccs/lib/nmf	Executable for the **cflow** command
/var/tmp/cf.*	Temporary files created by the **cflow** command

Related Information

The **as** command, **cpp** command, **lex** command, **lint** command, **nm** command, **pr** command, **yacc** command.

---

cfsadmin Command

Purpose

Administers disk space used for caching file systems with the Cache File-System (CacheFS).

Syntax

cfsadmin -c [-o ***param=****n* [,***param=****n*]] *cache_directory*

cfsadmin -d *cacheID*|**all** *cache_directory*

cfsadmin -l *cache_directory*

cfsadmin -s mntpnt . . .|**all**

cfsadmin -u *cache_directory*

Description

The **cfsadmin** command provides the following functions:

- Cache creation
- Deletion of cached file systems
- Listing of cache contents and statistics
- Resource parameter adjustment when the file system is unmounted.

For each form of the command, unless the **-u** flag is specified, you must specify a cache directory, that is, the directory under which the cache is actually stored. A path name in the front file system identifies the cache directory. When the **-s** flag is used, you must specify a mount point.

You can specify a cache ID when you mount a file system with CacheFS, or you can let the system generate one for you. The **-l** flag includes the cache ID in its listing of information. You must know the cache ID to delete a cached file system.

Flags

-c *cache_directory*
Creates a cache under the directory specified by *cache_directory*. This directory must not exist prior to cache creation.

-d
Removes the file system whose cache ID you specify and release its resources, or remove all file systems in the cache by specifying *cache_directory*. After deleting a file system from the cache, you must run the command to correct the resource counts for the cache.

-l *cache_directory*
Lists file systems stored in the specified cache, as well as statistics about them. Each cached file system is listed by cache ID. The statistics document resource utilization and cache resource parameters.

-o [***param=****n*]*cache_directory*
Allows changing parameter values by using “CacheFS Resource Parameters” as arguments.

-s *cache_directory*
Requests a consistency check on the specified file system (or all **cachefs** mounted file systems). The **-s** flag only works if the cache file system was mounted with **demandconst** enabled. Each file in the specified cache file system is checked for consistency with its corresponding file in the back file system. The consistency check is performed file by file as files are accessed. If no files are accessed, no checks are performed. Using this flag does not result in a sudden storm of consistency checks. The **-s** flag is not currently supported in this operating systems CacheFS.

-u *cache_directory*
Updates resource parameters of the specified cache directory. Parameter values can only be increased. To decrease the values, you must remove the cache and recreate it. All file systems in the cache directory must be unmounted when you use this flag. Changes will take effect the next time you mount any file system in the specified cache directory.
Note: The **-u** flag with no **-o** flag sets all parameters to their default values.

CacheFS Resource Parameters

You can specify the following cacheFS resource parameters as arguments to the **-o** flag. Separate multiple parameters with commas.

maxblocks=*n*
Maximum amount of storage space that CacheFS can use, expressed as a percentage of the total number of blocks in the front file system. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the space the **maxblocks** parameter allows will be available. The default is 90.

minblocks=*n*
The minimum amount of storage space, expressed as a percentage of the total number of blocks in the front file system, that CacheFS is always allowed to use without limitation by its internal control mechanisms. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the space the **minblocks** parameter attempts to reserve will be available. The default is 0.

threshblocks=*n* — A percentage of the total blocks in the front file system beyond which CacheFS cannot claim resources once its block usage has reached the level specified by **minblocks**. The default is 85.

maxfiles=*n* — Maximum number of files that CacheFS can use, expressed as a percentage of the total number of inodes in the front file system. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the inodes the **maxfiles** parameter allows will be available. The default is 90.

minfiles=*n* — Minimum number of files, expressed as a percentage of the total number of inodes in the front file system, that CacheFS is always allowed to use without limitation by its internal control mechanisms. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the inodes the **minfiles** parameter attempts to reserve will be available. The default is 0.

threshfiles=*n* — A percentage of the total inodes in the front file system beyond which CacheFS cannot claim inodes once its usage has reached the level specified by **minfiles**. The default is 85.

maxfilesize=*n* — Largest file size, expressed in megabytes, that CacheFS is allowed to cache. The default is -1, which means there is no limit on the largest file size.

Note: You cannot decrease the block or inode allotment for a cache. To decrease the size of a cache, you must remove it and create it again with different parameters.

Examples

1. To create a cache directory named **cache**, enter:
   ```
   cfsadmin -c /cache
   ```
2. To create a cache directory named **/cache1** that can claim a maximum of 60 percent of the blocks in the front file system, can use 40 percent of the front file system blocks without interference by CacheFS internal control mechanisms, and has a threshold value of 50 percent. The threshold value indicates that after CacheFS reaches its guaranteed minimum, it cannot claim more space if 50 percent of the blocks in the front file system are already used.
   ```
   cfsadmin -c -o maxblocks=60,minblocks=40,threshblocks=50 /cache1
   ```
3. To change the **maxfilesize** parameter for the cache directory **/cache2** to 2 megabytes, enter:
   ```
   cfsadmin -u -o maxfilesize=2 /cache2
   ```
4. To list the contents of a cache directory named **/cache3** and provides statistics about resource utilization, enter:
   ```
   cfsadmin -l /cache3
   ```
5. To remove the cached file system with cache ID 23 from the cache directory **/cache3** and free its resources (the cache ID is part of the information returned), enter:
   ```
   cfsadmin -d 23 /cache3
   ```
6. To remove all cached file systems from the **/cache3** directory, enter:
   ```
   cfsadmin -d all /cache3
   ```
7. To check all filesystems mounted with **demandconst** enabled for consistency. No errors will be reported if no **demandconst** filesystems were found. Enter:
   ```
   cfsadmin
   ```

Related Information

The **mount** command and **fsck_cachefs** command.

chargefee Command

Purpose
Charges end users for the computer resources they use.

Syntax
/usr/sbin/acct/chargefee *User Number*

Description
The **chargefee** command is used by someone with administrative authority to charge the individual specified by the *User* parameter for the number of work units specified by the *Number* parameter. The *Number* value can be an integer or a decimal value.

The **chargefee** command writes a record to the **/var/adm/fee** file. This information is merged with other accounting records by the **acctmerg** command to create the daily report.

> **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.

Examples
To charge `smith` for `10` units of work on a financial report, enter:

```
/usr/sbin/acct/chargefee smith 10
```

A record is created in the **/var/adm/fee** file, which the **acctmerg** command will merge with records in other accounting files to produce the daily report.

Files

/usr/sbin/acct	The path to the accounting commands.
/var/adm/fee	Accumulates the fees charged to each login name.

Related Information
The **acctmerg** command.

For more information about the Accounting System, the preparation of daily and monthly reports, and the accounting files, see the System accounting in *Operating system and device management*.

Setting up an accounting subsystem in *Operating system and device management* explains the steps you must take to establish an accounting system.

chauth Command

Purpose
Changes user-defined authorization attributes.

Syntax
chauth [**-R** *load_module*] *Attribute* **=** *Value ... Name*

Description

The **chauth** command modifies attributes for the authorization that is identified by the *Name* parameter. The command only modifies existing user-defined authorizations in the authorization database. System-defined authorizations cannot be modified with the **chauth** command. To change an attribute of a user-defined authorization, specify the attribute name and the new value with the *Attribute* **=** *Value* parameter. If any specified attribute or attribute value is not valid, the **chauth** command does not modify the authorization.

Important: Modifying the ID of an authorization can affect the system security because the current value of the ID might be used by some processes, files, and so on. In general, use the **id** attribute to modify the ID of an authorization when you are sure that the authorization is not used. The **chauth** command only allows the ID to be set to an unused value greater than 10 000. IDs less than 10 000 are reserved for system-defined authorizations.

If the system is configured to use multiple domains for the authorization database, authorization modification is performed according to the order specified by the **secorder** attribute of the authorizations database stanza in the **/etc/nscontrol.conf** file. Only the first matching authorization is modified. Duplicate authorizations from the remaining domains are not modified. Use the **-R** flag to modify the authorization from a specific domain.

When the system is operating in enhanced Role Based Access Control (RBAC) mode, modifications made to the authorization database are not used for security considerations until the database is sent to the kernel security tables through the **setkst** command.

Flags

-R *load_module*	Specifies the loadable module to use for the authorization modification.

Attributes

id	Specifies a unique integer that is used to identify the authorization. The value is a decimal integer ranging from 10 001 through 32 768.
dfltmsg	Specifies the default description to use if message catalogs are not in use. The value is a string.
msgcat	Specifies the message catalog file name containing the description of the authorization. If the **msgcat** attribute is specified, the **msgset** and **msgnum** attributes must also be specified. The value is a string. If the specified string contains a leading forward slash (/), the value is assumed to be an absolute path name. Otherwise, the user environment defines the directory search path as specified by the **catopen** routine.
msgset	Specifies the message set number in the file name to retrieve the message number. The file name is specified by the **msgcat** attribute, and the message number is specified by the **msgnum** attribute. The value is a decimal integer.
msgnum	Specifies the message number for the description of the authorization in the file and the set. The authorization is specified by the **msgcat** attribute, and the set number is specified by the **msgset** attribute. The value is a decimal integer.

Parameters

Name	Specifies the authorization to modify.

Security

The **chauth** command is a privileged command. You must assume a role that has the following authorization to run the command successfully.

aix.security.auth.change	Required to run the command.

RBAC Disclaimer

Files Accessed

File	Mode
/etc/security/authorizations	rw

Examples

1. To change the message catalog used to provide the authorization description for the `custom` authorization, use the following command:

   ```
   chauth msgcat="custom_auths.cat" custom
   ```

2. To change the message set and number that designates the authorization description for the `custom.test` authorization, use the following command:

   ```
   chauth msgset=5 msgnum=24 custom.test
   ```

3. To change the message catalog for the `custom.test` authorization in LDAP, use the following command:

   ```
   chauth -R LDAP msgset=5 custom.test
   ```

Related Information

The **mkauth**, **lsauth**, **rmauth**, **setkst** and **ckauth** commands.

The **getauthattr** subroutine, **getauthattrs** subroutine, **putauthattr** subroutine, **putauthattrs** subroutine.

The **/etc/security/authorizations**, **/etc/nscontrol.conf** , and **/usr/lib/security/methods.cfg** files.

RBAC in *Security*.

---

chauthent Command

Purpose

Changes the configured authentication methods for the system.

Syntax

chauthent [**-k5**] [**-k4**] [**-std**]

Description

The **chauthent** command sets the desired configuration based on the flags the user sets. The authentication methods are set in the order in which the flags are given to the command. If none of the flags are set, then the **rcmds** will be disabled from functioning. If the **-std** flag is set, it must be the last flag set or the command will fail.

> **Note:** The complete order of authentication methods must be specified each time. The command does not modify the current order when replacing it with the new one.

The user must have root authority to execute the command.

The **chauthent** command takes the flags set and calls the **set_auth_method** routine in **libauthm.a** to cause the change.

The **chauthent** command writes an error message to **stderr** and returns a -1 if **set_auth_method** fails.

Flags

-k5 Sets the Kerberos 5 authentication method.
-k4 Sets the Kerberos 4 authentication method.
-std Sets the Standard operating system authentication method.

Security

RBAC Disclaimer

Examples

1. Set all of the methods in descending order:
   ```
   chauthent  -k5   -k4  -std
   ```
2. Set all of the methods with Kerberos 4 attempted first:
   ```
   chauthent -k4 -k5 -std
   ```
3. Clear all of the methods:
   ```
   chauthent
   ```

Related Information

The **ftp** command, **lsauthent** command, **rcp** command, **rlogin** command, **rsh** command, **telnet**, **tn**, or **tn3270** command.

The **get_auth_method** and **set_auth_method** routines.

Communications and networks in *Networks and communication management*.

Authentication and the secure rcmds in *Networks and communication management*..

---

chC2admin Command

Purpose

Changes the name of the administrative host for a system.

Syntax

chC2admin [**-a** *address*] *hostname*

Description

The **chC2admin** command maintains the name of the C2 System Administrative Host as well as the NFS mount points and hostname entries as defined in **/etc/filesystems**.

Changing the name of the Administrative Host will cause the NFS file systems listed in **/etc/filesystems** to be updated and the contents of **/etc/security/admin_host** to be replaced.

The given *hostname* must be defined when this command is executed. If *hostname* cannot be resolved, a warning will be given. The **-a** option may be used to specify the IP address of hostname. When the **-a** option is given, *hostname* and *address* will be added to the **/etc/hosts** file.

Flags

-a *address*

Parameters

hostname	Specifies the hostname.

Exit Status

0	All updates have been made successfully.
1	Command has been executed on a non-C2 System.
2	Command failed while changing the administrative host.

Files

/usr/sbin/chC2admin	Contains the **chC2admin** commnad.

Related Information

The **isC2host** command, **lsC2admin** command, **mkC2admin** command, **rmC2admin** command.

---

chCCadmin Command

Purpose

Changes the name of the Common Criteria enabled System Administrative Host for a system.

Syntax

chCCadmin [**-a** *address*] *hostname*

Description

The **chCCadmin** command maintains the name of the Common Criteria enabled System Administrative Host as well as the NFS mount points and hostname entries as defined in **/etc/filesystems**.

Changing the name of the Administrative Host will cause the NFS file systems listed in **/etc/filesystems** to be updated and the contents of **/etc/security/admin_host** to be replaced.

The given *hostname* must be defined when this command is executed. If *hostname* cannot be resolved, a warning will be given. The **-a** option may be used to specify the IP address of hostname. When the **-a** option is given, *hostname* and *address* will be added to the **/etc/hosts** file.

Flags

-a *address*

Parameters

hostname Specifies the hostname.

Exit Status

0 All updates have been made successfully.

1 Command has been executed on a non-Common Criteria enabled System.

2 Command failed while changing the administrative host.

Files

/usr/sbin/chCCadmin Contains the **chCCadmin** commnad.

Related Information

The **isCChost** command, **lsCCadmin** command, **mkCCadmin** command, **rmCCadmin** command.

---

chcifscred Command

Purpose

Changes the password for a specific server/user entry stored in the **/etc/cifs_fs/cifscred** file.

Syntax

chcifscred -h *RemoteHost* **-u** *user* [**-p** *password*]

Description

The **chcifscred** command takes a server and user name as input. If this input has credentials listed in **/etc/cifs_fs/cifscred**, the command line prompts for a password to replace the existing password. The password is stored in an encrypted format.

Flags

-h *RemoteHost* 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.

-p *password* Specifies the new password for the specified user on the specified remote host.

-u *user* Specifies the user name whose password is changing for access to the specified host.

Exit Status

0 The command completed successfully.

>0 An error occurred.

Examples

1. To change the password stored for `user1` to mount on `server1`, with `server1` and `user1` credentials already residing in **/etc/cifs_fs/cifscred**, enter:

```
chcifscred -h server1 -u user1
```

Location

/usr/sbin/chcifscred

Files

/etc/cifs_fs/cifscred	Stores the CIFS credentials.

Related Information

The "chcifsmnt Command," lscifscred command, lscifsmnt command, mkcifscred command, mkcifsmnt command, rmcifscred command, rmcifsmnt command.

---

chcifsmnt Command

Purpose

Changes the mount options, server name, share, or credentials for a CIFS mount.

Syntax

chcifsmnt -f *MountPoint* [**-d** *RemoteShare*] [**-h** *RemoteHost*] [**-c** *user*] [**-p** *password*] [**-m** *MountTypeName*] [**-A**|**-a**] [**-l**|**-B**|**-N**] [**-t** {**rw**|**ro**}] [**-u** *uid*] [**-g** *gid*] [**-x** *fmode*] [**-w** *wrkgrp*]

Description

The **chcifsmnt** command changes the mount options, server name, share name, or credentials for a CIFS mount defined in **/etc/filesystems** file. If the share is not mounted, it will be mounted after the changes to the **/etc/filesystems** file are made. If the share is not already defined in **/etc/filesystems**, an error is returned.

Flags

-a	Specifies that the **/etc/filesystems** entry for this file system should not be automatically mounted at system restart. This is the default.
-A	Specifies that the **/etc/filesystems** entry for this file system should be automatically mounted at system restart.
-B	Specifies that the **/etc/filesystems** entry should be modified and that it should be remounted with the options specified. This is the default.
-c *user*	Specifies user name used to gain access to the CIFS share.
-d *RemoteShare*	Specifies the share name on the CIFS server that should be mounted.
-f *MountPoint*	Specifies the path name over which the CIFS share should be mounted.
-g *gid*	Specifies the GID that is assigned to files in the mount. The default is 0.
-h *RemoteHost*	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.
-l	Specifies that the **/etc/filesystems** entry should be modified, but should not be remounted.

-m *MountTypeName* Defines the mount type that will be added to the **/etc/filesystems** file, which allows for mounting all file systems of a specific type using the **-t** option of the **mount** command. By default, no type value will be added to **/etc/filesystems**.

-N Remounts the CIFS share with the options specified, but does not modify the **/etc/filesystems** file.

-p *password* 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 **cifscred** file (the password will be encrypted). If the **-p** option is not specified, and the credentials do not already exist in the **cifscred** file, the command line prompts the user to provide the password, and the credentials will be added to the **cifscred** file. If the server/user credentials already exist in the **cifscred** 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 *uid* Specifies the UID that is assigned to files in the mount. The default is 0.

-x *fmode* Specifies the owner, group, and other permission bits assigned to files in the mount. The default is 755.

-w *wrkgrp* 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

0 The command completed successfully.

>0 An error occurred.

Examples

1. To change the user name to `user1` for a CIFS mount defined on **/mnt**, enter:

   ```
   chcifsmnt -f /mnt -c user1
   ```

Location

/usr/sbin/chcifsmnt

Files

/etc/cifs_fs/cifscred Stores the CIFS credentials.

/etc/filesystems Stores the CIFS entry.

Related Information

The “chcifscred Command” on page 350, lscifscred command, lscifsmnt command, mkcifscred command, mkcifsmnt command, rmcifscred command, rmcifsmnt command.

chclass Command

Purpose

Change the attributes and resource entitlements of a Workload Management class.

Syntax

chclass -a *Attribute=Value* {[**-a** *Attribute=Value*]...} [**-c** | **-m** | **-b** | **-v** | **-C** | **-B** | **-P** | **-T** | **-L** | **-V** | **-A** *KeyWord=Value*] [**-d** *Config_Dir*] [**-S** *SuperClass*] *Name*

Description

The **chclass** command changes attributes for the class identified by the *Name* parameter. The class must already exist. To change an attribute, specify the attribute name and the new value with the *Attribute=Value* parameter. To change a limit or shares value, use option **-c** for cpu, **-m** for memory, and **-b** for disk I/O throughput, with the keyword value in **min**, **softmax**, **hardmax** or **shares**. 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).

> **Note:** Only the root user can change the attributes of 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 change the attributes of a subclass of this superclass.

Normally, **chclass** updates the attributes of a class in the relevant WLM property files, and the modifications are applied to the in-core class definition (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 change applies only to the in-core class attributes, and no property file is updated, making the changes 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:

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).

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.

iopriority Specifies the priority that is assigned to I/O requests. The I/O requests are issued by the threads that are classified to the class. The priority is used to prioritize I/O buffers at the device level. If the storage device does not support I/O priorities, the priority is ignored. Valid I/O priority values range from 0 through 15.

Class limits and shares for CPU, memory, or disk I/O resource:

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).

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 *AIX 5L Version 5.3 Files Reference*.

Class description:

description The class description text can be composed of any ASCII character, except colons (**:**) and commas (**,**).

Flags

-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 *KeyWord*=*Value* Changes a limit or shares value for disk I/O throughput. Possible *KeyWord*s are **min**, **softmax**, **hardmax**, or **shares**.

-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 *KeyWord*=*Value* Changes a limit or shares value for a CPU. Possible *KeyWord*s are **min**, **softmax**, **hardmax**, or **shares**.

-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**).

-d *Config_Dir* Uses the **/etc/wlm/***Config_Dir* directory as alternate directory for the properties files. If this flag is not present, the current configuration files in the directory pointed to by **/etc/wlm/current** are used. If an empty string is passed as the configuration name (`-d ""`) the modifications only affect the in-core class definition 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.

-m *KeyWord*=*Value* Changes a limit or shares value for memory. Possible *KeyWord*s are **min**, **softmax**, **hardmax**, or **shares**.

-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 changing the attributes of a subclass. There are two ways of specifying that the change is to be applied to the subclass Sub of superclass Super:

1. Specify the full name of the subclass as Super.Sub and not use -S.
2. Uses the -S flag to give the superclass name and use the short name for the subclass:

   ```
   chclass options -S Super  Sub
   ```

-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

RBAC Disclaimer

Files

classes	Contains the names and definitions of the classes.
limits	Contains the resource limits enforced on the classes.
shares	Contains the resource shares attributes for each class.

Related Information

The **wlmcntrl** command, **lsclass** command, **mkclass** command, and **rmclass** command.

---

chcod Command

Purpose

Manages Capacity Upgrade on Demand.

Syntax

chcod [**-r** *ResourceType* **-n** *NbrResources*] [**-c** *CustomerInfo*] [**-m** *MailAddr*] [**-h**]

Description

The **chcod** command manages Capacity Upgrade on Demand, or CUoD. CUoD enables the authorization of more *ResourceTypes*, such as processors, on the system than were initially authorized. The additional resources may be enabled if they are available, and if the system supports CUoD for that *ResourceType*. Only one *ResourceType* may be managed at a time. The change in the number of *ResourceTypes* takes effect after the next system boot. CUoD management also includes displaying the current number of *ResourceType*(s) that have CUoD support, monitoring the number of *ResourceType*(s) on the system, and notifying appropriately. Notification occurs on a monthly basis and also whenever *NbrResources* changes.

Notification takes the form of error logging and, optionally, sending e-mail. An entry is made in the system error log whenever the specified *ResourceType* changes and also on a monthly basis. The *CustomerInfo* text is included in the error log. If you specify an e-mail address with *MailAddr*, notification also occurs through an e-mail message sent to *MailAddr*. The *CustomerInfo* text is included in the text of the message. You can have notification by both error logging and e-mail if you specify both *CustomerInfo* and *MailAddr*.

With no flags specified, **chcod** displays the current value of *CustomerInfo*, *MailAddr*, the system's model name and serial number, and the current value(s) of *NbrResources* for any *ResourceType* that has CUoD support.

Note: Beginning with the IBM p650 and later models (all POWER4 Systems), CUoD is managed at the Hardware Management Console (HMC).

Flags

-c *CustomerInfo*
: Specifies the text string to include in the error log. This string is also included in the body of any e-mail message sent. *CustomerInfo* may not be more than 255 characters. Blanks may not be included in the string. After*CustomerInfo* has been specified, subsequent **chcod** uses do not have to specify the **-c** flag, but you do have the option of changing it. *CustomerInfo* may consist of alphanumeric characters and any of **.** (period), **,** (comma), **-** (hyphen), **(** (open parenthesis), or **)** (close parenthesis).

-h
: Displays the usage message.

-m *MailAddr*
: Specifies the e-mail address to which e-mail should be sent. *MailAddr* may not be more than 255 characters. If *MailAddr* is reset by specifying "" (a blank string), then only error logging will monitor the resources that have CUoD support. You must have e-mail configured on your system if you want to send notification to this e-mail address.

-n *NbrResources*
: Specifies the number of *ResourceTypes* to be authorized on the system. It must be zero or greater. If it is 0, CUoD is disabled for the specified *ResourceType*. If **-n** is specified, then **-r** must also be specified.

-r *ResourceType*
: Specifies the *ResourceType*, for example, `proc` for processors, to be enabled and monitored on the system. The system must support CUoD for *ResourceType*. If **-r** is specified, then **-n** must also be specified.

Examples

1. To initiate CUoD for processors, type:

   ```
   chcod -r proc -n 10 -m"someone@ibm.location.com" -c"Jane_Doe-Customer_Number_999999-(111)111-1111"
   ```

2. To change the *CustomerInfo*, type:

   ```
   chcod -c"Jane_Doe-Customer_Number_999999-(222)222-2222"
   ```

3. To stop the e-mail form of notification, type:

   ```
   chcod -m""
   ```

4. To see the current values of the resources with CUoD support, type:

   ```
   chcod
   ```

   A message similar to the following will be displayed:

   ```
   Current CustomerInfo = Jane_Doe-Customer_Number_999999-(222)222-2222
   Current MailAddr = someone@ibm.location.com
   Current model and serial number = IBM,7043-150 000974934C00
   Current number of authorized processors = 10 of 12 installed on system
   ```

chcomg Command

Purpose

Changes a previously-defined communication group for a peer domain.

Syntax

To change an attribute of a communication group:

chcomg [**–s** *sensitivity*] [**–p** *period*] [**–t** *priority*] [**–b**] [**–r**] [**–x b** | **r** | **br**] [**–e** *NIM_path*] [**–m** *NIM_parameters*] [**–h**] [**–TV**] *communication_group*

To change a reference in an interface resource to a different communication group:

chcomg [**–i n:***network_interface1*[**:***node1*][**,***network_interface2*[**:***node2*]...] | **–S n:***"network_interface_selection_string"*] [**–h**] [**–TV**] *communication_group*

Description

The **chcomg** command changes an existing communication group definition with the name specified by the *communication_group* parameter for the online peer domain. The communication group is used to define heartbeat rings for use by topology services and to define the tunables for each heartbeat ring. The communication group determines which devices are used for heartbeating in the peer domain.

The **chcomg** command must be run on a node that is currently online in the peer domain where the communication group is defined. One or more attributes can be changed with one **chcomg** command, but at least one change is required.

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.

The **chcomg** command can also be used to assign a communication group to an interface resource. 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. Before a communication group can be removed, any interface resources that refer to it must be reassigned.

More than half of the nodes must be online to change a communication group in the domain.

Flags

-s *sensitivity*
: Specifies the heartbeat sensitivity. This is the number of missed heartbeats that constitute a failure. The sensitivity is an integer greater than or equal to 4.

-p *period*
: Specifies the period, which is the number of seconds between heartbeats. The value of *period* can be an integer or a floating-point number that is greater than or equal to **1**.

-t *priority*
: Specifies the priority. The priority indicates the importance of this communication group with respect to others. It is used to order the heartbeat rings. The lower the number, the higher the priority. The highest priority is **1**.

-b
: Specifies that broadcast will be used if the underlying media support it. The **-b** flag cannot be used when specifying **-x b**.

-r
: Specifies that source routing will be used if the underlying media support it. The **-r** flag cannot be used when specifying **-x r**.

–x b | r | br
: Excludes control for the heartbeat mechanism. This indicates that one or more controls for heartbeat mechanisms should not be used even if the underlying media support it. The following can be excluded:

  b
  : Specifies that broadcast should not be used even if the underlying media support it.

  r
  : Specifies that source routing should not be used even if the underlying media support it.

Excluding more than one control is specified by listing the feature option letters consecutively (**-x br**).

-i n:*network_interface1***[:***node1***] [,***network_interface2***[:***node2***]...**
Assigns this communication group to the network interface resource defined by the network interface resource name and optionally the node name where it can be found.

If **-i** is specified, **-S** cannot be specified.

-S n: *"network_interface_selection_string"*
Assigns this communication group to the interface specified by the network interface selection string.

If **-S** is specified, **-i** cannot be specified.

-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 is a character string that is passed to the NIM when starting it.

-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 an existing communication group to be changed in the peer domain.

Security

The user of the **chcomg** 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 changed.

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, node **nodeA** is defined and online to peer domain **ApplDomain**.

1. To change the communication group **ComGrp1** for **ApplDomain** to a sensitivity of 4 and period of 3, run this command on **nodeA**:
   ```
   chcomg -s 4 -p 3 ComGrp1
   ```
2. To change the communication group **ComGrp1** for **ApplDomain** to use broadcast, run this command on **nodeA**:
   ```
   chcomg -b ComGrp1
   ```
3. To change the communication group **ComGrp1** for **ApplDomain** to no longer use source routing, run this command on **nodeA**:
   ```
   chcomg -x r ComGrp1
   ```
4. To change the communication group **ComGrp1** for **ApplDomain**, to use a NIM path of **/usr/sbin/rsct/bin/hats_nim**, and to use NIM parameters **-l 5** to set the logging level, run this command on **nodeA**:
   ```
   chcomg -e /usr/sbin/rsct/bin/hats_nim -m "-l 5" ComGrp1
   ```
5. To assign the communication group **ComGrp1** for **ApplDomain** to the network interface resource named **eth0** on **nodeB**, run this command on **nodeA**:
   ```
   chcomg -i n:eth0:nodeB ComGrp1
   ```
6. To assign the communication group **ComGrp1** for **ApplDomain** to the network interface resource that uses the subnet 9.123.45.678, run this command on **nodeA**:
   ```
   chcomg -S n:"Subnet == '9.123.45.678'" ComGrp1
   ```

Location

/usr/sbin/rsct/bin/chcomg

Related Information

Books: *RSCT: Administration Guide*, for information about peer domain operations

Commands: **lscomg**, **lsrpdomain**, **lsrpnode**, **mkcomg**, **preprpnode**, **rmcomg**

Information Files: **rmccli**, for general information about RMC-related commands

---

chcondition Command

Purpose

Changes any of the attributes of a defined condition.

Syntax

To change a condition's attributes:

chcondition [**–r** *resource_class*]
[**–e** *"event_expression"*] [**–E** *"rearm_expression"*]
[**–d** *"event_description"*] [**–D** *"rearm_description"*]
[**–m l** | **m** | **p**] [**–n** *node_name1*[**,***node_name2*...]]
[**––qnotoggle** | **––qtoggle**]
[**–s** *"selection_string"*] [**–S c** | **w** | **i**]
[**–h**] [**–TV**] *condition*[**:***node_name*]

To rename a condition:

chcondition –c *new_condition* [**–h**] [**–TV**] *condition*[**:***node_name*]

To lock or unlock a condition:

chcondition { **-L** | **-U** } [**–h**] [**–TV**] *condition*[**:***node_name*]

Description

The **chcondition** command changes the attributes of a defined condition to the values supplied. If the name of the condition is changed using the **-c** flag, any condition/response associations remain intact.

If a particular condition is needed for system software to work properly, it may be locked. A locked condition cannot be modified or removed until it is unlocked. If the condition you specify on the **chcondition** command is locked, it will not be modified; instead an error will be generated informing you that the condition is locked. To unlock a condition, you can use the **-U** flag. However, since a condition is typically locked because it is essential for system software to work properly, you should exercise caution before unlocking it. To lock a condition so it cannot be modified, use the **-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

–c *new_condition*
Assigns a new name to the condition. *new_condition*, which replaces the current name, is a character string that identifies the condition. If *new_condition* contains one or more spaces, it must be enclosed in quotation marks. A name cannot be null, consist of all spaces, or contain embedded double quotation marks.

–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 a dynamic attribute of *resource_class*, a mathematical comparison symbol (**>**, for example), and a constant.

–d *"event_description"*
Describes the event expression.

–D *"rearm_description"*
Describes the rearm expression.

–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. The condition applies only to the local node (the node where the condition is defined). Only the local node is used in evaluating the selection string.

–L Locks a condition so it cannot be modified or removed. When locking a condition using the **-L** flag, no other operation can be performed by this command.

m Specifies *management domain* scope. The condition applies to the management domain in which the node where the condition is defined belongs. 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. 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.

––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 which resource class this condition will monitor. The **lsrsrcdef** command can be used to list the resource class names.

–s *"selection_string"*
Specifies a selection string that is applied to all of the *resource_class* attributes to determine which resources *event_expression* should monitor. The default is to monitor all resources within *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)

–U Unlocks a condition so it can be modified or removed. If a condition is locked, this is typically because it is essential for system software to work properly. For this reason, you should exercise caution before unlocking it. When unlocking a condition using the **-U** flag, no other operation can be performed by this command.

–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 an existing condition that is defined on *node_name*.

node_name Specifies the node in a 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.

Security

The user of the **chcondition** command needs write permission to the **IBM.Condition** resource class on the node where the condition is defined. 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 change the condition name from ″FileSystem space used″ to ″Watch FileSystem space″, run this command:

   ```
   chcondition -c "Watch FileSystem space"  "FileSystem space used"
   ```

2. To change a rearm expression and rearm description for a condition with the name ″tmp space used″, run this command:

   ```
   chcondition -E "PercentTotUsed < 80" \
   -D "Start monitoring tmp again after it is less than 80 percent full" \
   "tmp space used"
   ```

In these examples, which apply to management domains, the node where the command is run is on the management server.

1. To change the condition with the name ″FileSystem space used″ on the management server to check for space usage that is greater than 95%, run this command:

   ```
   chcondition -e "PercentTotUsed > 95" "FileSystem space used"
   ```

2. To change the condition with the name ″NodeB FileSystem space used″ on **NodeB** to check for space usage that is greater than 95%, run this command:

   ```
   chcondition -e "PercentTotUsed > 95" \
   "NodeB FileSystem space used":NodeB
   ```

This example applies to a peer domain:

1. To change the condition defined on **NodeA** with the name ″FileSystem space used″ to check for space usage that is greater than 95%, run this command:

   ```
   chcondition -e "PercentTotUsed > 95" \
   "FileSystem space used":NodeA
   ```

Location

/usr/sbin/rsct/bin/chcondition

Related Information

Books:

- *CSM: Administration Guide,* for information about node groups
- *CSM: Command and Technical Reference,* for information about the **nodegrp** command
- *RSCT: Administration Guide,* for information about RMC operations and about how to use expressions and selection strings

Commands: **lscondition**, **lscondresp**, **mkcondition**, **nodegrp**, **rmcondition**

Information Files: **rmccli**

---

chcons Command

Purpose

Redirects the system console to a specified device or file to be effective on the next startup of the system.

Syntax

chcons [**-a login** { **=disable** | **=enable**}] [**-a console_logname=***file*] [**-a console_logsize=***size*] [**-a console_logverb=***number*] [**-a console_tagverb=***number*] *PathName*

Description

The **chcons** command changes the system console effective on the next system startup. The current operation of the system console is not affected.

The *PathName* parameter must be a fully qualified path name to a device or file that is to become the system console.

If the *PathName* parameter specifies a file that does not exist, the **chcons** command creates the file at the next system startup. If the file does exist, the **chcons** command sends any console message output to the file. For a regular file, the system does not start the login program.

If the console path name is a character device, the system starts the login program on the device. Login is enabled on the console at all run levels. If no login is desired, use the **-a login=disable** flag.

CAUTION: If the console is the only login terminal on the system, you cannot log in at the next start of the system using the **-a login=disable** flag.

Additional Information

The **chcons** command saves the specified information into the database to be used on the next start-up of the system with the console configuration method. This method checks the specified device path name to determine if it is a character special file. If it is not, or does not exist, the device path name is assumed to be a file, and the console is set accordingly. If the device path name is a character special file, the console configuration method uses the base name as a logical name and attempts to look up the device name in the device database. If the device is found and available, the console is set to the device.

If the device is not found or is found but not available, a console finder routine is run that displays a prompt requesting that a new system console device be selected. By default, the tty on the S1 port and all graphics displays will display the prompt. The **/etc/consdef** file must be modified to display the prompt on S2 or other ports.

For a device, an entry in the **inittab** file with the console identifier is set to the respawn action to allow a login on the console if the console login was specified as the **enable** parameter. This causes a login to be available at all run levels. If the console login was specified with the **disable** parameter or if a file is designated as the console, the console entry in the **inittab** file is set to the OFF action, and login is disabled on the console for all run levels.

Flags

-a login= [**disable** | **enable**]	Enables or disables the login on the console for all run levels at the next start-up of the system.
-a console_logname=*file*	Specifies the full path name to use for the console output log file.
-a console_logsize=*size*	Specifies the size, in bytes, of the console output log file.
-a console_logverb=*number*	Specifies the verbosity level for console output logging. Zero disables logging; 1 through 9 enable logging.
-a console_tagverb=*number*	Specifies the verbosity level for console output tagging. Zero disables tagging, 1 through 9 enable tagging.

Examples

1. To change the system console to a file called **console.out** in the **/tmp** directory, enter:
   ```
   chcons /tmp/console.out
   ```
2. To change the system console to a terminal with the `tty3` logical name, enter:
   ```
   chcons /dev/tty3
   ```
3. To change the system console to the terminal associated with the `/dev/tty3` device and ensure a login at the console, enter:
   ```
   chcons -a login=enable /dev/tty3
   ```
4. To change the system console to a terminal with the `tty0` logical name and disable login at the console, enter:
   ```
   chcons -a login=disable /dev/tty0
   ```
5. To change the console to the default physical LFT display, enter:
   ```
   chcons /dev/lft0
   ```

Files

/dev/console	Specifies the special file for system console access.
/etc/consdef	Enables non-default terminal to be selected as the console device.
/usr/sbin/chcons	Specifies the command file.

Related Information

The **init** command, **lscons** command, **swcons** command.

The **inittab** file, **consdef** file.

The **console** special file.

---

chcore Command

Purpose

Changes the corefile settings.

Syntax

chcore [**-R** *registry*] [**-c** {on|off|default}] [**-p** {on|off|default}] [**-l** {*path*| default] [**-n** {on|off|default}] [*username* | **-d**]

Description

The **chcore** command is the user interface to change the core settings. It has the following usage:

```
chcore [-R registry] options [username|-d]
```

where,

options is at least one (and possibly more) of the following:

```
-c {on|off|default}
```

setting for core compression

```
-p {on|off|default}
```

setting for core location

```
-l path
```

specify directory to use

```
-n {on|off|default}
```

setting for core naming

If **-d** is specified, **chcore** will change the default setting for the system. The **-d** option is mutually exclusive with a specified *username* and with any specification of a *registry*. If neither **-d** nor a *username* is supplied, **chcore** will change the setting for the current user. Both the **-d** option and the ability to change settings for another user (other than the current user) are privileged operations, and may only be run by root or another user with system authority. Any changes made will not take effect until the next login session.

To change attributes an alternate Identification and Authentication (I&A) mechanism, the **-R** flag can be used to specify the I&A load module. If the **-R** flag is not specified, the **chcore** command uses the default attributes. Load modules are defined in the **/usr/lib/security/methods.cfg** file.

For more information on core naming concepts, refer to the core File Format in *AIX 5L Version 5.3 Files Reference*.

Note: The core settings changed by the **chcore** command are persistent across reboots of the system.

Flags

-c {on|off|default}	Setting for core compression.
-d	Changes the default setting for the system.
-l *path*	Directory path for stored corefiles.
-n {on|off|default}	Setting for core naming.
-p {on|off|default}	Setting for core location.
-R *registry*	Specifies the loadable I&A module.

Security

The command can only be run by root or another user with system authority.

RBAC Disclaimer

Examples

1. To make any process run by root dump compressed core files and restore the location of the core files to the system default, type:

   ```
   chcore -c on -p default root
   ```

   Note: If no default is specified, cores will dump in the current directory.
2. To enable a default core path for the system, type:

   ```
   chcore -p on -l /corefiles -d
   ```

   Note: All users who do not explicitly disable the core path with **chcore -p off** or override the core path with **chcore -l** will dump core files into the directory **/corefiles**. If a user does not have write permission to that directory, or the directory does not exist, no corefile will be generated.

Files

/usr/lib/security/methods.cfg	Contains load module definitions.
/etc/security/user	Contains extended user attributes.

Related Information

The **lscore** command.

The core File Format in *AIX 5L Version 5.3 Files Reference*.

---

chcosi Command

Purpose

Manages a Common Operating System Image (COSI).

Syntax

To install software:

chcosi -i -s *Source* [**-f***Fileset* | **-b** *installp_bundle* | **-F** *Fixes* | **-B** *fix_bundle*] [**-c**] [**-R**] [**-v**] *COSI*

To update software:

chcosi -u -s *Source* [**-f***Fileset* | **-b** *installp_bundle* | **-F** *Fixes* | **-B** *fix_bundle*] [**-c**] [**-R**] [**-v**] *COSI*

To reject software:

chcosi -j [**-f***Fileset* | **-b** *installp_bundle* | **-F** *Fixes* | **-B** *fix_bundle*] [**-R**] [**-v**] *COSI*

To remove software:

chcosi -r {**-f***Fileset* | **-b** *installp_bundle* | **-F** *Fixes* | **-B** *fix_bundle*} [**-R**] [**-v**] *COSI*

To remove software:

chcosi -u [**-f***Fileset* | **-b** *installp_bundle* | **-F** *Fixes* | **-B** *fix_bundle*] [**-R**] [**-v**] *COSI*

Description

The **chcosi** command manages a Common Operating System Image (COSI) created from the **mkcosi** command. Management tasks include installing, updating, rejecting, removing, and committing the software on the common image.

For installing and updating software on a common image, the required *Source* parameter specifies where the command gets installable images. The particular installable images are taken from the **-f**, **-b**, **-F**, **-B** flag and parameters. For the install, update, reject, and commit operations, if the **-f**, **-b**, **-F**, **-B** flags and parameters are not specified, the operation uses an assume-all value. So if the operation is an install or an update, all images from the source are used in the operation. If the operation is a reject or a commit, all software is committed or rejected from the common image. If the **-c** flag is specified with the install or update operation, the software is committed instead of applied. If a common image to be managed is being used by thin servers, a clone is created from the common image and the manage operation is performed on the clone image. The naming convention for the clone is the original common image name with the suffix _X{*count*}, where *count* is a number that is incremented every time a common image is cloned.

The **chcosi** command depends on the **bos.sysmgt.nim.master** fileset being present on the system. This command fails to execute if the **mkcosi** command is not run first to create a common image for managing.

Flags

-b *installp_bundle*	Specifies an **installp_bundle** NIM resource to be performed against the common image.
-B *fix_bundle*	Specifies a **fix_bundle** NIM resource to be performed against the common image.
-c	Specifies that the software to be installed or updated on the common image is put in the COMMIT state.
-f *Fileset*	Specifies a list of filesets to be performed against the common image.
-F *Fixes*	Specifies a list of fixes to be performed against the common image.
-i	Specifies the software to be installed.
-j	Specifies the software to be rejected.
-r	Specifies the software to be removed.
-R	Specifies the operation that is applied to requisite software.
-s *Source*	Specifies the source for common image management. The source can be an **lpp_source**, a device with installable media, a directory to installable images, or a remote location to installable images.
-u	Specifies the software to be updated or committed.
-v	Enables verbose debug output when the **chcosi** command runs.

Exit Status

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

Security

Access Control: You must have root authority to run the **chcosi** command.

Examples

1. To install **csm.core** software from a CD-ROM onto a common image named `cosi1`, enter:

   ```
   chcosi -i -s cd0 -f csm.core cosi1
   ```

   The **csm.core** fileset is installed on the `cosi1` common image, and the fileset is placed in an APPLIED state.

Location

/usr/sbin/chcosi

Files

/etc/niminfo	Contains variables used by NIM.

Related Information

The “cpcosi Command” on page 582, **lscosi** command, **mkcosi** command, **mkts** command, **nim** command, **nim_clients_setup** command, **nim_master_setup** command, **nimconfig** command, **rmcosi** command.

---

chdev Command

Purpose

Changes the characteristics of a device.

Syntax

chdev -l *Name* [**-a** *Attribute*=*Value ...*] [**-f** *File*] [**-h**] [**-p** *ParentName*] [**-P** | **-T**] [**-q**] [**-w** *ConnectionLocation*]

Description

The **chdev** command changes the characteristics of the specified device with the given device logical name that is specified with the **-l** *Name* flag. The device can be in the Defined, Stopped, or Available state. Some changes may not be allowed when the device is in the Available state. When changing the device characteristics, you can supply the flags either on the command line or in the specified **-f** *File* flag.

When neither the **-P** nor the **-T** flags are specified, the **chdev** command applies the changes to the device and updates the database to reflect the changes. If the **-P** flag is specified, only the database is updated to reflect the changes, and the device itself is left unchanged. This is useful in cases where a device cannot be changed because it is in use; in which case, the changes can be made to the database with the **-P** flag, and the changes will be applied to the device when the system is restarted. The **-T** flag is used to make a temporary change in the device without the change being reflected in the database. It is temporary

in that the device will revert to the characteristics described in the database when the system is restarted. Not all devices support the **-P** and **-T** flags. If a device that is in the Defined state, changes are only applied to the database.

Attention: To protect the Configuration database, the **chdev** command is not interruptible. Stopping this command before it is complete could result in a corrupted database.

You can use the Devices application in Web-based System Manager (wsm) or the System Management Interface Tool (SMIT) **smit chdev** fast path to change device characteristics.

Flags

-a *Attribute*=*Value*	Specifies the device attribute-value pairs used for changing specific attribute values. The *Attribute*=*Value* parameter can use one attribute value pair or multiple attribute value pairs for one **-a** flag. If you use an **-a** flag with multiple attribute value pairs, the list of pairs must be enclosed in quotes with spaces 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.
-f *File*	Reads the necessary flags from the named *File* parameter.
-h	Displays the command usage message.
-l *Name*	Specifies the device logical name in the Customized Devices object class whose characteristics are to be changed.
-P	Changes the device's characteristics permanently in the Customized Devices object class without actually changing the device. This is useful for devices that cannot be made unavailable and cannot be changed while in the available state. The change is made to the database, and the changes are applied to the device when the system is rebooted. This flag cannot be used with the **-T** flag. Not all devices support the **-P** flag.
-p *ParentName*	Specifies the new device logical name of the parent device in the Customized Devices object class. Use this flag only when changing the parent of the device. Not all devices support the **-p** flag.
-q	Suppresses the command output messages from standard output and standard error.
-T	Changes the characteristics of the device temporarily without changing the Customized Devices object class for the current start of the system. This flag cannot be used with the **-P** flag. Not all devices support the **-T** flag.
-w *ConnectionLocation*	Specifies the new connection location of the device on the parent. Use this flag only when changing the connection location of the device. Not all devices support the **-w** flag.

Security

Access Control

Only the root user and members of the security group should have execute (x) access to this command.

Auditing Events

Auditing Event	Information
DEV_Change	Parameters to the method the **cfgmgr** command calls.

RBAC Disclaimer

Examples

1. To change the retension instructions of the `rmt0` 4mm SCSI tape drive so that the drive does not move the tape to the beginning, then to the end, and then back to the beginning each time a tape is inserted or the drive is powered on, enter the following:

   ```
   chdev -l rmt0 -a ret=no
   ```

   The system displays a message similar to the following:

   ```
   rmt0 changed
   ```

2. To change one or more attributes of the `tok0` token-ring adapter to preset values as described in the `changattr` file, enter the following:

   ```
   chdev -l tok0 -f changattr
   ```

   The system displays a message similar to the following:

   ```
   tok0 changed
   ```

3. To change the SCSI ID of the available `scsi0` SCSI adapter that cannot be changed made unavailable due to available disk drives connected to it, enter the following:

   ```
   chdev  -l scsi0 -a id=6 -P
   ```

   The system displays a message similar to the following:

   ```
   scsi0 changed
   ```

   To apply the change to the adapter, shutdown and restart the system.

4. To move the defined `tty11` tty device to port 0 on the `sa5` serial adapter, enter the following:

   ```
   chdev  -l tty11  -p sa5  -w 0
   ```

   The system displays a message similar to the following:

   ```
   tty11 changed
   ```

5. To change the maximum number of processes allowed per user to 100, enter the following:

   ```
   chdev -l sys0 -a maxuproc=100
   ```

   The system displays a message similar to the following:

   ```
   sys0 changed
   ```

6. To delete the `alias4=10.3.4.3` Object Data Manager (ODM) entry from the `en2` standard Ethernet network interface, enter the following:

   ```
   chdev -l en2 -a delalias4=10.3.4.3
   ```

   The system displays a message similar to the following:

   ```
   en2 changed
   ```

7. To delete the `alias6=fe80::20b4:40ff:fe00:f016/64` ODM entry from the `en3` standard Ethernet network interface, enter the following:

   ```
   chdev -l en3 -a delalias6=fe80::20b4:40ff:fe00:f016/64
   ```

   The system displays a message similar to the following:

   ```
   en3 changed
   ```

Files

/usr/sbin/chdev Specifies the command file.

Related Information

The **lsattr** command, **lsconn** command, **lsdev** command, **lsparent** command, **mkdev** command, and **rmdev** command.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

The System Management Interface Tool (SMIT) in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* explains the structure, main menus, and tasks that are done with SMIT.

---

chdisp Command

Purpose

The **chdisp** command changes the default display being used by the Low Function Terminal Subsystem.

Syntax

chdisp { **-d** *DeviceName* | **-p** *DeviceName* }

Description

The **chdisp** command changes the display used by the low function terminal (LFT) subsystem.

To generate a list of available displays and their respective display identifiers and descriptions, use the **lsdisp** command. For an example of the listing displayed, see the **lsdisp** command example listing.

> **Note:** The **chdisp** command can be used only on an LFT.

You can use the Devices application in Web-based System Manager (wsm) to change device characteristics. You could also use the System Management Interface Tool (SMIT) **smit chdisp** fast path to run this command for certain devices.

Flags

-d *DeviceName*	Changes the default display currently being used by the LFT. This change is temporary resulting in the default display reverting back to the original display when the system is rebooted.
-p *DeviceName*	Changes the default display to the specified display at the next reboot. This stays in effect until the user changes the default display again. The user must have superuser access to use this option.

Examples

1. To temporarily change the default display to a display with a device name `ppr0`, enter:

   ```
   chdisp  -d ppr0
   ```

2. To permanently change the default display beginning with the next reboot to a display with the device name `gda1`, enter:

   ```
   chdisp  -p gda1
   ```

Files

/bin/chdisp	Contains the **chdisp** command.

Related Information

The **lsdisp** command.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

LFT Subsystem Component Structure Overview in *AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts*.

---

checkeq or checkmm Command

Purpose

Checks documents formatted with memorandum macros.

Syntax

{ **checkeq** | **checkmm** } [*File*...]

Description

The **checkeq** command is used to check for syntax errors in the specified files (*File*) that have been prepared for the **neqn** or **eqn** command. The **checkeq** command reports missing or unbalanced delimiters and the **.EQ** and **.EN** macro pair.

The **checkeq** command is functionally equivalent to the **checkmm** command.

The **checkmm** (check memorandum macros) command is used to check for syntax errors in files that have been prepared for the **mm** command or **mmt** command. For example, the **checkmm** command checks that you have a **.DE** (display end) macro corresponding to every **.DS** (display start) macro. *File* specifies files to be checked by the **checkeq** or **checkmm** command.

The output for the **checkmm** command is the number of lines checked and a list of macros that are unfinished because of missing macros.

Related Information

The **eqn** command, **mm** command, **mmt** command, **mvt** command, **neqn** command, **tbl** command.

The **.DE** and **.DS** macros, **.EN** and **.EQ** macros, **mm** macro package.

---

checknr Command

Purpose

Checks **nroff** and **troff** files.

Syntax

checknr [**-a.***Macro1***.***Macro2 ...*] [**-c.***Command1***.***Command2* ...] [**-f**] [**-s**] [*File* ...]

Description

The **checknr** command checks a list of **nroff** or **troff** input files for certain kinds of errors involving mismatched opening and closing delimiters and unknown commands. If no files are specified, the **checknr** command checks standard input.

Delimiters checked are:

- Font changes using the **\f***Newfont* ... **\f**P.
- Size changes using the **\s***Newsize* ... **\s**0.
- Macros that come in open and close forms (such as the **.TS** and **.TE** macros) that must always come in pairs.

The **checknr** command can handle both the **ms** and **me** macro packages.

The **checknr** command is intended to be used on documents that are prepared with the **checknr** command in mind, much the same as the **lint** command. The **checknr** command requires a certain document writing style for the **\f** and **\s** commands, in that each **\f***Newfont* must be terminated with **\fP** and each **\s***Newsize* must be terminated with **\s0**. While it works to go directly into the next font or to explicitly specify the original font or point size, such a practice produces error messages from the **checknr** command.

File specifies **nroff** or **troff** input files for errors involving mismatched opening and closing delimiters and unknown commands. The default is standard input.

Flags

-a.*Macro1***.***Macro2*
: Adds pairs of macros to the list. This flag must be followed by groups of six characters, each group defining a pair of macros. The six characters are a period, *Macro1*, another period, and *Macro2*. For example, to define the pair, **.BS** and **.ES**, use **-a.BS.ES**.

  Note: There is no way to define a 1-character macro name using the **-a** flag.

-c.*Command1***.***Command2*
: Defines otherwise undefined commands that would get error messages from the **checknr** command.

-f
: Causes the **checknr** command to ignore **\f** font changes.

-s
: Causes the **checknr** command to ignore **\s** size changes.

Note: The **checknr** command does not correctly recognize certain reasonable constructs, such as conditionals.

Related Information

The **checkeq** command, **lint** command, **nroff** command, **troff** command.

The **me** macro package, **ms** macro package.

---

chfilt Command

Purpose

Changes a filter rule.

Syntax

chfilt -v 4|6 -n *fid* [**-a D**|**P**|**I**|**L**|**E**|**H**|**S**] [**-s** *s_addr*] [**-m** *s_mask*] [**-d** *d_addr*] [**-M** *d_mask*] [**-g Y**|**N**] [**-c** *protocol*] [**-o** *s_opr*] [**-p** *s_port*] [**-O** *d_opr*] [**-P** *d_port*] [**-r R**|**L**|**B**] [**-w I**|**O**|**B**] [**-l Y**|**N**] [**-f Y**|**N**|**O**|**H**] [**-t** *tid*] [**-i** *interface*][**-D** *description*] [**-e** *expiration_time*] [**-x** *quoted_pattern* | **-X** *pattern_filename* | **-C** *antivirus_filename*]

Description

Use the **chfilt** command to change the definition of a filter rule in the filter rule table. Auto-generated filter rules and manual filter rules can be changed by this command. If an auto-generated filter rule is modified by the **chfilt** command it will then become a manual filter rule. IPsec filter rules for this command can be configured using the **genfilt** command, IPsec smit (IP version 4 or IP version 6), or Web-based System Manager in the Virtual Private Network submenu.

Flags

-a *Action*
The following *Action* values are allowed:
- **D** (Deny) blocks traffic.
- **P** (Permit) allows traffic.
- **I** makes this an IF filter rule.
- **L** makes this an ELSE filter rule.
- **E** makes this an ENDIF filter rule.
- **H** makes this a SHUN_HOST filter rule.
- **S** makes this a SHUN_PORT filter rule.

-C *anitvirus_filename*
Specifies the antivirus file name. The **-C** flag understands some versions of ClamAV Virus Database (http://www.clamav.net).

-c *protocol*
Protocol. The valid values are: **udp**, **icmp**, **icmpv6**, **tcp**, **tcp/ack**, **ospf**, **ipip**, **esp**, **ah**, and **all**. Value **all** indicates that the filter rule will apply to all the protocols. The protocol can also be specified numerically (between 1 and 252).

-d *d_addr*
Destination address. It can be an IP address or a host name. If a host name is specified, the first IP address returned by the name server for that host will be used. This value along with the destination subnet mask will be compared against the destination address of the IP packets.

-D
Filter description. A short description text for the filter rule.

-e *expiration_time*
Specifies the amount of time the rule should remain active in minutes. The *expiration_time* does not remove the filter rule from the database. The *expiration_time* relates to the amount of time the filter rule is active while processing network traffic. If no *expiration_time* is specified, the live time of the filter rule is infinite. If the *expiration_time* is specified in conjunction with a SHUN_PORT (**-a S**) or SHUN_HOST (**-a H**) filter rule, then this is the amount of time the remote port or remote host is denied or shunned once the filter rule parameters are met. If this *expiration_time* is specified independent of a shun rule, this is the amount of time the filter rule will remain active after the filter rules are loaded into the kernel and start processing network traffic.

-f
Fragmentation control. This flag specifies that this rule will apply to either all packets (**Y**), fragment headers and unfragmented packets only (**H**), fragments and fragment headers only (**O**), or unfragmented packets only (**N**).

-g
Apply to source routing? Must be specified as **Y** (yes) or **N** (No). If **Y** is specified, this filter rule can apply to IP packets that use source routing.

-i *interface*
The name of IP interface(s) to which the filter rule applies. Examples are: **all**, **tr0**, **en0**, **lo0**, and **pp0**.

-l
Log control. Must be specified as **Y** (yes) or **N** (No). If specified as **Y**, packets that match this filter rule will be included in the filter log.

-M *d_mask*
Destination subnet mask. This will be applied to the Destination address(**-d** flag) when compared with the destination address of the IP packets.

-m *s_mask*
Source subnet mask. This will be applied to the Source address (**-s** flag) when compared with the source address of the IP packet.

-n *fid*
The ID of the filter rule you want to change. It must exist in the filter rule table and for IP version 4, it cannot be 1 (rule 1 is a system reserved rule and is unchangeable).

-O *d_opr*
Destination port or ICMP code operation. This is the operation that will be used in the comparison between the destination port/ICMP code of the packet with the destination port or ICMP code (**-P** flag). The valid values are: **lt**, **le**, **g**t, **ge**, **eq**, **neq**, and **any**. This value must be **any** when the **-c** flag is **ospf**.

-o *s_opr*	Source port or ICMP type operation. This is the operation that will be used in the comparison of the source port/ICMP type of the packet with the source port or ICMP type (**-p** flag) specified in this filter rule. The valid values are: **lt**, **le**, **gt**, **ge**, **eq**, **neq**, and **any**. The value must be **any** when the **-c** flag is **ospf**.
-P *d_port*	Destination port/ICMP code. This is the value/code that will be compared to the destination port (or ICMP code) of the IP packet.
-p *s_port*	Source port or ICMP type. This is the value/type that will be compared to the source port (or ICMP type) of the IP packet.
-r	Specifies whether the rule will apply to forwarded packets (**R**), packets destined or originated from the local host (**L**), or both (**B**).
-s *s_addr*	Specifies the source address. It can be an IP address or a host name. If a host name is specified, the first IP address returned by the name server for that host will be used. This value along with the source subnet mask will be compared against the source address of the IP packets.
-t *tid*	Specifies the ID of the tunnel related to this filter rule. All the packets that match this filter rule must go through the specified tunnel.
-v	Specifies the IP version of the target filter rule.
-w	Specifies whether the rule will apply to incoming packets (**I**), outgoing packets (**O**), or both (**B**).
-X *pattern_filename*	Specifies the pattern file name. If more than one patterns are associated with this filter rule, then a pattern file name must be used. The pattern file name must be in the format of one pattern per line. A pattern is an unquoted character string. This file is read once when the filter rules are activated. For more information, see the **mkfilt** command.
-x *quoted_pattern*	Specifies the quoted character string or pattern. The **-x** *pattern* flag is compared against network traffic.

Security

RBAC Disclaimer

---

chfn Command

Purpose

Changes a user's gecos information.

Syntax

chfn [**-R** *load_module*] [*Name*]

Description

The **chfn** command changes a user's gecos information. Gecos information is general information stored in the **/etc/passwd** file. This information is not used by the system. The type of information you store in this field is up to you. Some system administrators store information such as the user's full name, phone number, and office number.

The **chfn** command is interactive. After you enter the command, the system displays the current gecos information and prompts you to change it. To exit the **chfn** command without changing any information, press Enter.

You can use any printable characters in the gecos information string except a **:** (colon), which is an attribute delimiter.

By default, the **chfn** command changes the gecos information of the user who runs the command. You can also use this command to change the gecos information of other users. However, you must have execute permission for the **chuser** command to change the gecos information for another user.

For users that were created using an alternate Identification and Authentication mechanism (I&A) , 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.

Flag

-R	Specifies the loadable I&A module used to change the user's gecos information

Security

Access Control

All users should have execute (x) access to this command since the program enforces its own access policy. This command should be installed as a program in the trusted computing base (TCB). The command should be owned by the security group with the **setgid** (SGID) bit set.

Files Accessed

Mode	File
x	**/usr/bin/chuser**
rw	**/etc/passwd**

RBAC Disclaimer

Limitations

Changing a user's gecos information may not be supported by all loadable I&A modules. If the loadable I&A module does not change a user's gecos information, an error is reported.

Examples

1. If you are John Smith and want to change your gecos information, type:

   ```
   chfn
   ```

   The current gecos string appears, followed by a prompt that asks if a change should be made:

   ```
   current gecos:
        "John Smith;555-1746;room 74"
   change (y/n)? >
   ```

   To change the room number from `74` to `36`, type `y` to request a change and type the revised information when the `to? >` prompt appears:

   ```
   current gecos:
        "John Smith;555-1746;room 74"
   change (y/n)? > y
   to? > John Smith;555-1746;room 36
   ```

2. If you are John Smith and want to view your gecos information but not change it, type:

   ```
   chfn
   ```

   The current gecos string appears, followed by a prompt that asks if a change should be made:

   ```
   current gecos:
        "John Smith;555-1746;room 74"
   change (y/n)? >
   ```

   If you decide not to change the information, type `n` after the `change (y/n)?` prompt or press the Enter key:

```
current gecos:
     "John Smith;555-1746;room 74"
change (y/n)? > n
```

This is your opportunity to indicate that the information should remain unchanged. If you enter `y`, you are committed to enter an information string or use the Enter key to set the string to null. Note that the function of the Enter key differs before and after a `y` character is entered.

3. If you have execute (x) permission for the **chuser** command and want to change the gecos information for the `johns` user, type:

   ```
   chfn johns
   ```

   The current gecos string and prompts appear as in Example 1.

4. To change the gecos for an LDAP I&A load module defined user `davis`, type:

   ```
   chfn -R LDAP davis
   ```

Files

/usr/bin/chfn	Specifies the path to the **chfn** command.
/usr/bin/chuser	Changes user information.
/etc/passwd	Contains basic user attributes.

Related Information

The **chgroup** command, **chgrpmem** command, **chuser** command, **lsgroup** command, **lsuser** command, **mkgroup** command, **mkuser** command, **passwd** command, **pwdadm** command, **rmgroup** command, **rmuser** command, **setgroups** command, **setsenv** command.

Security describes the identification and authentication of users, discretionary access control, the trusted computing base, and auditing.

---

chfont Command

Purpose

Changes the default font selected at boot time.

Syntax

chfont [*FontID*]

Description

The chfont command changes the font used by a display at system restart.

To see a list of available fonts with their respective font ids, font names, the glyph size and the font encoding, see the **lsfont** command. For an example of the listing displayed, see the **lsfont** command example listing.

You must have root authority to run this command.

> **Note:** This command can be used only on an LFT (Low Function Terminal).

You can use the Devices application in Web-based System Manager (wsm) to change device characteristics. You could also use the System Management Interface Tool (SMIT) **smit chfont** fast path to run this command.

Parameter

FontID The font id of the new font.

Examples

To change the font used by this display to the third font in the font palette, enter:

```
chfont 2
```

Files

/bin/chfont	Contains the **chfont** command.
/usr/lpp/fonts	Contains the font directory.

Related Information

The **lsfont** command, **mkfont** command.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

LFT Subsystem Component Structure Overview in *AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts*.

---

chfs Command

Purpose

Changes attributes of a file system.

Syntax

chfs [**-n** *NodeName*] [**-m** *NewMountPoint*] [**-u** *MountGroup*] [**-A** { **yes** | **no** }] [**-p** { **ro** | **rw** }] [**-t** { **yes** | **no** }] [**-a** *Attribute***=***Value*] [**-d** *Attribute*] *FileSystem*

Description

The **chfs** command changes the attributes of a file system. The new mount point, automatic mounts, permissions, and file system size can be set or changed. The *FileSystem* parameter specifies the name of the file system, expressed as a mount point.

Some file system attributes are set at the time the file system is created and cannot be changed. For the Journaled File System (JFS), such attributes include the fragment size, block size, number of bytes per i-node, compression, and the minimum file system size. For the Enhanced Journaled File System (JFS2), the block size cannot be changed.

The **chfs** command also accepts attributes that have no meaning to the file system. The attributes are saved in the **/etc/filesystems** file, but the file system does not act on the attributes. Additional attributes must be limited. The total size of a stanza in the **/etc/filesystems** file cannot exceed 512 bytes. If the size exceeds the limit, the stanza is no longer recognized.

The **chfs** command ignores any *Attribute=Value* pair that the command does not understand but add them to an appropriate stanza in the **/etc/filesystems**.

Example:

```
chfs -a abcd=1G /
```

This will set the new **abcd** attribute to the value of **+1G** in the root stanza in **/etc/filesystems**.

You can use the File Systems application in Web-based System Manager to change file system characteristics. You could also use the System Management Interface Tool (SMIT) **smit chfs** fast path to run this command.

Flags

-A Specifies the attributes for auto-mount.

- **yes** File system is automatically mounted at system restart.
- **no** File system is not mounted at system restart.

-d *Attribute* Deletes the specified attribute from the **/etc/filesystems** file for the specified file system.

-m *NewMountPoint* Specifies a new mount point for the specified file system.

-n *NodeName* Specifies a node name for the specified file system. The node name attribute in the **/etc/filesystems** file is updated with the new name. The node name attribute is specific to certain remote virtual file system types, such as the NFS (Network File System) virtual file system type.

-p Sets the permissions for the file system.

- **ro** Specifies read-only permissions.
- **rw** Specifies read-write permissions.

-t Sets the accounting attribute for the specified file system.

- **yes** File system accounting is to be processed by the accounting subsystem.
- **no** File system accounting is not to be processed by the accounting subsystem; this is the default.

-u *MountGroup* Specifies the mount group. Mount groups are used to group related mounts, so that they can be mounted as one instead of mounting each individually. For example, when performing certain tests, if several scratch file systems always need to be mounted together, they can each be placed in the `test` mount group. They can then all be mounted with a single command, such as the **mount -t** `test` command.

-a *Attribute=Value* Specifies the *Attribute=Value* pairs dependent on virtual file system type. To specify more than one Attribute=Value pair, provide multiple **-a** *Attribute=Value* parameters.

The following attribute or value pairs are specific to the Journaled File System (JFS):

-a *size=NewSize*

Specifies the size of the Journaled File System. Size can be specified in units of 512-byte blocks, Megabytes or Gigabytes. If Value has the M suffix, it is interpreted to be in Megabytes. If Value has a G suffix, it is interpreted to be in Gigabytes. If Value begins with a +, it is interpreted as a request to increase the file system size by the specified amount. If the specified size is not evenly divisible by the physical partition size, it is rounded up to the closest number that is evenly divisible.

The volume group in which the file system resides defines a maximum logical volume size and also limits the file system size.

The maximum size of a JFS file system is a function of its fragment size and the **nbpi** value. These values yield the following size restrictions:

NBPI	Minimum AG Size	Fragment Size	Maximum Size (GB)
512	8	512, 1024, 2048, 4096	8
1024	8	512, 1024, 2048, 4096	16
2048	8	512, 1024, 2048, 4096	32
4096	8	512, 1024, 2048, 4096	64
8192	8	512, 1024, 2048, 4096	128
16384	8	1024, 2048, 4096	256
32768	16	2048, 4096	512
65536	32	4096	1024
131072	64	4096	1024

-a *log=LVName*

Specifies the full path name of the filesystem logging logical volume name of the existing log to be used. The log device for this filesystem must reside on the same volume group as the filesystem.

-a *splitcopy=NewMountPointName*

Splits off a mirrored copy of the file system and mounts it read-only at the new mount point. This provides a copy of the file system with consistent JFS meta-data that can be used for backup purposes. User data integrity is not guaranteed, so it is recommended that file system activity be minimal while this action is taking place. Only one copy may be designated as an online split mirror copy.

-a *copy=Copy#*

Specifies which mirror copy to split off when used in conjunction with the *splitcopy* attribute. The default copy is the second copy. Valid values are 1, 2, or 3.

The following attribute or value pairs are specific to the Enhanced Journaled File System (JFS2):

-a *Attribute=Value*

-a *size=NewSize*

Specifies the size of the Enhanced Journaled File System in 512-byte blocks, Megabytes or Gigabytes. If Value has the M suffix, it is interpreted to be in Megabytes. If Value has a G suffix, it is interpreted to be in Gigabytes. If Value begins with a +, it is interpreted as a request to increase the file system size by the specified amount. If Value begins with a -, it is interpreted as a request to reduce the file system size by the specified amount.

If the specified size does not begin with a + or -, but it is greater or smaller than the file system current size, it is also a request to increase or reduce the file system size.

If the file system has an **inlinelog**, the **inlinelog** size remains unchanged if the new size of this file system is the same as the current file system size. If the specified size is not evenly divisible by the physical partition size, it is rounded up to the closest number that is evenly divisible. If the file system is on a striped logical volume, the size of the new file system is rounded to the nearest multiple of the striping width multiplied by the physical partition size. The striping width is the number of hard disks that form the striped logical volume.

This attribute is required when creating a JFS2 file system unless the **-d** flag has been specified. The volume group in which the file system resides defines a maximum logical volume size and limits the file system size. The maximum size is determined by the file system block size:

```
fs block size (byte)        MAX fssize (TB)
========================================
512                         4
1024                        8
2048                        16
4096                        32
```

When a request to reduce the file system size is successful, the logical volume should be equal to or smaller than the original LV size depending on the requested filesystem size.

Both *size* and *logsize* attributes can be specified in one **chfs** request to resize the filesystem and its **inlinelog** sizes.

Notes:

1. JFS2 does not have nbpi or fragment size values to affect the resulting size of the file system.
2. You cannot shrink a filesystem if the requested size is less that a physical partition size. At least one physical partition size is asked to be reduced.
3. Shrinking a file system that has snapshots is not allowed.
4. During a shrink of the file system, writes to the file system are blocked.
5. During the period that the shrink or extend is running, the file system is not accessible. Large file systems with inline logs might not be usable for as long as several minutes. The inline log must be completely reformatted.
6. When the new file system size is specified, but its **inlinelog** size is NOT specified, the new *logsize* will be adjusted (extended/shrunk) proportionally, based on the specified extended/shrunk file system size. The log size increase or reduction should not be more than 40% of the file system size increase or reduction.
7. When a new file system size is not specified and there is an **inlinelog**, if a new *logsize* is specified, the file system size might be changed to include the new log size.
8. The freed space reported by the **df** command is not necessary the space that can be truncated by a **shrinkFS** request due to filesystem fragmentation. A fragmented filesystem may not be shrunk if it does not have enough free space for an object to be moved out of the region to be truncated, and **shrinkFS** does not perform filesystem defragmentation. In this case, the **chfs** command should fail with the returned code 28 (ENOSPC).

-a *[log | logname]=LVName*

Specifies the full path name of the filesystem logging logical volume name of the existing log to be used. The log device for this filesystem must reside on the same volume group as the filesystem. Keyword INLINE can be used to specify that the log is in the logical volume with the JFS2 file system. The file system must have been created with an INLINE log to use this option. This option updates the /etc/filesystems file so that if the name of the logical volume containing the file system changes the log will be recognized.

Note: For a file system using OUTLINE log, this option can be used to change the outline log from one logical volume to another logical volume as long as the logical volume is properly formatted and the type of the logical volume is jfs2log. If a file systems is mounted at the time **chfs** is called to change the outline log, the **/etc/filesystems** will show the change, but the actual log will not be changed until the next mount for the file system (which follows a **umount** operation or a system crash and recovery). For a file system using INLINE log, this option does not support switching logs between INLINE and OUTLINE log. Currently, to switch from inlinelog to outlinelog (or vise versa), the file system has to be removed and recreated.

In release AIX 5L and AIX 5.1, if the file system is using inlinelog, the log entry is the same as the file system in /etc/filesystems:

```
EX:  /j2.1:
dev             = /dev/fslv00
vfs             = jfs2
log             = /dev/fslv00
mount           = false
account         = false
```

But, from AIX 5.2 and later releases, if the file system is using inlinelog, the log entry is the keyword INLINE in /etc/filesystems:

```
EX:  /j2.23:
dev             = /dev/fslv04
vfs             = jfs2
log             = INLINE
mount           = false
options         = rw
account         = false
```

If the file system was created at AIX 5L or AIX 5.1, and later upgraded to AIX 5.2 or later releases, then **chfs** can be used to alter the inlinelog name in **/etc/filesystems**.

-a options=*mountOptions*

Specifies which **mount** option is passed into the **chfs** command. For a list of the valid options, refer to the **mount** command.

-a *logsize=LogSize*

Specifies the size for an INLINE log in MBytes. The input size must be a positive value. If the inline log size is greater than or equal to 1, the input size must be an integer. If the input is floating point value of less than 1 and greater than or equal to 0, the input size is ignored and the default inline log size is taken. If value begins with a + (plus sign), it is interpreted as a request to increase the INLINE log size by the specified amount. If value begins with a - (minus sign), it is interpreted as a request to reduce the INLINE log size by the specified amount.

The input is ignored if an INLINE log not being used. The INLINE log size cannot be greater than 10% of the size of the file system and it cannot be greater than 2047 MBytes.

-a *ea=v2*

Converts the JFS2 file system extended attribute (ea) format. A JFS2 file system using the v1 format can be converted to one using v2 format. After it is converted the file system cannot be converted back to v1. 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. 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.

-a freeze={ timeout | 0 | off}

Specifies that the file system must be frozen or thawed, depending on the value of **timeout**. The act of freezing a file system produces a nearly consistent on-disk image of the file system, and writes all dirty file system metadata and user data to the disk. In its frozen state, the file system is read-only, and anything that attempts to modify the file system or its contents must wait for the freeze to end. The value of **timeout** must be either `0`, `off`, or a positive number. If a positive number is specified, the file system is frozen for a maximum of **timeout** seconds. If **timeout** is `0` or `off`, the file system will be thawed, and modifications can proceed.

Attention: Freezing base file systems (**/**, **/usr**, **/var**, **/tmp**) can result in unexpected behavior.

-a refreeze={timeout}

Specifies that the timeout for a frozen file system be reset. The **timeout** is reset to the value specified. The file system must still be frozen (using the **-a freeze** option or the **fscntl** interface).

-a vix={yes|no}

Specifies whether the file system can allocate inode extents smaller than the default of 16K if there are no contiguous 16K extents free in the file system. After a file system is enabled for small free extents, it cannot be accessed on earlier versions of AIX and the marking cannot be removed.

yes File system can allocate variable length inode extents.

no File system must use default size of 16 KB for inode extents. This has no effect if the file system already contains variable length inode extents.

-a efs=yes

Converts a file system to an Encrypted File System (EFS).

The **chfs** command changes an existing file system into an EFS file system. When the file system is EFS enabled, the **ea** attribute is automatically converted to store scalable extended attributes (**v2**).

Restriction: The **chfs** commands prevents conversion of the following file systems (mount points) to EFS because the security infrastructures (kernel extensions, libraries and so on) are not available during boot:

- **/**
- **/usr**
- **/var**
- **/opt**

-a managed={yes | no}

Enables Data Management Application Programming Interface (DMAPI) on a JFS2 file system.

Security

Access Control

Only the root user or a member of the **system** group can run this command.

RBAC Disclaimer

Examples

1. To change the file system size of the `/test` Journaled File System, enter:

   ```
   chfs  -a size=24576 /test
   ```

   This command changes the size of the `/test` Journaled File System to 24576 512-byte blocks, or 12MB (provided it was previously no larger than this).
2. To increase the size of the `/test` Journaled File System, enter:

   ```
   chfs  -a size=+8192 /test
   ```

   This command increases the size of the `/test` Journaled File System by 8192 512-byte blocks, or 4 MB.
3. To convert a JFS2 file system to a version which can support NFS4 ACLs, type:

   ```
   chfs -a ea=v2 /test
   ```
4. To change the mount point of a file system, enter:

   ```
   chfs  -m /test2 /test
   ```

   This command changes the mount point of a file system from `/test` to `/test2`.
5. To delete the accounting attribute from a file system, enter:

   ```
   chfs -d account /home
   ```

   This command removes the accounting attribute from the **/home** file system. The accounting attribute is deleted from the `/home:` stanza of the **/etc/filesystems** file.
6. To split off a copy of a mirrored file system and mount it read-only for use as an online backup, enter:

   ```
   chfs -a splitcopy=/backup -a copy=2 /testfs
   ```

   This mount a read-only copy of **/testfs** at **/backup**.
7. To change the file system size of the **/test** Journaled File System, enter:

   ```
   chfs -a size=64M /test
   ```

   This command changes the size of the **/test** Journaled File System to 64MB (provided it was previously no larger than this).
8. To reduce the size of the **/test** JFS2 file system, enter:

   ```
   chfs  -a size=-16M /test
   ```

   This command reduces the size of the **/test** JFS2 file system by 16MB.
9. To freeze a file system, enter:

   ```
   chfs  -a freeze=60 /adl
   ```

   This command freezes the **/adl** file system for a maximum of 60 seconds.
10. To thaw a file system, enter:

    ```
    chfs  -a freeze=off /zml
    ```

    This command thaws the **/zml** file system.

File

/etc/filesystems Lists the known file systems and defines their characteristics.

Related Information

The **crfs** command, **mkfs** command, **mklv** command.

The System accounting in *Operating system and device management* explains the file system accounting subsystem.

The File systems in *Operating system and device management* explains file system types, management, structure, and maintenance.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

The System Management Interface Tool (SMIT) in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* explains the structure, main menus, and tasks that are done with SMIT.

---

chgif Method

Purpose

Reconfigures an instance of a network interface.

Syntax

chgif [**-d** | **-T**] **-l** *InterfaceInstance* **-a** *"Attribute=Value ..."*

Description

The **chgif** method first modifies the database and then reconfigures the specified network interface instance (*InterfaceInstance*) by issuing a call to the **ifconfig** command. Only one interface can be changed per command invocation, and at least one attribute must be specified. This method is not normally used on the command line. Rather, it is called by high-level commands.

Note: The **chgif** method is a programming tool and should not be executed from the command line.

Flags

-a*"Attribute=Value ..."* Specifies pairs of attributes and values that configure the Interface instance. The *AttributeValue* pairs must be surrounded by quotes.

Valid attribute values are as follows:

netaddr
Specifies the Internet address of the network interface.

state (up/down)
Marks the interface as up or down.

trailers (on/off)
Turns the trailer link-level encapsulation on or off.

arp (on/off)
Enables or disables the use of the Address Resolution Protocol.

allcast (on/off)
Specifies whether to broadcast packets to all token-ring networks or just the local token-ring network. This attribute applies only to token-ring networks.

hwloop (on/off)
Enables or disables hardware loopback mode.

netmask
Specifies the network mask in dotted-decimal format.

security *SecurityLevelKeyword*
(**inet** only) Specifies the security level associated with the interface. The value of the *SecurityLevelKeyword* variable can be one of the following:

- **none**
- **unclassified**
- **confidential**
- **secret**
- **top_secret**

When the level of security is defined as **none** or **unclassified**, no IP Option header is added to the IP header.

authority *AuthorityLevelKeyword*
(**inet** only) Specifies the security authority level associated with the interface. The value of the *AuthorityLevelKeyword* variable can be one or more of the following:

genser Defense Communications Agency

siop Department of Defense Organization of the Joint Chiefs of Staff

dsccs-spintcom
Defense Intelligence Agency

dsccs-criticom
National Security Agency

When more than one level of authority is specified, the values are separated by commas without embedded spaces.

mtu Maximum IP packet size for this system.

broadcast
Specifies the address to use for representing broadcasts to networks.

dest Specifies the destination address on a point-to-point link.

-d Specifies that changes are made only in the configuration database. Changes take effect at the next system restart.

-l *InterfaceInstance* Specifies the instance of the network interface to be reconfigured.

-T Makes a temporary change in the device without the change being reflected in the database. It is temporary in that the device reverts to the characteristics described in the database when the system is restarted.

Related Information

The **chdev** command, **ifconfig** command.

The **chginet** method.

The **odm_run_method** subroutine.

TCP/IP protocols in *Networks and communication management*, TCP/IP addressing, TCP/IP network interfaces in *Networks and communication management*.

Object Data Manager (ODM) Overview for Programmers in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

Writing a Device Method in *AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts*.

---

chginet Method

Purpose

Reconfigures the Internet instance.

Syntax

chginet [**-d**] [**-a**″*Attribute*=*Value* ...″]

Description

The **chginet** method reconfigures the Internet instance, and can also change the *HostName* variable and any static routes that are defined. The **chginet** method calls the **hostname** command to change the host name. The **chginet** method also calls the **route** command to change any static routes. The **chdev** command calls method.

Note: The **chginet** method is a programming tool and should not be entered from the command line.

Flags

-a″*Attribute*=*Value* ...″
Specifies the customized attributes of the Internet instance. The following are valid attributes:

hostname
Specifies the name of the host.

gateway
Specifies the default gateway.

route
Specifies the route. The format of the *Value* variable of the **route** attribute is: route = *destination*, *gateway*, [*metric*].

delroute
Specifies the route to delete. The format of the value is: delroute = *type*, -netmask, *netmask*, *destination*, *gateway*, [*metric*].

The value of the *type* parameter can be **net** or **host**.

-d
Specifies that changes are made only in the configuration database. Changes take effect with the next IPL.

Examples

To change an Internet instance and specify a route, enter a method in the following format:

```
chginet  -a″route=192.9.200.0,bcroom″
```

This example specifies a new route. The new route is being set to network `192.9.200.0`, the `bcroom` gateway.

Related Information

The **chdev** command, **hostname** command, **mkdev** command, **route** command.

The **odm_run_method** subroutine.

TCP/IP network interfaces in *Networks and communication management*.

Object Data Manager (ODM) Overview for Programmers in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

Writing a Device Method in *AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts*.

---

chgroup Command

Purpose

Changes attributes for groups.

Syntax

chgroup [**-R** *load_module*] *Attribute*=*Value ... Group*

Description

Attention: Do not use the **chgroup** command if you have a Network Information Service (NIS) database installed on your system, as this could cause serious system database inconsistencies.

The **chgroup** command changes attributes for the group specified by the *Group* parameter. The group name must already exist. To change an attribute, specify the attribute name and the value you want to change it to in the *Attribute*=*Value* parameter.

To change the attributes for a group that was created with an alternate Identification and Authentication (I&A) mechanism, the **-R** flag can be used to specify the I&A loadable module. Load modules are defined in the **/usr/lib/security/methods.cfg** file.

You can use the Users application in Web-based System Manager (wsm) to change user characteristics. You could also use the System Management Interface Tool (SMIT) **smit chgroup** fast path to run this command.

Changing the ID for an account can compromise system security and as a result one should not do so. However, when the ID is changed using the **chgroup** command, ID collision checking is also controlled by the **dist_uniqid** attribute in the usw stanza of the **/etc/security/login.cfg** file. The behavior of ID collision control is the same as that described for the **mkgroup** command.

Restrictions on Changing Groups

To ensure the security of group information, there are restrictions on using the **chgroup** command. Only the root user or users with UserAdmin or aix.security.group.change authorization can use the **chgroup** command to change any group. These changes include:

- Make a group an administrative group by setting the **admin** attribute to true.
- Change any attributes of an administrative group.
- Add users to an administrative group's administrators list.

An administrative group is a group with the **admin** attribute set to true. Members of the **security** group can change the attributes of nonadministrative groups including adding users to the list of administrators.

Flag

-R Specifies the loadable I&A module used to change user's attributes.

Attributes

You change attributes by specifying an *Attribute*=*Value* parameter. If you have the proper authority you can set the following group attributes:

adms Defines the users who can perform administrative tasks for the group, such as setting the members and administrators of the group. This attribute is ignored if **admin = true**, since only the root user can alter a group defined as administrative. The *Value* parameter is a list of comma-separated user login names. If you do not specify a *Value* parameter, all the administrators are removed.

admin Defines the administrative status of the group. You can specify the following values:

- **true** Defines the group as administrative. Only the root user can change the attributes of groups defined as administrative.
- **false** Defines a standard group. The attributes of these groups can be changed by the root user or a member of the security group. This is the default value.

id The group ID. The *Value* parameter is a unique integer string. Changing this attribute compromises system security and, for this reason, you should not change this attribute.

projects Defines the list of projects to which the user's processes can be assigned. The value is a list of comma-separated project names and is evaluated from left to right. The project name should be a valid project name as defined in the system. If an invalid project name is found on the list, it will be reported as an error.

users A list of one or more users in the form: *User1*,*User2*,...,*Usern*. Separate group member names with commas. Each user must be defined in the database configuration files. You cannot remove users from their primary group.

efs_initialks_mode Specifies the initial mode of the group keystore. You can specify the following values:

- **admin** Root or other security privileged system users can open the group keystore using the admin key.
- **guard** Root users cannot open the group keystore using the admin key.

The default value is **admin**.

The attribute specifies the initial mode of the group keystore. You can use the attribute with the **mkgroup** command. After the keystore has been created, changing the attribute value with the **chuser**, **chgroup**, or **chsec** command, or manual editing does not change the mode of the keystore unless the keystore is deleted and a new one is created. To change the keystore mode, use the **efskeymgr** command.

Restriction: The attribute is valid only when the system is EFS-enabled.

efs_keystore_algo Specifies the algorithm that is used to generate the private key of the group during the keystore creation. You can specify the following values:

- **RSA_1024**
- **RSA_2048**
- **RSA_4096**

The default value is **RSA_1024**.

You can use the attribute with the **mkgroup** command. After the keystore has been created, changing the value of this attribute with the **chuser**, **chgroup**, or **chsec** command, or manual editing does not regenerate the private key unless the keystore is deleted and a new one is created. To change the algorithm for the keys, use the **efskeymgr** command.

Restriction: The attribute is valid only when the system is EFS-enabled.

efs_keystore_access Specifies the database type of the group keystore. You can specify the following values:

file Creates the **/var/efs/groups/***grpname***/keystore** keystore file associated with the group.

none The keystore is not created. All other keystore attributes have no effect.

The default value is **file**.

Restriction: The attribute is valid only when the system is EFS-enabled.

The **adms** and **admin** attributes are set in the **/etc/security/group** file. The remaining attributes are set in the **/etc/group** file. If any of the attributes you specify with the **chgroup** command are invalid, the command makes no changes at all.

Exit Status

This command returns the following exit values:

0	The command runs successfully and all requested changes are made.
>0	An error occurred. The printed error message gives further details about the type of failure.

Security

Access Control

This command should grant execute (x) access only to the root user and 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 Events

Event	Information
GROUP_Change	group, attributes

Files Accessed

Mode	File
rw	**/etc/group**
rw	**/etc/security/group**
r	**/etc/passwd**

RBAC Disclaimer

Limitations

Changing a group's attributes may not be supported by all loadable I&A modules. If the loadable I&A module does not support changing a group's attributes, an error is reported.

Examples

1. To add `sam` and `carol` to the `finance` group, which currently only has `frank` as a member, type:

   ```
   chgroup users=sam,carol,frank  finance
   ```

2. To remove `frank` from the `finance` group, but retain `sam` and `carol`, and to remove the administrators of the `finance` group, type:

   ```
   chgroup users=sam,carol adms= finance
   ```

   In this example, two attribute values were changed. The name `frank` was omitted from the list of members, and the value for the `adms` attribute was left blank.

3. To change the LDAP I&A loadable module group user's attribute, type:

   ```
   chgroup -R LDAP users=sam,frank monsters
   ```

Files

/usr/bin/chgroup	Specifies the path to the **chgroup** command.
/etc/group	Contains the basic attributes of groups.
/etc/security/group	Contains the extended attributes of groups.
/etc/passwd	Contains the basic attributes of users.

Related Information

The **chfn** command, **chgrpmem** command, **chsh** command, **chuser** command, **lsgroup** command, **lsuser** command, **mkgroup** command, **mkuser** command, **passwd** command, **pwdadm** command, **rmgroup** command, **rmuser** command, **setgroups** command, **setsenv** command.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to *Security*.

---

chgrp Command

Purpose

Changes the group ownership of a file or directory.

Syntax

chgrp [**-f**] [**-h**] [**-R**] *Group* { *File* ... | *Directory* ... }

chgrp -R [**-f**] [**-H** | **-L** | **-P**] *Group* { *File*... | *Directory*... }

Description

The **chgrp** command changes the group of the file or directory specified by the *File* or *Directory* parameter to the group specified by the *Group* parameter. The value of the *Group* parameter can be a group name from the group database or a numeric group ID. When a symbolic link is encountered and you have not specified the **-h** or **-P** flags, the **chgrp** command changes the group ownership of the file or directory pointed to by the link and not the group ownership of the link itself.

Although the **-H**, **-L** and **-P** flags are mutually exclusive, specifying more than one is not considered an error. The last flag specified determines the behavior that the command will exhibit.

If you specify the **-h** flag, the **chgrp** command has the opposite effect and changes the group ownership of the link itself and not that of the file or directory pointed to by the link.

If you specify both the **-h** flag and the **-R** flag, the **chgrp** command descends the specified directories recursively, and when a symbolic link is encountered, the group ownership of the link itself is changed and not that of the file or directory pointed to by the link.

Flags

-f Suppresses all error messages except usage messages.

-h Changes the group ownership of an encountered symbolic link and not that of the file or directory pointed to by the symbolic link.

-H If the **-R** option is specified and a symbolic link referencing a file of type directory is specified on the command line, **chgrp** shall change the group of the directory referenced by the symbolic link and all files in the file hierarchy below it.

-L If the **-R** option is specified and a symbolic link referencing a file of type directory is specified on the command line or encountered during the traversal of a file hierarchy, **chgrp** shall change the group of the directory referenced by the symbolic link and all files in the file hierarchy below it.

-P If the **-R** option is specified and a symbolic link is specified on the command line or encountered during the traversal of a file hierarchy, **chgrp** shall change the group ID of the symbolic link if the system supports this operation. The **chgrp** utility shall not follow the symbolic link to any other part of the file hierarchy.

-R Descends directories recursively, setting the specified group ID for each file. When a symbolic link is encountered and the link points to a directory, the group ownership of that directory is changed but the directory is not further traversed. If the **-h**, **-H**, **-L** or **-P** flags are not also specified, when a symbolic link is encountered and the link points to a directory, the group ownership of that directory is changed but the directory is not traversed further.

Exit Status

This command returns the following exit values:

0 Successful completion.

>0 An error occurred.

Security

RBAC Disclaimer

Examples

1. To change the group ownership of the file or directory named `proposals` to `staff`:

   ```
   chgrp staff proposals
   ```

   The group access permissions for `proposals` now apply to the `staff` group.
2. To change the group ownership of the directory named `proposals`, and of all the files and subdirectories under it, to `staff`:

   ```
   chgrp -R staff proposals
   ```

   The group access permissions for `proposals` and for all the files and subdirectories under it now apply to the `staff` group.

Files

/usr/bin/chgrp	The **chgrp** command
/etc/group	File that identifies all known groups

Related Information

The **chown** command, **groups** command.

The **chown** subroutine, **fchown** subroutine.

Security describes system security.

File ownership and user groups in *Operating system and device management* introduces file ownership and permissions to access files and directories.

---

chgrpmem Command

Purpose

Changes the administrators or members of a group.

Syntax

chgrpmem [**-R** *load_module*] [{ **-a** | **-m** } { **+** | **-** | **=** } *User* ...] *Group*

Description

The **chgrpmem** command changes the administrators or members of the group specified by the *Group* parameter. Use this command to add, delete, or set a group's members or administrators list. You cannot remove users from their primary group. A user's primary group is maintained in the **/etc/passwd** file. If you specify only a group with the **chgrpmem** command, the command lists the group's members and administrators.

To change the administrators or members of a group that were created with an alternate Identification and Authentication (I&A) mechanism, the **-R** flag can be used to specify the I&A loadable module. Load modules are defined in the **/usr/lib/security/methods.cfg** file.

To add, delete, or set a user as a group administrator, specify the **-a** flag. Otherwise, to add, delete, or set a user as a group member, specify the **-m** flag. You must specify one of these flags and an operator to change a user's group membership. The operators do the following:

+	Adds the specified user.
-	Deletes the specified user.
=	Sets the list of administrators or members to the specified user.

You can specify more than one *User* parameter at a time. To do this, specify a comma-separated list of user names.

See the **chgroup** command for a list of restrictions that apply to changing group information.

Flags

-a	Changes a group's administrators list.
-m	Changes the group's members list.

-R Specifies the loadable I&A module used to change the administrators or members of a group.

Exit Status

This command returns the following exit values:

0	The command runs successfully and all requested changes are made.
>0	An error occurred. The printed error message gives further details about the type of failure.

Security

Access Control

All users should have execute (x) access to this command because the command itself enforces the access rights. This command should be installed as a program in the trusted computing base (TCB). The command should be owned by the **security** group with the **setgid** (SGID) bit set.

Files Accessed

Mode	File
x	**/usr/bin/chgroup**
r	**/etc/passwd**
r	**/etc/group**
rw	**/etc/security/group**

RBAC Disclaimer

Examples

1. To remove `jones` as an administrator of the `f612` group, enter:

   ```
   chgrpmem  -a - jones f612
   ```

2. To add members `davis` and `edwards` to group `f612`, enter:

   ```
   chgrpmem  -m + davis,edwards f612
   ```

3. To list members and administrators of group `staff`, enter:

   ```
   chgrpmem staff
   ```

4. To list members of the LDAP I&A loadable module group `monsters`, enter:

   ```
   chgrpmem -R LDAP monsters
   ```

Files

/usr/bin/chgrpmem	Specifies the path to the **chgrpmem** command.
/etc/passwd	Contains the basic attributes of users.
/etc/group	Contains the basic attributes of groups.
/etc/security/group	Contains the extended attributes of groups.

Related Information

The **chfn** command, **chgroup** command, **chsh** command, **chuser** command, **lsgroup** command, **lsuser** command, **mkgroup** command, **mkuser** command, **passwd** command, **pwdadm** command, **rmgroup** command, **rmuser** command, **setgroups** command, **setsenv** command.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to *Security*.

chhwkbd Command

Purpose

Changes keyboard attributes stored in the Object Data Manager (ODM) database.

Syntax

chhwkbd [**-d** *Delay*] [**-r** *Repetition*] [**-c** *ClickerVolume*] [**-a** *AlarmVolume*] [**-m** [″KR″ | ″JP″ | ″TW″]] [**-t** [″nonum″]]

Description

The **chhwkbd** command changes the following keyboard attributes stored in the ODM database:

- Repetition delay
- Repetition rate
- Clicker volume
- Alarm volume
- Korean, Japanese, and Chinese keyboard identification
- Numeric pad emulation enable/disable

Changes to the keyboard attributes take effect after system restart.

You can use the Devices application in Web-based System Manager (wsm) to change device characteristics. You could also use the System Management Interface Tool (SMIT) **smit chgkbd** fast path to run this command.

Flags

-a *AlarmVolume*
Sets the alarm volume to the specified value. Values for the *AlarmVolume* variable are defined below:

0	off
1	low
2	medium
3	high

-c *ClickerVolume*
Sets the clicker volume to the specified value. Values for the *ClickerVolume* variable are defined below:

0	off
1	low
2	medium
3	high

-d *Delay*
Sets the keyboard repetition delay to the specified value. The *Delay* variable can be 250, 500, 750, or 1000 msec. The default value is 500 msec.

-m [″KR″ | ″JP″ | ″TW″]
Provides extended keyboard identification for the following keyboards:

″**KR**″ Korean keyboard

″**JP**″ Japanese keyboard

″**TW**″ Chinese keyboard

Use the **-m** flag without specifying a value to remove extended keyboard identification.

> **Note:** This flag is valid only when an IBM RS/6000 106-key keyboard or an IBM PS/2 keyboard or equivalent keyboard is attached to the workstation.

The **-m** flag is set automatically when the locale is selected using SMIT.

-r *Repetition*
Sets the rate of repetition to the specified value. The *Repetition* variable can be an integer from 2 to 30 inclusive. The default value is 11 characters per second.

-t [″nonum″]
Enables or disables numeric pad emulation. To enable numeric pad emultaion, specify the ″nonum″ parameter. Use the **-t** flag without specifying a value to disable numeric pad emulation.

Notes:

1. This flag is valid only when an IBM PS/2 keyboard or equivalent keyboard is attached to the workstation.
2. ″nonum″ means no numeric keypad.

Examples

1. To change the keyboard repetition delay rate to 250 msec, enter:

```
chhwkbd  -d 250
```

2. To change the keyboard repetition rate to 30 characters per second, enter:

```
chhwkbd  -r 30
```

File

/usr/bin/chhwkbd Contains the **chhwkbd** command.

Related Information

Low Function Terminal (LFT) Subsystem Overview in *AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

chiscsi Command

Purpose

Changes iSCSI target data.

Syntax

chiscsi -l *AdapterName* **-g static -t** *TargetName* [**-n** *PortNumber* **-i** *IPaddress*] [**-p** *password*] [**-T** *NewTargetName*] [**-N** *NewPortNumber*] [**-I** *NewIPaddress*]

chiscsi -l *AdapterName* **-g auto -t** *TargetName* [**-p** *password*] [**-T** *NewTargetName*]

Description

The **chiscsi** command changes iSCSI target data in 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 2nd 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

-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 *NewIPaddress*	Specifies the new IP address of the iSCSI target when it is being changed.
-i *IPaddress*	Specifies the IP address of the iSCSI target.
-l *AdapterName*	Specifies the adapter name for the iSCSI TCP/IP Offload Engine (TOE) adapter that is attached to this iSCSI target. It can also specify the iSCSI protocol device for the iSCSI software solution device.
-N *NewPortNumber*	Specifies the new port number of the iSCSI target when it is being changed.
-n *NewPortNumber*	Specifies the port number on which the iSCSI target is accessed. The default port number is `3260`.
-p *password*	Specifies the new password for this iSCSI target.
-T *NewTargetName*	Specifies the new iSCSI target name when it is being changed.
-t *TargetName*	Specifies the iSCSI target name (for example, `iqn.sn9216.iscsi-hw1`).

Exit Status

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

Security

The **chiscsi** command is executable only by root.

Examples

1. To change the password of a statically configured iSCSI target to `my password`, enter:

   ```
   chiscsi -l ics0 -g static -t qn.mds9216.iscsi_hw -n 3260 -i 10.1.2.116 -p "my password"
   ```

2. To change the IP address of a statically configured iSCSI target to `10.1.3.141`, enter:

```
chiscsi -l ics0 -g static -t qn.mds9216.iscsi_hw -n 3260 -i 10.1.2.116 -I 10.1.3.141
```

Location

/usr/sbin/chiscsi

Files

src/bos/usr/sbin/iscsia	Contains the common source files from which the iSCSI commands are built.

Related Information

The lsiscsi command, mkiscsi command, rmiscsi command.

---

chitab Command

Purpose

Changes records in the **/etc/inittab** file.

Syntax

chitab { [*Identifier*] **:** [*RunLevel*] **:** [*Action*] **:** [*Command*] }

Description

The **chitab** command changes a record in the **/etc/inittab** file. The *Identifier***:***Run Level***:***Action***:***Command* parameter string is the new entry to the **/etc/inittab** file. You can search for a specific record by using fields in the *Identifier* portion of the parameter string. The command finds the specified *Identifier* and changes that record.

Note: The **chitab** command can not comment out an entry in the **/etc/inittab** file.

Parameters

The *Identifier***:***Run Level***:***Action***:***Command* parameter string specifies a record in the **/etc/inittab** file where the following parameters apply:

Action
A 20-character parameter that informs the **init** command how to process the *Command* parameter you specify. The **init** command recognizes the following actions:

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.

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.

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.

initdefault
Start the process identified in this record only when the **init** command is originally invoked. The **init** command uses this line to determine which run level to originally enter. It does this by taking the highest run level specified in the *RunLevel* field and using that as its initial state. If the *RunLevel* parameter is empty, this 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 **/etc/inittab** file, it requests an initial run level from the operator at initial program load (IPL) time.

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.

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.

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.

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.

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.

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.

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. If you are operating in a diskless environment, specifying the **wait** action causes your system to boot more quickly.

Command A 1024-character field specifying the shell command.

Identifier A 14-character parameter that uniquely identifies an object. The *Identifier* must be unique. If the *Identifier* is not unique, the command is unsuccessful. The *Identifier* cannot be changed; if you try to change it, the command is unsuccessful.

RunLevel A 20-character parameter defining the run levels in which the *Identifier* can be processed. Each process started by the **init** command can be assigned one or more run levels in which it can be started.

Security

RBAC Disclaimer

Examples

To change the run level of a record for `tty2`, enter:

```
chitab "tty002:23:respawn:/usr/sbin/getty /dev/tty"
```

The quotes are required when the record being added has spaces or tabs.

Files

/etc/inittab	Indicates which processes the **init** command starts.

Related Information

The **init** command, **lsitab** command, **mkitab** command, **rmitab** command.

---

chkbd Command

Purpose

Changes the software keyboard map to be loaded into the system at the next IPL (Initial Program Load).

Syntax

chkbd *KeyMapPathName*

Description

The **chkbd** command changes the default software keyboard map loaded at system IPL. The *KeyMapPathname* parameter provides the location of the software keymap file. This pathname can be absolute or simply the filename. If only the filename is specified then the command will look for it in the default directory **/usr/lib/nls/loc**.

> **Note:** This command can be used only on an LFT display.

For a list of all available keyboard maps, use the **lskbd** command.

You can use the Devices application in Web-based System Manager (wsm) to change device characteristics. You could also use the System Management Interface Tool (SMIT) **smit chkbd** fast path to run this command.

Parameter

KeyMapPathName	Provides the location of the software keymap file.

Files

/bin/chkbd	Contains the **chkbd** command.
/usr/lib/nls/loc	Contains the keyboard directory.

Related Information

Keyboard Technical Reference

Low Function Terminal (LFT) Subsystem Overview in *AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

chkey Command

Purpose
Changes your encrypting key.

Syntax
/usr/bin/chkey

Description
The **chkey** command prompts you for a password and uses it to encrypt a new encryption key. Once the key is encrypted, the **ypupdated** daemon updates the **/etc/publickey** file.

Related Information
The **keylogin** command, **newkey** command.

The **keyserv** daemon, **ypupdated** daemon.

The **/etc/publickey** file.

Network File System (NFS) Overview for System Management in *Networks and communication management*.

Network Information Service (NIS) in *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*.

NIS Reference.

chlang Command

Purpose
Changes the language settings for system or user.

Syntax

To Modify the Environment or Profile File Changing the Default Language Setting:
chlang [**-u** *UID* | *Uname*] [**-m** *MsgTransLst* | **-M**] *Language*

To Modifiy the Environment or Profile File without Changing the Default Language Setting:
chlang [**-u** *UID* | *Uname*] **-m** *MsgTransLst* | **-M**

To Remove the NLSPATH Setting from the Environment or Profile File:
chlang -d [**-u** *UID* | *UName*]

Description
The **chlang** command is a high-level shell command that changes the language settings for either the entire system or an individual user. If the effective id of the invoker is root and the **-u** option was not used, the language settings will be changed for the entire system in the **/etc/environment** file. If the effective id of the invoker is not root, or if the **-u** option was used, the language settings will be changed for an individual user in the user's **.profile** file.

When **chlang** is run with a language and no options, the **LANG** environment variable will be set to the language specified.

When chlang is run with the **-m** option, the **LANG** and **NLSPATH** environment variables will be set. In addition, the **LC_MESSAGES** variable will be set to the first value specified in the *MsgTransLst* of the **-m** flag if it is different from the **Language** parameter and the **Language** parameter has a system supplied translation available.

When **chlang** is run with the **-d** option, the **NLSPATH** environment variable will be removed.

Notes:

1. Changes made to the NLS environment by **chlang** are not immediate when either **/etc/environment** or the user's **.profile** are modified. Changes to **/etc/environment** requires rebooting the sytsem. Changes to a user's **.profile** requires logging in again or running the **.profile** file.
2. When modifying a user's configuration file, if the user uses the C shell (**/usr/bin/csh**) their **.cshrc** file will be modified rather than the **.profile** file.

Flags

-d	Used to remove the **NLSPATH** environment variable. This option will remove **NLSPATH** from either **/etc/environment** or the user's **.profile**. If **NLSPATH** was not currently in the file being modified, a warning message will be displayed.
-m *MsgTransLst*	Used to make modifications to the **NSLPATH** environment variable. *MsgTransLst* is a colon-separated list of message translations (locale names) that indicates the message translation hierarchy required for the system or user. If the first language in the list is different from the **Language** parameter and **Language** parameter has system supplied translation, then the **LC_MESSAGES** environment variable will be set to that first value. If the first language-territory in the list is the same as the language being set, the **LC_MESSAGES** environment variable will be removed. All entries in the list become hard coded directories in the **NLSPATH** environment.
-M	Used to reset the **LC_MESSAGES** environment variable and set the **NLSPATH** environment variable to the default translation hierarchy, which is: **/usr/lib/nls/msg/%L/%N:** **/usr/lib/nls/msg/%L/%N.cat:**
-u *UID* or *UName*	Used to make modification to an individual user. The user can be specified by either user id number or user login name. If the effective id of **chlang** is root, the **-u** parameter must be used to change the language environment for any specific user ID, including root itself (no **-u** parameter in this case will update the **/etc/environment** file rather than root's **.profile**). If the effective id is not root, the **-u** parameter is not needed. If it is specified, it must be equal to the effective id of the invoker.
Language	This is the language-territory (locale name) that will become the locale setting for the LANG environment variable.

Exit Status

0	Indicates successful completion.
>0	Indicates an error occurred.

Examples

1. Assume the preferred locale is Norwegian, and the language translations in order of preference are Norwegian, Swedish, and English. The command to achieve this for user *amcleod* is as follows:

```
chlang -u amcleod -m no_NO:sv_SE:en_US no_NO
```

The following settings would be made in the **.profile** for user *amcleod*. Because the first language in the message translation list is Norwegian, as is the **Language** parameter, **LC_MESSAGES** would not be set by **chlang**. If **LC_MESSAGES** had been set, it would be removed:

```
LANG=no_NO

NLSPATH=/usr/lib/nls/msg/%L/%N:
        /usr/lib/nls/msg/no_NO/%N:
        /usr/lib/nls/msg/sv_SE/%N:
        /usr/lib/nls/msg/en_US/%N:
        /usr/lib/nls/msg/%L/%N.cat:
        /usr/lib/nls/msg/no_NO/%N.cat:
        /usr/lib/nls/msg/sv_SE/%N.cat:
        /usr/lib/nls/msg/en_US/%N.cat
```

2. Assume the preferred locale is French, and the language translations in order of preference are French Canadian and English. To achieve this for a non-root user enter:

   ```
   chlang -m fr_CA:en_US fr_FR
   ```

   The following settings would be made in the **.profile** file for the user invoking **chlang**. Because the first language in the message translation list is different from the cultural convention (locale), **LC_MESSAGES** is set by **chlang**.

   ```
   LANG=fr_FR

   LC_MESSAGES=fr_CA

   NLSPATH=/usr/lib/nls/msg/%L/%N:
           /usr/lib/nls/msg/fr_CA/%N:
           /usr/lib/nls/msg/en_US/%N:
           /usr/lib/nls/msg/%L/%N.cat:
           /usr/lib/nls/msg/fr_CA/%N.cat:
           /usr/lib/nls/msg/en_US/%N.cat
   ```

3. Assume that a system administrator (root authority) in Spain is configuring a system from another country, and needs to change the default language environment so the machine operates properly in its new location. To change the default in the **/etc/environment** file, enter:

   ```
   chlang -m es_ES es_ES
   ```

   The following settings would be made in the **/etc/environment** file.

   ```
   LANG=es_ES

   NLSPATH=/usr/lib/nls/msg/%L/%N:
           /usr/lib/nls/msg/es_ES/%N:
           /usr/lib/nls/msg/%L/%N.cat:
           /usr/lib/nls/msg/es_ES/%N.cat
   ```

Files

/usr/bin/chlang	Change language command
/etc/environment	Specifies basic environment for all processes
$HOME/.profile	Specifies environment for specific user needs

Related Information

The **environment** file, **profile** file format.

National Language Support Overview in *AIX 5L Version 5.3 National Language Support Guide and Reference*.

Understanding Locale Environment Variables in *Operating system and device management*.

chlicense Command

Purpose

Changes the number of fixed licenses and the status of the floating licensing of the system.

Syntax

chlicense [[**-D** | **-l**] **-u** *FixedUsers*]] [[**-v**] **-f** *FloatingStatus*]

Note: At least one flag must be specified with the **chlicense** command.

Description

There are two types of user licensing: fixed and floating. Fixed licensing is always enabled and the number of licenses can be changed using **-u** flag of the **chlicense** command. Floating licensing is enabled or disabled using the **-f** flag.

Flags

Note: At least one flag must be specified with the **chlicense** command.

-D	The **-D** flag causes the new fixed-license value to be updated in the **login.cfg** file only. This is the option if the **-l** flag is not issued. You must restart the system before the new number takes effect.
-f *FloatingStatus*	Changes the status of the floating licensing of the system. The status must be either **on** or **off**. The status of **on** enables the floating licensing and **off** disables the floating licensing. The **-f** flag is optional.
-l	The **-l** flag causes the chlicense command to modify the current value of the fixed-license counting semaphore, in addition to modifying the value in the **login.cfg** file.
-u *FixedUser*	Changes the number of fixed licenses on a system. The value of *FixedUser* must be a number greater than 0. The **-u** flag is optional.

Security

RBAC Disclaimer

Examples

1. To enable the floating licensing for the system, enter:
   ```
   chlicense -f on
   ```
2. To disable the floating licensing for the system, enter:
   ```
   chlicense -f off
   ```
3. To change the number of fixed licenses to 125 and to enable floating licensing on the system, enter:
   ```
   chlicense -u 125 -f on
   ```
4. To immediatly increase the number of fixed licenses to 5, enter:
   ```
   chlicense -I -u 5
   ```

Related Information

The **lslicense** and **monitord** daemon.

chlpclacl Command

Purpose

Changes the access controls for the least-privilege (LP) resource class (**IBM.LPCommands**).

Syntax

To add one or more accesses to the **IBM.LPCommands** Class ACL or to overwrite the **IBM.LPCommands** Class ACL with one or more accesses:

chlpclacl [**-a** | **-n** *host1*[**,***host2*,...]] [**-o**] [**-h**] [**-TV**] *ID_1 perm1* [*ID_2 perm2*] ...

To add one or more accesses to the **IBM.LPCommands** Class ACL or to overwrite the **IBM.LPCommands** Class ACL with one or more accesses all using the same permissions:

chlpclacl [**-a** | **-n** *host1*[**,***host2*,...]] **-l** [**-o**] [**-h**] [**-TV**] *ID_1* [*ID_2*...] *perm*

To delete one or more accesses from the **IBM.LPCommands** Class ACL:

chlpclacl [**-a** | **-n** *host1*[**,***host2*,...]] **-d** [**-h**] [**-TV**] *ID_1* [*ID_2*...]

To add accesses to (or remove accesses from) the **IBM.LPCommands** Class ACL or to overwrite the **IBM.LPCommands** Class ACL, with the accesses specified in a file:

chlpclacl [**-a** | **-n** *host1*[**,***host2*,...]] [**-o** | **-d**] **-f** *file_name* [**-h**] [**-TV**]

To set the **IBM.LPCommands** Class ACL to deny all accesses:

chlpclacl [**-a** | **-n** *host1*[**,***host2*,...]] **-x** [**-h**] [**-TV**]

Description

The **chlpclacl** command changes the access control list (ACL) that is associated with the least-privilege (LP) resource class (**IBM.LPCommands**). This command allows an access to be added to or removed from the **IBM.LPCommands** Class ACL. This ACL controls access to such class operations as creating LP resources and deleting LP resources. One Class ACL exists on each node for the **IBM.LPCommands** class.

To add accesses to the **IBM.LPCommands** Class ACL, specify the ID and the permission the ID is to have. More than one ID and permission pair can be specified. If you want to add multiple IDs and they will all have the same permission, use the **-l** flag to indicate that the format of the command is a list of IDs followed by a single permission that applies to all of the IDs. If you use the **-o** flag, the IDs and permissions specified with the command will overwrite the existing accesses. The previously-defined accesses in the Class ACL are deleted.

To delete accesses from the **IBM.LPCommands** Class ACL, use the **-d** flag and specify the IDs to be deleted.

Use the **-f** flag to indicate that the accesses are specified in a file. Each line of the file will be an ID and permission for that ID. If the **-d** flag is used with the **-f** flag, only the ID is needed on each line. Everything after the first space is ignored.

This command runs on any node. If you want this command to run on all of the nodes in a domain, use the **-a** flag. If you want this command to run on a subset of nodes in a domain, use the **-n** flag. Otherwise, this command runs on the local node.

Flags

–a Changes **IBM.LPCommands** Class ACLs 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 **chlpclacl** 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, **chlpclacl –a** runs in the management domain. To run **chlpclacl –a** in the peer domain, you must set **CT_MANAGEMENT_SCOPE** to **2**.

–d Removes the ACL entry for the specified ID from the **IBM.LPCommands** Class ACL.

–f *file_name*
Indicates that the accesses are specified in *file_name*. Each line of this file consists of an ID and the permission for that ID. If the **-d** flag is used with the **-f** flag, only the ID is needed on each line. Everything after the first space is ignored.

–l Indicates that there is a list of IDs followed by a single permission that is used for all of the IDs.

–n *host1***[,***host2***,...]**
Specifies the nodes in the domain on which the **IBM.LPCommands** Class ACL should be changed. By default, the **IBM.LPCommands** Class ACL is changed 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 will run once for the first valid scope found.

–o Indicates that the specified accesses overwrite any existing ACL entries for the **IBM.LPCommands** Class ACL. Any ACL entries in the **IBM.LPCommands** Class ACL are deleted.

–x Sets the **IBM.LPCommands** Class ACL to deny all accesses to the **IBM.LPCommands** class attributes and class operations. Any ACL entries in the **IBM.LPCommands** Class ACL are deleted.

–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

ID Specifies the network identity of the user. If the same *ID* is listed more than once, the last permission specified is used. For a description of how to specify the network identity, see the **User identities** section of the **lpacl** information file.

perm Specifies the permission allowed for *ID*. *perm* is specified as a string of one or more characters, where each character represents a particular permission. The valid values for *perm* are:

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 **lpacl** information file for descriptions of these permissions.

Security

To run the **chlpclacl** command, you need read and administrator permission in the Class ACL of the **IBM.LPCommands** resource class. Permissions are specified in the LP ACLs on the contacted system. See the **lpacl** information 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 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_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.

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 give user **joe** on **nodeA** write permission to the **IBM.LPCommands** class so that he can create LP resources on **nodeA**, run one of these commands on **nodeA**:

   ```
   chlpclacl joe@NODEID w

   chlpclacl joe@LOCALHOST w
   ```

2. **nodeA** and **nodeB** are in a peer domain. To give user **joe** on **nodeB** write permission to the **IBM.LPCommands** class so that he can create LP resources on **nodeB**, run this command on **nodeA**:

   ```
   chlpclacl -n nodeB joe@LOCALHOST  w
   ```

   In this example, specifying **joe@NODEID** instead of **joe@LOCALHOST** gives **joe** on **nodeA** write permission to the **IBM.LPCommands** class on **nodeB**.

3. To give user **joe** on **nodeA** write permission to the **IBM.LPCommands** class and **bill** on **nodeA** administrator permission and write permission to the **IBM.LPCommands** class on **nodeA**, run this command on **nodeA**:

   ```
   chlpclacl joe@LOCALHOST w bill@LOCALHOST wa
   ```

4. To give user **joe** on **nodeA** administrator permission to the **IBM.LPCommands** class on **nodeA**, overwriting the current **IBM.LPCommands** Class ACL so that this is the only access allowed, run this command on **nodeA**:

   ```
   chlpclacl -o joe@LOCALHOST a
   ```

5. To give users **joe**, **bill**, and **jane** on **nodeA** read and write permissions to the **IBM.LPCommands** class on **nodeA**, run this command on **nodeA**:

   ```
   chlpclacl -l joe@LOCALHOST  bill@LOCALHOST  jane@LOCALHOST  rw
   ```

6. To delete access for **joe** on **nodeA** from the **IBM.LPCommands** class on **nodeA**, run this command on **nodeA**:

   ```
   chlpclacl -d  joe@LOCALHOST
   ```

7. To add a list of accesses that are in a file named **/mysecure/aclfile** on **nodeA** to the **IBM.LPCommands** class on **nodeA**, run this command on **nodeA**:

   ```
   chlpclacl -f /mysecure/aclfile
   ```

   The contents of **/mysecure/aclfile** on **nodeA** could be:

   ```
   joe@LOCALHOST     w
   bill@LOCALHOST    wa
   jane@LOCALHOST    rw
   ```

8. To deny all accesses to the **IBM.LPCommands** class on **nodeA**, run this command on **nodeA**:

   ```
   chlpclacl -x
   ```

Location

/usr/sbin/rsct/bin/chlpclacl Contains the **chlpclacl** command

Related Information

Books: *RSCT: Administration Guide*, for information about:

- the least-privilege (LP) resource manager
- how to use ACLs

Commands: **chlpracl**, **chlpriacl**, **chlprsacl**, **lslpclacl**, **lslpcmd**, **lslpracl**, **lslpriacl**, **lslprsacl**, **mklpcmd**, **rmlpcmd**, **runlpcmd**

Information Files: **lpacl**, for general information about LP ACLs

---

chlpcmd Command

Purpose

Changes the attribute values of a least-privilege (LP) resource.

Syntax

To change the attribute values of an LP resource:

- On the local node:

  chlpcmd [**–l 0** | **1**] [**–c 0** | **1** | **2** | **3**] [**–h**] [**–TV**] *resource_name attr1*=*value1* [*attr2*=*value2*...]

  chlpcmd –r [**–h**] [**–TV**] *resource_name*
- On all nodes in a domain:

  chlpcmd –a [**–l 0** | **1**] [**–c 0** | **1** | **2** | **3**] [**–h**] [**–TV**] *resource_name attr1*=*value1* [*attr2*=*value2*...]

  chlpcmd –a –r [**–h**] [**–TV**] *resource_name*
- On a subset of nodes in a domain:

  chlpcmd –n *host1* [**,***host2***,**...] [**–l 0** | **1**] [**–c 0** | **1** | **2** | **3**] [**–h**] [**–TV**] *resource_name attr1*=*value1* [*attr2*=*value2*...]

  chlpcmd –n *host1* [**,***host2***,**...] **–r** [**–h**] [**–TV**] *resource_name*

Description

Use the **chlpcmd** command to change any of the read/write attribute values of an LP resource. 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). Use the **-r** flag to recalculate and assign the **CheckSum** attribute. Use the **-c** flag to change the **ControlFlags** attribute. Use the **-l** flag to change the **Lock** attribute. Use *attr*=*value* parameters to modify these attributes: **Name**, **CommandPath**, **RunCmdName**, **FilterScript**, **FilterArg**, and **Description**.

This command runs on any node. If you want this command to run on all of the nodes in a domain, use the **-a** flag. If you want this command to run on a subset of nodes in a domain, use the **-n** flag. Otherwise, this command runs on the local node.

Flags

–a Changes attribute values for *resource_name* 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 **chlpcmd** 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, **chlpcmd –a** runs in the management domain. To run **chlpcmd –a** in the peer domain, you must set **CT_MANAGEMENT_SCOPE** to **2**.

–n *host1***[,***host2***,...]**
Specifies one or more nodes in the domain on which the LP resource is to be changed. By default, the LP resource is changed on the local node. This flag is valid only in a management domain or a peer domain. If the **CT_MANAGEMENT_SCOPE** environment 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 **chlpcmd** command runs once for the first valid scope that the LP resource manager finds.

–r Recalculates and assigns the **CheckSum** attribute value for this LP resource. Use the **-r** flag when:
- You have modified the command or script that this LP resource represents.
- You want to change the **CheckSum** value from **0** to the correct value after the command or script becomes available on the system.

–l 0 | 1
Locks or unlocks the resource. You can use this flag to protect the resource from being deleted by accident. The default value is **0**, which means no lock is set. To lock the resource, use **chlpcmd -l 1**.

–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 **runlpcmd** 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 **runlpcmd** 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.

–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
Specifies the name of the LP resource to change.

attr1=*value1* **[***attr2*=*value2***...]**
Specifies one or more read/write attributes and their new values.

Security

To run the **chlpcmd** command, you need:

- read permission in the Class ACL of the **IBM.LPCommands** resource class.
- write 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 **lpacl** 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 **CT_MANAGEMENT_SCOPE** 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 change the **Lock** attribute of LP resource **lpcommand1** before deleting a resource on a local node, enter:

   ```
   chlpcmd -l 0 lpcommand1
   ```

2. Suppose **nodeA** is in a management domain and **CT_MANAGEMENT_SCOPE** is set to **3**. To recalculate the **CheckSum** attribute value of LP resource **lpcommand2** on **nodeA**, enter:

   ```
   chlpcmd -r -n nodeA lpcommand2
   ```

Location

/usr/sbin/rsct/bin/chlpcmd Contains the **chlpcmd** command

Related Information

Books: *RSCT Administration Guide* , for information about:
- modifying LP ACLs
- LP resource attributes and their definitions

Commands: **lphistory**, **lslpcmd**, **mklpcmd**, **rmlpcmd**, **runlpcmd**

Information Files:
- **lpacl**, for general information about LP ACLs
- **rmccli**, for general information about RMC commands and *attr*=*value* syntax

---

chlpracl Command

Purpose

Changes the access controls for a least-privilege (LP) resource.

Syntax

To add one or more accesses to a Resource ACL or to overwrite a Resource ACL with one or more accesses:

chlpracl [**-a** | **-n** *host1*[**,***host2*,...]] [**-o**] [**-r**] [**-h**] [**-TV**] *resource ID_1 perm1* [*ID_2 perm2*] ...

To add one or more accesses to a Resource ACL or to overwrite an Resource ACL with one or more accesses all using the same permissions:

chlpracl [**-a** | **-n** *host1*[**,***host2*,...]] **-l** [**-o**] [**-r**] [**-h**] [**-TV**] *resource ID_1* [*ID_2*...] *perm*

To delete one or more accesses from a Resource ACL:

chlpracl [**-a** | **-n** *host1*[**,***host2*,...]] **-d** [**-r**] [**-h**] [**-TV**] *resource ID_1* [*ID_2*...]

To add accesses to (or remove accesses from) a Resource ACL or to overwrite a Resource ACL, with the accesses specified in a file:

chlpracl [**-a** | **-n** *host1*[**,***host2*,...]] [**-o** | **-d**] **-f** *file_name* [**-r**] [**-h**] [**-TV**] *resource*

To set a Resource ACL so that no permissions are allowed, or to use the Resource Shared ACL:

chlpracl [**-a** | **-n** *host1*[**,***host2*,...]] { **-b** | **-x** } [**-r**] [**-h**] [**-TV**] *resource*

To set all of the Resource ACLs so that no permissions are allowed, or to use the Resource Shared ACL:

chlpracl [**-a** | **-n** *host1*[**,***host2*,...]] { **-B** | **-X** } [**-h**] [**-TV**]

Description

The **chlpracl** command changes the access control list (ACL) that is associated with a least-privilege (LP) resource. This command allows an access to be added to or removed from the Resource ACL. This ACL controls access to such resource operations as listing attribute values and running LP commands. One Resource ACL exists for each LP resource.

For controlling access to the LP resource, three different types of Resource ACLs exist:

1. Resource ACL
2. Resource Initial ACL
3. Resource Shared ACL

The **chlpracl** command allows the Resource ACL to indicate that the Resource Shared ACL should be used in its stead to control access. For descriptions of these ACLs, see the **lpacl** information file.

To add an access to the Resource ACL, specify the name of the LP resource, the ID, and the permission the ID is to have. More than one ID and permission pair can be specified. If you want to add multiple IDs and they will all have the same permission, use the **-l** flag to indicate that the format of the command is a list of IDs followed by a single permission that applies to all of the IDs. If you use the **-o** flag, the IDs and permissions specified with the command will overwrite the existing accesses. The previously-defined accesses in the ACL are deleted.

To delete accesses from the Resource ACL, use the **-d** flag and specify the name of the LP resource and the IDs to be deleted.

Use the **-f** flag to indicate that the accesses are specified in a file. Each line of the file will be an ID and permission for that ID. If the **-d** flag is used with the **-f** flag, only the ID is needed on each line. Everything after the first space is ignored.

This command runs on any node. If you want this command to run on all of the nodes in a domain, use the **-a** flag. If you want this command to run on a subset of nodes in a domain, use the **-n** flag. Otherwise, this command runs on the local node.

Flags

–a Changes the Resource ACLs for *resource* 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 **chlpracl** 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, **chlpracl –a** runs in the management domain. To run **chlpracl –a** in the peer domain, you must set **CT_MANAGEMENT_SCOPE** to **2**.

–b Bypasses the ACL for the specified LP resource. The Resource Shared ACL is used for access control for this LP resource. Any ACL entries in the Resource ACL are deleted.

–B Bypasses the ACLs for all LP resources. The Resource Shared ACL is used for access control for all LP resources. Any ACL entries in the Resource ACLs are deleted. One Resource Shared ACL exists for each **IBM.LPCommands** class (or node).

–d Removes the ACL entry for the specified ID from the specified Resource ACL.

–f *file_name*
Indicates that the accesses are specified in *file_name*. Each line of this file consists of an ID and the permission for that ID. If the **-d** flag is used with the **-f** flag, only the ID is needed on each line. Everything after the first space is ignored.

–l Indicates that there is a list of IDs followed by a single permission that is used for all of the IDs.

–n *host1***[,***host2***,...]**
Specifies the nodes in the domain on which the Resource ACL should be changed. By default, the Resource ACL is changed 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 will run once for the first valid scope found.

–o Indicates that the specified ACL accesses overwrite any existing ACL entries for the specified Resource ACL. Any ACL entries in the Resource ACL are deleted.

–r Indicates that *resource* is a ″typical″ RSCT resource handle. The resource handle must be enclosed in quotation marks. The Resource ACL of the resource handle is modified.

–x Sets the Resource ACL for the specified LP resource to deny all accesses to the LP resource. Any ACL entries in the Resource ACL are deleted.

–X Sets the Resource ACL of all LP resources to deny all accesses to the LP resource. Any ACL entries in the Resource ACLs are deleted.

–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
Specifies the name of the LP resource for which the Resource ACL is changed.

ID Specifies the network identity of the user. If the same *ID* is listed more than once, the last permission specified is used. For a description of how to specify the network identity, see the **lpacl** information file.

perm Specifies the permission allowed for *ID*. *perm* is specified as a string of one or more characters, where each character represents a particular permission. The valid values for *perm* are:

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 **lpacl** information file for a description of each permission and how it applies.

Security

To run the **chlpracl** command, you need:

- read permission in the Class ACL of the **IBM.LPCommands** resource class.
- read and administrator permission in the Resource ACL.

  As an alternative, the Resource ACL can direct the use of the Resource Shared ACL if these permissions exist in the Resource Shared ACL.

Permissions are specified in the LP ACLs on the contacted system. See the **lpacl** information 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 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_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.

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 give user **joe** on **nodeA** the ability to run the LP command **lpcommand1** on **nodeA**, run one of these commands on **nodeA**:

   ```
   chlpracl lpcommand1 joe@NODEID  x

   chlpracl lpcommand1 joe@LOCALHOST  x
   ```

2. **nodeA** and **nodeB** are in a peer domain. To give user **joe** on **nodeB** the ability to run the LP command **lpcommand1** on **nodeB**, run this command on **nodeA**:

   ```
   chlpracl -n nodeB lpcommand1 joe@LOCALHOST  x
   ```

   In this example, specifying **joe@NODEID** instead of **joe@LOCALHOST** gives **joe** on **nodeA** the ability to run the LP command **lpcommand1** on **nodeB**.

3. To give user **joe** on **nodeA** execute permission to the LP command **lpcommand1** and **bill** on **nodeA** administrator permission and write permission to the same resource on **nodeA**, run this command on **nodeA**:

   ```
   chlpracl lpcommand1 joe@LOCALHOST  x  bill@LOCALHOST  wa
   ```

4. To give user **joe** on **nodeA** administrator permission to the LP command **lpcommand1** on **nodeA**, overwriting the current ACLs for **lpcommand1** so that this is the only access allowed, run this command on **nodeA**:

   ```
   chlpracl -o lpcommand1 joe@LOCALHOST x
   ```

5. To give users **joe**, **bill**, and **jane** on **nodeA** the ability to run the LP command **lpcommand1** on **nodeA**, run this command on **nodeA**:

   ```
   chlpracl lpcommand1 -l joe@LOCALHOST  bill@LOCALHOST  jane@LOCALHOST  x
   ```

6. To delete access for **joe** on **nodeA** from the ACLs for the LP command **lpcommand1** on **nodeA**, run this command on **nodeA**:

   ```
   chlpracl -d lpcommand1 joe@LOCALHOST
   ```

7. To add a list of accesses that are in a file named **/mysecure/aclfile** on **nodeA** to the LP command **lpcommand1** on **nodeA**, run this command on **nodeA**:

   ```
   chlpracl -f /mysecure/aclfile lpcommand1
   ```

The contents of **/mysecure/aclfile** on **nodeA** could be:

```
joe@LOCALHOST     x
bill@LOCALHOST                  ax
jane@LOCALHOST    wx
```

8. To bypass the Resource ACL for the LP command **lpcommand1** on **nodeA**, and use the Resource Shared ACL to control access to it, run this command on **nodeA**:

   ```
   chlpracl -b lpcommand1
   ```

9. To bypass the Resource ACLs for all of the LP resources on **nodeA**, and use the Resource Shared ACL to control accesses, run this command on **nodeA**:

   ```
   chlpracl -B
   ```

10. To deny all accesses to the LP command **lpcommand1** on **nodeA**, run this command on **nodeA**:

    ```
    chlpracl -x lpcommand1
    ```

Location

/usr/sbin/rsct/bin/chlpracl Contains the **chlpracl** command

Related Information

Books: *RSCT: Administration Guide*, for information about:

- the least-privilege (LP) resource manager
- how to use ACLs

Commands: **chlpclacl**, **chlpriacl**, **chlprsacl**, **lslpclacl**, **lslpcmd**, **lslpracl**, **lslpriacl**, **lslprsacl**, **mklpcmd**, **rmlpcmd**, **runlpcmd**

Information Files: **lpacl**, for general information about LP ACLs

---

chlpriacl Command

Purpose

Changes the access controls for the least-privilege (LP) Resource Initial ACL.

Syntax

To add one or more accesses to the Resource Initial ACL or to overwrite the Resource Initial ACL with one or more accesses:

chlpriacl [**-a** | **-n** *host1*[**,***host2***,**...]] [**-o**] [**-h**] [**-TV**] *ID_1 perm1* [*ID_2 perm2*] ...

To add one or more accesses to the Resource Initial ACL or to overwrite the Resource Initial ACL with one or more accesses all using the same permissions:

chlpriacl [**-a** | **-n** *host1*[**,***host2***,**...]] **-l** [**-o**] [**-h**] [**-TV**] *ID_1* [*ID_2*...] *perm*

To delete one or more accesses from the Resource Initial ACL:

chlpriacl [**-a** | **-n** *host1*[**,***host2***,**...]] **-d** [**-h**] [**-TV**] *ID_1* [*ID_2*...]

To add accesses to (or remove accesses from) the Resource Initial ACL or to overwrite the Resource Initial ACL, with the accesses specified in a file:

chlpriacl [**-a** | **-n** *host1*[**,***host2***,**...]] [**-o** | **-d**] **-f** *file_name* [**-h**] [**-TV**]

To set the Resource Initial ACL to use the Resource Shared ACL or so that no permissions are allowed:

chlpriacl [**-a** | **-n** *host1*[**,***host2*,...]] { **-b** | **-x** } [**-h**] [**-TV**]

Description

The **chlpriacl** command changes the access control list (ACL) that is associated with the least-privilege (LP) Resource Initial ACL. This command allows a user to be added to or removed from the Resource Initial ACL. This ACL is used to initialize a Resource ACL when the LP resource is created. The Resource Initial ACL can consist of ACL entries that define permissions to the LP resource or it can indicate that the Resource Shared ACL should be used to control access instead of the Resource ACL. One Resource Initial ACL exists on each node for the **IBM.LPCommands** class.

To add accesses to the Resource Initial ACL, specify the ID and the permission the ID is to have. More than one ID and permission pair can be specified. If you want to add multiple IDs and they will all have the same permission, use the **-l** flag to indicate that the format of the command is a list of IDs followed by a single permission that applies to all of the IDs. If you use the **-o** flag, the IDs and permissions specified with the command will overwrite the existing accesses. The previously-defined accesses in the ACL are deleted.

To delete accesses from the Resource Initial ACL, use the **-d** flag and specify the IDs to be deleted.

Use the **-f** flag to indicate that the accesses are specified in a file. Each line of the file will be an ID and permission for that ID. If the **-d** flag is used with the **-f** flag, only the ID is needed on each line. Everything after the first space is ignored.

This command runs on any node. If you want this command to run on all of the nodes in a domain, use the **-a** flag. If you want this command to run on a subset of nodes in a domain, use the **-n** flag. Otherwise, this command runs on the local node.

Flags

–a Changes the Resource Initial ACLs 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 **chlpriacl** 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, **chlpriacl –a** runs in the management domain. To run **chlpriacl –a** in the peer domain, you must set **CT_MANAGEMENT_SCOPE** to **2**.

–b Sets the Resource Initial ACL to indicate that the Resource ACL is bypassed and that the Resource Shared ACL is used for access control for the LP resource. Any ACL entries in the Resource Initial ACL are deleted. When a new LP resource is created, the Resource Shared ACL is used for it.

–d Removes the ACL entry for the specified ID from the Resource Initial ACL.

–f *file_name*
Indicates that the accesses are specified in *file_name*. Each line of this file consists of an ID and the permission for that ID. If the **-d** flag is used with the **-f** flag, only the ID is needed on each line. Everything after the first space is ignored.

–l Indicates that there is a list of IDs followed by a single permission that is used for all of the IDs.

–n *host1*[**,***host2*,**...**]
Specifies the node in the domain on which the Resource Initial ACL should be changed. By default, the Resource Initial ACL is changed 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 will run once for the first valid scope found.

–o Indicates that the specified ACL entries overwrite any existing ACL entries for the Resource Initial ACL. Any ACL entries in the Resource Initial ACL are deleted.

–x Sets the Resource Initial ACL to deny all accesses to the LP resource. Any ACL entries in the Resource Initial ACL are deleted. When a new LP resource is created, all accesses will be denied to it.

–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

ID Specifies the network identity of the user. If the same *ID* is listed more than once, the last permission specified is used. For a description of how to specify the network identity, see the **lpacl** information file.

perm Specifies the permission allowed for *ID*. *perm* is specified as a string of one or more characters, where each character represents a particular permission. The valid values for *perm* are:

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 **lpacl** information file for a description of each permission and how it applies.

Security

To run the **chlpriacl** command, you need read and administrator permission in the Class ACL of the **IBM.LPCommands** resource class. Permissions are specified in the LP ACLs on the contacted system. See the **lpacl** information 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 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_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.

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 give user **joe** on **nodeA** execute permission in the Resource Initial ACL on **nodeA**, run one of these commands on **nodeA**:

```
chlpriacl joe@NODEID  x

chlpriacl joe@LOCALHOST  x
```

2. **nodeA** and **nodeB** are in a peer domain. To give user **joe** on **nodeB** execute permission to the Resource Initial ACL on **nodeB**, run this command on **nodeA**:

   ```
   chlpriacl -n nodeB joe@LOCALHOST  x
   ```

   In this example, specifying **joe@NODEID** instead of **joe@LOCALHOST** gives **joe** on **nodeA** execute permission to the Resource Initial ACL on **nodeB**.
3. To give user **joe** on **nodeA** execute permission and **bill** on **nodeA** administrator permission and read permission to the Resource Initial ACL on **nodeA**, run this command on **nodeA**:

   ```
   chlpriacl joe@LOCALHOST  x  bill@LOCALHOST  ra
   ```
4. To give user **joe** on **nodeA** execute permission to the Resource Initial ACL on **nodeA**, overwriting the current ACLs so that this is the only access allowed, run this command on **nodeA**:

   ```
   chlpriacl -o joe@LOCALHOST x
   ```
5. To give users **joe**, **bill**, and **jane** on **nodeA** read permission and write permission to the Resource Initial ACL on **nodeA** on **nodeA**, run this command on **nodeA**:

   ```
   chlpriacl -l joe@LOCALHOST  bill@LOCALHOST jane@LOCALHOST  rw
   ```
6. To delete access for **joe** on **nodeA** from the Resource Initial ACL on **nodeA**, run this command on **nodeA**:

   ```
   chlpriacl -d  joe@LOCALHOST
   ```
7. To add a list of accesses that are in a file named **/mysecure/aclfile** on **nodeA** to the Resource Initial ACL on **nodeA**, run this command on **nodeA**:

   ```
   chlpriacl -f /mysecure/aclfile
   ```

   The contents of **/mysecure/aclfile** on **nodeA** could be:

   ```
   joe@LOCALHOST     x
   bill@LOCALHOST    rw
   jane@LOCALHOST    rwa
   ```
8. To set the Resource Initial ACL on **nodeA** so it indicates that the Resource Shared ACL on **nodeA** is used to control accesses for newly-created LP resources on **nodeA**, run this command on **nodeA**:

   ```
   chlpriacl -b
   ```
9. To set the Resource Initial ACL on **nodeA** so that it denies all accesses for newly-created LP resources on **nodeA**, run this command on **nodeA**:

   ```
   chlpriacl -x
   ```

Location

/usr/sbin/rsct/bin/chlpriacl Contains the **chlpriacl** command

Related Information

Books: *RSCT: Administration Guide*, for information about:

- the least-privilege (LP) resource manager
- how to use ACLs

Commands: **chlpclacl**, **chlpracl**, **chlprsacl**, **lslpclacl**, **lslpcmd**, **lslpracl**, **lslpriacl**, **lslprsacl**, **mklpcmd**, **rmlpcmd**, **runlpcmd**

Information Files: **lpacl**, for general information about LP ACLs

---

chlprsacl Command

Purpose

Changes the access controls for the least-privilege (LP) Resource Shared ACL.

Syntax

To add one or more accesses to the Resource Shared ACL or to overwrite the Resource Shared ACL with one or more accesses:

chlprsacl [**-a** | **-n** *host1*[**,***host2*,...]] [**-o**] [**-h**] [**-TV**] *ID_1 perm1* [*ID_2 perm2*] ...

To add one or more accesses to the Resource Shared ACL or to overwrite the Resource Shared ACL with one or more accesses all using the same permissions:

chlprsacl [**-a** | **-n** *host1*[**,***host2*,...]] **-l** [**-o**] [**-h**] [**-TV**] *ID_1* [*ID_2*...] *perm*

To delete one or more accesses from the Resource Shared ACL:

chlprsacl [**-a** | **-n** *host1*[**,***host2*,...]] **-d** [**-h**] [**-TV**] *ID_1* [*ID_2*...]

To add accesses to (or remove accesses from) the Resource Shared ACL or to overwrite the Resource Shared ACL, with the accesses specified in a file:

chlprsacl [**-a** | **-n** *host1*[**,***host2*,...]] [**-o** | **-d**] **-f** *file_name* [**-h**] [**-TV**]

To set the Resource Shared ACL so that no permissions are allowed:

chlprsacl [**-a** | **-n** *host1*[**,***host2*,...]] **-x** [**-h**] [**-TV**]

Description

The **chlprsacl** command changes the access control list (ACL) that is associated with the Resource Shared ACL. This command allows a user to be added to or removed from the Resource Shared ACL. This ACL:

- is used to control accesses to LP resources when the Resource ACL indicates that it (the Resource Shared ACL) has control
- can control access to one or more LP resources
- can consist of ACL entries that define permissions to the LP resources

One Resource Shared ACL exists on each node for the **IBM.LPCommands** class.

The **chlpracl** command is used to indicate that the access to an LP resource is controlled by the Resource Shared ACL. The **chlpriacl** command is used to indicate that accesses to newly-created LP resources are controlled by the Resource Shared ACL, by modifying the Resource Initial ACL.

To add accesses to the Resource Shared ACL, specify the ID and the permission the ID is to have. More than one ID and permission pair can be specified. If you want to add multiple IDs and they will all have the same permission, use the **-l** flag to indicate that the format of the command is a list of IDs followed by a single permission that applies to all of the IDs. If you use the **-o** flag, the IDs and permissions specified with the command will overwrite the existing accesses. The previously-defined accesses in the ACL are deleted.

To delete accesses from the Resource Shared ACL, use the **-d** flag and specify the IDs to be deleted.

Use the **-f** flag to indicate that the accesses are specified in a file. Each line of the file will be an ID and permission for that ID. If the **-d** flag is used with the **-f** flag, only the ID is needed on each line. Everything after the first space is ignored.

This command runs on any node. If you want this command to run on all of the nodes in a domain, use the **-a** flag. If you want this command to run on a subset of nodes in a domain, use the **-n** flag. Otherwise, this command runs on the local node.

Flags

–a Changes the Resource Shared ACLs 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 **chlprsacl** 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, **chlprsacl –a** runs in the management domain. To run **chlprsacl –a** in the peer domain, you must set **CT_MANAGEMENT_SCOPE** to **2**.

–d Removes the ACL entry for the specified ID from the Resource Shared ACL.

–f *file_name*
Indicates that the accesses are specified in *file_name*. Each line of this file consists of an ID and the permission for that ID. If the **-d** flag is used with the **-f** flag, only the ID is needed on each line. Everything after the first space is ignored.

–l Indicates that there is a list of IDs followed by a single permission that is used for all of the IDs.

–n *host1***[,***host2***,...]**
Specifies the node in the domain on which the Resource Shared ACL should be changed. By default, the Resource Shared ACL is changed 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 will run once for the first valid scope found.

–o Indicates that the specified ACL entries overwrite any existing ACL entries for the Resource Shared ACL. Any ACL entries in the Resource Shared ACL are deleted.

–x Sets the Resource Shared ACL to deny all accesses to the LP resources that use the Resource Shared ACL. Any ACL entries in the Resource Shared ACL are deleted.

–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

ID Specifies the network identity of the user. If the same *ID* is listed more than once, the last permission specified is used. For a description of how to specify the network identity, see the **lpacl** information file.

perm Specifies the permission allowed for *ID*. *perm* is specified as a string of one or more characters, where each character represents a particular permission. The valid values for *perm* are:

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 **lpacl** information file for a description of each permission and how it applies.

Security

To run the **chlprsacl** command, you need read and administrator permission in the Class ACL of the **IBM.LPCommands** resource class. Permissions are specified in the LP ACLs on the contacted system. See the **lpacl** information 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 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_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.

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 give user **joe** on **nodeA** execute permission in the Resource Shared ACL on **nodeA**, run one of these commands on **nodeA**:

   ```
   chlprsacl joe@NODEID  x

   chlprsacl joe@LOCALHOST  x
   ```

2. **nodeA** and **nodeB** are in a peer domain. To give user **joe** on **nodeB** execute permission to the Resource Shared ACL on **nodeB**, run this command on **nodeA**:

   ```
   chlprsacl -n nodeB joe@LOCALHOST  x
   ```

   In this example, specifying **joe@NODEID** instead of **joe@LOCALHOST** gives **joe** on **nodeA** execute permission to the Resource Shared ACL on **nodeB**.

3. To give user **joe** on **nodeA** execute permission and **bill** on **nodeA** administrator permission and read permission to the Resource Shared ACL on **nodeA**, run this command on **nodeA**:

   ```
   chlprsacl joe@LOCALHOST  x  bill@LOCALHOST  ra
   ```

4. To give user **joe** on **nodeA** execute permission to the Resource Shared ACL on **nodeA**, overwriting the current ACLs so that this is the only access allowed, run this command on **nodeA**:

   ```
   chlprsacl -o joe@LOCALHOST x
   ```

5. To give users **joe**, **bill**, and **jane** on **nodeA** read permission and write permission to the Resource Shared ACL on **nodeA** on **nodeA**, run this command on **nodeA**:

   ```
   chlprsacl -l joe@LOCALHOST  bill@LOCALHOST jane@LOCALHOST  rw
   ```

6. To delete access for **joe** on **nodeA** from the Resource Shared ACL on **nodeA**, run this command on **nodeA**:

   ```
   chlprsacl -d  joe@LOCALHOST
   ```

7. To add a list of accesses that are in a file named **/mysecure/aclfile** on **nodeA** to the Resource Shared ACL on **nodeA**, run this command on **nodeA**:

   ```
   chlprsacl -f /mysecure/aclfile
   ```

   The contents of **/mysecure/aclfile** on **nodeA** could be:

   ```
   joe@LOCALHOST     x
   bill@LOCALHOST    rw
   jane@LOCALHOST    rwa
   ```

8. To set the Resource Shared ACL on **nodeA** so that it denies all accesses for LP resources that use it on **nodeA**, run this command on **nodeA**:

   ```
   chlprsacl -x
   ```

Location

/usr/sbin/rsct/bin/chlprsacl Contains the **chlprsacl** command

Related Information

Books: *RSCT: Administration Guide*, for information about:

- the least-privilege (LP) resource manager
- how to use ACLs

Commands: **chlpclacl**, **chlpracl**, **chlpriacl**, **lslpclacl**, **lslpcmd**, **lslpracl**, **lslpriacl**, **lslprsacl**, **mklpcmd**, **rmlpcmd**, **runlpcmd**

Information Files: **lpacl**, for general information about LP ACLs

---

chlv Command

Purpose

Changes only the characteristics of a logical volume.

Syntax

To Change the Characteristics of a Logical Volume

chlv [**-a** *position*] [**-b** *badblocks*] [**-d** *schedule*] [**-e** *Range*] [**-L** *label*] [**-o Y | N**] [**-p** *permission*] [**-r** *relocate*] [**-s** *strict*] [**-t** *type*] [**-u** *upperbound*] [**-v** *verify*] [**-w** *mirrorwriteconsistency*] [**-x** *maximum*] [**-T O | F**] [**-U** *userid*] [**-G** *groupid*] [**-P** *modes*] [**-m** *copyn=mirrorpool*] [**-M**] *logicalvolume* ...

To Change the Name of a Logical Volume

chlv -n *newlogicalvolume logicalvolume*

Note: Changing the name of a log logical volume requires that you run the **chfs -a** *log=LVName* on each file system using that log.

Description

The **chlv** command changes the characteristics of a logical volume according to the command flags. The *logicalvolume* parameter can be a logical volume name or logical volume ID. Each current characteristic for a logical volume remains in effect unless explicitly changed with the corresponding flag.

The changes you make with the **-a**, **-e**, **-s**, and **-u** flags take effect only when new partitions are allocated or partitions are deleted. The other flags take effect immediately.

To change the name of a logical volume, use the **-n** flag and use the *newlogicalvolume* parameter to represent the new logical volume name. Do not use other flags with this syntax.

If the *volume group* which contains logical volume being changed is in big vg format, **U**, **G**, and **P** flags can be used to set the ownership, group and permissions respectively, of the special device files. Only root user will be able to set these values. If the *volume group* is exported, these values can be restored upon import if **R** flag is specified with importvg command.

Notes:

1. Changes made to the logical volume are not reflected in the file systems. To change file system characteristics, use the **chfs** command.
2. To use this command, you must either have root user authority or be a member of the **system** group.
3. 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.

You can use the Volumes application in Web-based System Manager (wsm) to change logical volume characteristics. You could also use the System Management Interface Tool (SMIT) **smit chlv** fast path to run this command.

Flags

Notes:

1. When changing the characteristics of a striped logical volume, the **-d**, and **-e** flags are not valid.
2. When changing the characteristics of a logical volume in a snapshot volume group or in a volume group that has a snapshot volume group, the **-a**, **-b**, **-r**, **-t**, **-v**, **-w**, **-x**, **-U**, **-G**, **-P**, **-o**, **-d**, **-e**, **-u** and **-s** flags are not valid.

-a *position*
Sets the intraphysical volume allocation policy (the position of the logical partitions on the physical volume). The *position* variable is represented by 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 *badblocks* variable is represented by one of the following:

y Causes bad-block relocation to occur.

n Prevents bad block relocation from occurring.

-d *schedule*
Sets the scheduling policy when more than one logical partition is written. Must use parallel or sequential to mirror striped lv. The *schedule* variable is represented by one of the following:

p Establishes a parallel 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.

When specifying policy of parallel or sequential strictness, set to **s** for super strictness.

-e *range*
Sets the interphysical volume allocation policy (the number of physical volumes to extend across, using the volumes that provide the best allocation). The value of the *range* variable is limited by the *upperbound* variable, set with the **-u** flag, and is represented by one of the following:

x Allocates logical partitions across the maximum number of physical volumes.

m Allocates logical partitions across the minimum number of physical volumes.

-G *groupid*
Specifies group ID for the logical volume special file.

-L *label*
Sets the logical volume label. The maximum size of the *label* variable is 127 characters.

-m *copyn=mirrorpool*
Enables mirror pools to the copies of a logical volume. A mirror pool is assigned to a copy using the *copyn=mirrorpool* parameter. Specify a mirror pool for each copy of the logical volume. To specify more than one *copyn=mirrorpool* pair, provide multiple **-m** *copyn=mirrorpool* flags.

-M
Disables mirror pools for this logical volume.

-n *newlogicalvolume*
Changes the name of the logical volume to that specified by the *newlogicalvolume* variable. Logical volume names must be unique system wide and can range from 1 to 15 characters.

-o Y | N
Turns on/off serialization of overlapping ios. If serialization is turned on 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 like file systems and databases do serialization so serialization should be turned `off`. The default for new logical volumes is `off`.

-p *permission*
Sets the access permission to read-write or read-only. The *permission* variable is represented by one of the following:

w Sets the access permission to read-write.

r Sets the access permission to read-only.

Note: Mounting a JFS file system on a read-only logical volume is not supported.

-P *modes*
Specifies permissions (file modes) for the logical volume special file.

-r *relocate*
Sets the reorganization flag to allow or prevent the relocation of the logical volume during reorganization. The *relocate* variable is represented by one of the following:

y Allows the logical volume to be relocated during reorganization. If the logical volume is striped, the **chlv** command will not let you change the relocation flag to **y**.

n Prevents the logical volume from being relocated during reorganization.

-s *strict*
Determines the strict allocation policy. Copies of a logical partition can be allocated to share or not to share the same physical volume. The *strict* variable is represented by one of the following:

y Sets a strict allocation policy, so copies of a logical partition cannot share the same physical volume.

n Does not set a strict allocation policy, so copies of a logical partition can share the same physical volume.

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

Note: When changing a non superstrict logical volume to a superstrict logical volume you must use the **-u** flag.

-t *type*
Sets the logical volume type. The maximum size is 31 characters. If the logical volume is striped, you cannot change *type* to boot.

-T O | F
For logical volumes in big and scalable volume groups, the **-T** flag changes the device subtype reported by the **IOCINFO** ioctl call. Use the **-T O** option for a subtype of **DS_LVZ** and the **-T F** option for a subtype of **DS_LV**.

Tip: The **-T** flag does not change any behavior of a logical volume beyond the reported subtype.

-U *userid*
Specifies user ID for the logical volume special file.

-u *upperbound*
Sets the maximum number of physical volumes for new allocation. The value of the *upperbound* 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 *stripewidth*.

-v *verify*
Sets the write-verify state for the logical volume. Causes all writes to the logical volume either to be verified with a follow-up read or not to be verified with a follow-up read. The *verify* variable is represented by one of the following:

y Causes all writes to the logical volume to be verified with a follow-up read.

n Causes all writes to the logical volume not to be verified with a follow-up read.

-w *mirrorwriteconsistency*

y or a Turns on *active* mirror write consistency which ensures data consistency among mirrored copies of a logical volume during normal I/O processing.

p Turns on *passive* mirror write consistency which ensures data consistency among mirrored copies during volume group synchronization after a system interruption. **Note:** This functionality is only available on Big Volume Groups.

n No mirror write consistency. See the **-f** flag of the **syncvg** command.

-x *maximum*
Sets the maximum number of logical partitions that can be allocated to the logical volume.

Security

RBAC Disclaimer

Examples

1. To change the interphysical volume allocation policy of logical volume `lv01`, enter:

   ```
   chlv   -e m
   lv01
   ```

   The interphysical volume allocation policy is set to minimum.
2. To change the type of logical volume `lv03`, enter:

   ```
   chlv   -t copy  lv03
   ```
3. To change the permission of logical volume `lv03` to read-only, enter:

   ```
   chlv   -p r  lv03
   ```

   Logical volume `lv03` now has read-only permission.
4. To change the type to `paging` and the maximum number of physical volumes for logical volume `lv03`, enter:

   ```
   chlv  -t paging  -u 10 lv03
   ```

   The change in the type of logical volume takes effect immediately, but the change in the maximum number of physical volumes does not take effect until a new allocation is made.
5. To change the allocation characteristics of logical volume `lv07`, enter:

   ```
   chlv  -a e  -e x  -r y  -s n  -u 5 lv07
   ```

Files

/usr/sbin Directory where **chlv** command resides.

Related Information

The **chfs** command, **extendlv** command, **lslv** command, **mklv** command, **mklvcopy** command, **reorgvg** command, **rmlvcopy** command, **syncvg** command.

The Logical volume storage in *Operating system and device management* explains the Logical Volume Manager, physical volumes, logical volumes, volume groups, organization, ensuring data integrity, and allocation characteristics.

The System Management Interface Tool (SMIT) in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* explains the structure, main menus, and tasks that are done with SMIT.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

---

chlvcopy Command

Purpose

Marks or unmarks mirror copy as a split mirror.

Syntax

chlvcopy [**-f**] { **-B** [**-s**] } | { **-b** [**-c** *copy*] [**-f**] [**-P**] [**-l** *newlvname*] [**-w**] } LV *name*

Description

Notes:

1. To use this command, you must either have root user authority or be a member of the system group.
2. If persistence is used either by using the **-P** flag or by creating a child backup logical volume device by using the **-l** flag, it will cause the volume group to be usable only on AIX 4.3.2 or later. This is true even after removal of split mirror copy designation of the parent logical volume and the child backup logical volumes.
3. For **chlvcopy** to be successful in a concurrent volume group environment, all the concurrent nodes must be at AIX 4.3.2 or later.
4. The **chlvcopy** command is not allowed if the logical volume is in a volume group that has a snapshot volume group or a snapshot volume group.
5. **chfs** should be used to create a split mirror copy when a filesystem resides on the logical volume to be copied.

All partitions of a logical volume must be fresh before **chlvcopy** can mark a mirror copy as a split mirror. Only one copy may be designated as an online split mirror copy.

Although the **chlvcopy** command can mark online split mirror copies on logical volumes that are open (including logical volumes containing mounted file systems), this is not recommended unless the application is at a known state at the time the copy is marked as a split mirror. The split mirror copy is internally consistent at the time the **chlvcopy** command is run, but consistency is lost between the logical volume and the split mirror copy if the logical volume is accessed by multiple processes simultaneously and the application is not at a known state. When marking an open logical volume, data may be lost or corrupted. Logical volumes should be closed before marking online split mirror copies in order to avoid a potential corruption window.

If the persistence flag is not set to prevent the loss of backup data, the volume group should be set to not automatically varyon and the **-n** flag should be used with **varyonvg** to prevent stale partitions from being resynced. If the persistence flag (**-P**) is set, the following applies: In the event of a crash while an online split mirror copy exists (or multiples exist), the existence of copies is retained when the system is rebooted.

Flags

-b	Marks a mirror copy as a split mirror copy.
-c *copy*	Mirror copy to mark as split mirror copy. The allowed values of copy are 1, 2, or 3. If this option is not specified the default for copy is the last mirror copy of the logical volume.
-B	Unmarks a mirror as split mirror copy. It will also attempt to remove the child backup logical volume, if one was created with the **-l** option.
-f	Forces split mirror copy even if there are stale partitions. If used with the **-B** option, the child backup logical volume if one was created with the **-l** option, will be removed with the **force** option.
-l *newlvname*	New name of the backup logical volume. Specifying the **-l** flag also sets the persistence option, allowing applications to access split mirror copy via *newlvname*.
-P	Maintains information about the existence of an online split mirror copy across a reboot and also allows other nodes (in a concurrent mode environment) to be aware of the existence of the online split mirror copy.
-s	Starts a background **syncvg** for the logical volume.
-w	Allows split mirror copy to be writable (default is to create the split mirror copy as READ ONLY).
LV *name*	Logical volume to act on.

Related Information

The **readlvcopy** and **chfs** commands.

---

chmaster Command

Purpose

The **chmaster** command executes the **ypinit** command and restarts the NIS daemons to change a master server.

Syntax

/usr/etc/yp/chmaster [**-s** *HostName* [**,** *HostName ...*]] [**-O** | **-o**] [**-E** | **-e**] [**-P** | **-p**] [**-U** | **-u**] [**-C** | **-c**] [**-I** | **-B** | **-N**]

Description

The **chmaster** command invokes the **ypinit** command to update the NIS maps for the current domain, assuming that the domain name of the system is currently set. After the **ypinit** command completes successfully, the **chmaster** command comments or uncomments the entries in the **/etc/rc.nfs** file for the **ypserv** command, **yppasswdd** command, **ypupdated** command, and **ypbind** command.

You can use the Network application in Web-based System Manager (wsm) to change network characteristics. You could also use the System Management Interface Tool (SMIT) **smit chmaster** fast path to run this command.

Flags

-B	Updates the **/etc/rc.nfs** file to start the appropriate daemons, invokes the **ypinit** command, and starts the daemons.

-C	Starts the **ypbind** daemon along with the **ypserv** daemon. This flag is the default.
-c	Suppresses the start of the **ypbind** daemon.
-E	Exits from the **ypinit** command and the **chmaster** command if errors are encountered. This flag is the default.
-e	Suppresses an exit from the **ypinit** command and the **chmaster** command if errors are encountered.
-I	Directs the **chmaster** 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.
-N	Invokes the **ypinit** command and starts the appropriate daemons. No changes are made to the **/etc/rc.nfs** file.
-O	Overwrites existing maps for this domain.
-o	Prevents the overwriting of NIS maps. This flag is the default.
-P	Starts the **yppasswdd** daemon along with the **ypserv** daemon.
-p	Suppresses the start of the **yppasswdd** daemon. This flag is the default.
-s *HostName* [**,** *HostName*]	Specifies the slave host names for the slave for this master server. The **chmaster** command automatically adds the current host to this list.
-U	Starts the **ypupdated** daemon along with the **ypserv** daemon.
-u	Suppresses the start of the **ypupdated** daemon. This flag is the default.

Examples

To invoke the **ypinit** command to rebuild the NIS maps for the current domain, enter:

```
chmaster -s chopin -O -p -u -B
```

In this example, the `chmaster` command overwrites the existing maps and the **yppasswdd** and **ypupdated** daemons are not started. The host name `chopin` is specified to be a slave server.

Files

/etc/rc.nfs	Contains the startup script for the NFS and NIS daemons.
/var/yp/*domainname*	Contains the NIS maps for the NIS domain.

Related Information

The **mkclient** command, **rmyp** command, **smit** command, **ypinit** command.

The **ypbind** daemon, **yppasswdd** daemon, **ypserv** daemon, **ypupdated** daemon.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

The System Management Interface Tool (SMIT) in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* explains the structure, main menus, and tasks that are done with SMIT.

Network Information Service (NIS) in *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*.

NIS Reference.

chmod Command

Purpose

Changes file modes.

Syntax

To Change File Modes Symbolically

chmod [**-R**] [**-h**] [**-f**] [[**u**] [**g**] [**o**] | [**a**]] { { **-** | **+** | **=** } [**r**] [**w**] [**x**] [**X**] [**s**] [**t**] } { *File ...* | *Directory ...* }

To Change File Modes Numerically

chmod [**-R**] [**-h**] [**-f**] *PermissionCode* { *File ...* | *Directory ...* }

Description

The **chmod** command modifies the mode bits and the extended access control lists (ACLs) of the specified files or directories. The mode can be defined symbolically or numerically (absolute mode).

When a symbolic link is encountered and you have not specified the **-h** flag, the **chmod** command changes the mode of the file or directory pointed to by the link and not the mode of the link itself. If you specify the **-h** flag, the **chmod** command prevents this mode change.

If you specify both the **-h** flag and the **-R** flag, the **chmod** command descends the specified directories recursively, and when a symbolic link is encountered, the mode of the file or directory pointed to by the link is not changed.

Flags

-f Suppresses all error reporting except invalid permissions and usage statements.

-h Suppresses a mode change for the file or directory pointed to by the encountered symbolic link.

> **Note:** This behavior is slightly different from the behavior of the **-h** flag on the **chgrp** and **chown** commands because mode bits cannot be set on symbolic links.

-R Descends only directories recursively, as specified by the pattern *File...*|*Directory...*. The **-R** flag changes the file mode bits of each directory and of all files matching the specified pattern. See Example 6.

When a symbolic link is encountered and the link points to a directory, the file mode bits of that directory are changed but the directory is not further traversed.

Symbolic Mode

To specify a mode in symbolic form, you must specify three sets of flags.

> **Note:** Do not separate flags with spaces.

The first set of flags specifies who is granted or denied the specified permissions, as follows:

u File owner.

g Group and extended ACL entries pertaining to the file's group.

o All others.

a User, group, and all others. The **a** flag has the same effect as specifying the **ugo** flags together. If none of these flags are specified, the default is the **a** flag and the file creation mask (umask) is applied.

The second set of flags specifies whether the permissions are to be removed, applied, or set:

- Removes specified permissions.
+ Applies specified permissions.
= Clears the selected permission field and sets it to the permission specified. If you do not specify a permission following **=**, the **chmod** command removes all permissions from the selected field.

The third set of flags specifies the permissions that are to be removed, applied, or set:

r Read permission.
w Write permission.
x Execute permission for files; search permission for directories.
X Execute permission for files if the current (unmodified) mode bits have at least one of the user, group, or other execute bits set. The **X** flag is ignored if the *File* parameter is specified and none of the execute bits are set in the current mode bits.

Search permission for directories.
s Set-user-ID-on-execution permission if the **u** flag is specified or implied. Set-group-ID-on-execution permission if the **g** flag is specified or implied.
t For directories, indicates that only file owners can link or unlink files in the specified directory. For files, sets the **save-text** attribute.

Numeric or Absolute Mode

The **chmod** command also permits you to use octal notation for the mode. The numeric mode is the sum of one or more of the following values:

4000 Sets user ID on execution.
2000 Sets group ID on execution.
1000 Sets the link permission to directories or sets the **save-text** attribute for files.
0400 Permits read by owner.
0200 Permits write by owner.
0100 Permits execute or search by owner.
0040 Permits read by group.
0020 Permits write by group.
0010 Permits execute or search by group.
0004 Permits read by others.
0002 Permits write by others.
0001 Permits execute or search by others.

Notes:

1. Specifying the mode numerically disables any extended ACLs. Refer to ″Access control Lists″ in *Operating system and device management* for more information.
2. Changing group access permissions symbolically also affects the AIXC ACL entries. The group entries in the ACL that are equal to the owning group of the file are denied any permission that is removed from the mode. Refer to ″Access control Lists″ in *Operating system and device management* for more information.
3. You can specify multiple symbolic modes separated with commas. Operations are performed in the order they appear from left to right.
4. You must specify the mode symbolically or use an explicit 4-character octal with a leading zero (for example, 0755) when removing the set-group-ID-on-execution permission from directories.
5. For a non-AIXC ACL associated file system object, any request (either symbolically or numerically) that results in a operation to change the base permissions bits (rwxrwxrwx) in mode bits results in replacement of the existing ACL with just the mode bits.

Exit Status

This command returns the following exit values:

0	The command executed successfully and all requested changes were made.
>0	An error occurred.

Security

Access Control

This program should be installed as a normal user program in the Trusted Computing Base.

Only the owner of the file or the root user can change the mode of a file.

RBAC Disclaimer

Examples

1. To add a type of permission to several files:

   ```
   chmod  g+w  chap1  chap2
   ```

   This adds write permission for group members to the files `chap1` and `chap2`**.**

2. To make several permission changes at once:

   ```
   chmod  go-w+x  mydir
   ```

   This denies group members and others the permission to create or delete files in `mydir` (**go-w**) and allows group members and others to search `mydir` or use it in a path name (**go+x**). This is equivalent to the command sequence:

   ```
   chmod  g-w  mydir
   chmod  o-w  mydir
   chmod  g+x  mydir
   chmod  o+x  mydir
   ```

3. To permit only the owner to use a shell procedure as a command:

   ```
   chmod  u=rwx,go= cmd
   ```

   This gives read, write, and execute permission to the user who owns the file (**u=rwx**). It also denies the group and others the permission to access `cmd` in any way (**go=**).

   If you have permission to execute the `cmd` shell command file, then you can run it by entering:

   ```
   cmd
   ```

   Note: Depending on the **PATH** shell variable, you may need to specify the full path to the `cmd` file.

4. To use Set-ID Modes:

   ```
   chmod  ug+s  cmd
   ```

   When the `cmd` command is executed, the effective user and group IDs are set to those that own the `cmd` file. Only the effective IDs associated with the child process that runs the `cmd` command are changed. The effective IDs of the shell session remain unchanged.

   This feature allows you to permit access to restricted files. Suppose that the `cmd` program has the Set-User-ID Mode enabled and is owned by a user called `dbms`. The user `dbms` is not actually a person, but might be associated with a database management system. The user `betty` does not have permission to access any of `dbms`'s data files. However, she does have permission to execute the `cmd` command. When she does so, her effective user ID is temporarily changed to `dbms`, so that the `cmd` program can access the data files owned by the user `dbms`.

This way the user `betty` can use the `cmd` command to access the data files, but she cannot accidentally damage them with the standard shell commands.

5. To use the absolute mode form of the **chmod** command:

   ```
   chmod  644  text
   ```

   This sets read and write permission for the owner, and it sets read-only mode for the group and others. This also removes all extended ACLs that might be associated with the file.

6. To recursively descend directories and change file and directory permissions given the tree structure:

   ./dir1/dir2/file1

   ./dir1/dir2/file2

   ./dir1/file1

   enter this command sequence:

   ```
   chmod -R 777 f*
   ```

   which will change permissions on ./dir1/file1.

   But given the tree structure of:

   ./dir1/fdir2/file1

   ./dir1/fdir2/file2

   ./dir1/file3

   the command sequence:

   ```
   chmod -R 777 f*
   ```

   will change permissions on:

   ./dir1/fdir2

   ./dir1/fdir2/file1

   ./dir1/fdir2/file2

   ./dir1/file3

File

/usr/bin/chmod	Contains the **chmod** command .

Related Information

The **acledit** command, **aclget** command, **aclput** command, **chown** command, **chgrp** command, **ls** command.

The **chmod** subroutine, **fchmod** subroutine.

File ownership and user groups in *Operating system and device management* introduces file ownership and permissions to access files and directories.

Security describes system security.

Installing and Configuring the Trusted Computing Base in *Security*.

chmp Command

Purpose

Changes the characteristics of a mirror pool.

Syntax

chmp -A [**-c** *aiocachelvname*] [**-h** *highwatermark*] **-m** *mirrorpoolname vgname*

chmp -h *highwatermark* **-m** *mirrorpoolname vgname*

chmp -S [**-f**] **-m** *mirrorpoolname vgname*

Description

The **chmp** command can perform the following operations:

- Configure a mirror pool for asynchronous mirroring using the **-A** flag.
- Set the high watermark of the I/O-cache logical volume with the **-h** flag.
- Detect a change in size of the I/O-cache logical volume and take appropriate actions.
- Change the mirror pool from asynchronous mirroring to synchronous mirroring with the **-S** flag.
- Change the I/O-cache logical volume that is used for asynchronous mirroring.

Notes:

1. All disks in all mirror pools must be accessible to be configured for asynchronous mirroring.
2. After a mirror pool is configured for asynchronous mirroring, some active disks are needed from each mirror pool to convert the mirror pool from asynchronous mirroring to synchronous mirroring. If you want to remove one or more mirror pools from a site that is down, disable asynchronous mirroring using the **chmp** command with the **-S** and **-f** flags.
3. Asynchronous mirroring is only supported on nonconcurrent scalable volume groups with mirror pools set to be super strict.
4. You must disable the auto-on and bad-block relocation options of the volume group.
5. The volume group cannot be a snapshot volume group. The volume group cannot contain active paging-space logical volumes.
6. The volume group must be varied on to make mirror pool changes.

Flags

-A	Configures a mirror pool for asynchronous mirroring.
-c *aiocachelvname*	Specifies the name of an asynchronous I/O-cache logical volume. The logical volume must be of the **aio_cache** type and must not reside in the mirror pool that is specified with the **-m** flag. If you do not specify the **-c** flag, the **chmp** command attempts to find the appropriate logical volume of the **aio_cache** type.
-f	Forces a mirror pool from asynchronous mirroring to synchronous mirroring, even if the remote I/O cache is not accessible.
-h *highwatermark*	Specifies the I/O-cache high watermark. The value is the percent of I/O cache size. The default value is 100%. The flag also detects an increase in size of the I/O-cache logical volumes and takes the appropriate action.
-m *mirrorpoolname*	Specifies the mirror pool name.
-S	Changes a mirror pool from asynchronous mirroring to synchronous mirroring.

Parameters

vgname	Specifies the volume group name where the mirror pool resides.

Examples

1. To set up an asynchronous mirroring for the mirror pool, enter the following sequence of commands:
   a. Create a scalable volume group with the mirror pool set to be super strict on local disks `hdisk1`, `hdisk2`, and `hdisk3`:

```
mkvg -f -S -M s -y gmvg1 hdisk1 hdisk2 hdisk3
```

b. Disable the volume group auto-on and bad-block relocation:

```
chvg -a n -b n gmvg1
```

c. Add the local disks into mirror pool MP1:

```
chpv -p MP1 hdisk1 hdisk2 hdisk3
```

d. Create a logical volume for user data:

```
mklv -b n -p copy1=MP1 -y user_data_lv gmvg1 10
```

e. Add the remote physical-volume devices hdisk4, hdisk5, and hdisk6 to mirror pool MP2 in volume group gmvg1:

```
extendvg -f -p MP2 gmvg1 hdisk4 hdisk5 hdisk6
```

f. Add the remote mirror copy in the volume group using the **mirrorvg** command:

```
mirrorvg -c 2 -p copy2=MP2 gmvg1
```

g. Add a logical volume of the **aio_cache** type in the local mirror pool:

Note: A mirror pool can contain only one I/O-cache logical volume. If the I/O-cache logical volume is mirrored, each copy must be in a local mirror pool.

```
mklv -t aio_cache -p copy1=MP1 -y mp1_aiolv gmvg1 1
```

h. Set up asynchronous mirroring for mirror pool MP2:

```
chmp -A -c mp1_aiolv  -h 80 -m MP2 gmvg1
```

2. To change the mirror pool from asynchronous mirroring to synchronous mirroring, enter the following command:

```
chmp -S -m MP2 gmvg1
```

3. To change the mirroring attributes, such as high watermark, enter the following command:

```
chmp -h 90 -m MP2 gmvg
```

4. To replace the I/O-cache logical volume with a different I/O-cache logical volume, enter the following sequence of commands:

a. Change the mirror pool from asynchronous mirroring to synchronous mirroring:

```
chmp -S -m MP2 gmvg1
```

b. Remove the current I/O-cache logical volume mp1_aiolv that resides in mirror pool MP1:

```
rmlv mp1_aiolv
```

c. Create a new I/O-cache logical volume in mirror pool MP1:

```
mklv -t aio_cache -p copy1=MP1 -y mp1_new_aiolv gmvg1 1
```

d. Set up asynchronous mirroring for mirror pool MP2 using the new I/O-cache logical volume:

```
chmp -A -c mp1_new_aiolv -h 90 -m MP2 gmvg1
```

Related Information

The **lsmp** command.

HACMP/XD for Geographic LVM: Planning and Administration Guide provides information about asynchronous mirroring of geographic LVM.

Mirror Pools in *Operating system and device management*.

chnamsv Command

Purpose

Changes TCP/IP-based name service configuration on a host.

Syntax

chnamsv [**-a***″Attribute=Value ...″* | **-A** *FileName*]

Description

The **chnamsv** high-level command changes a TCP/IP-based name service configuration on a host. The command changes the **/etc/resolv.conf** file only. The command does not change the name server database.

If you change the name service configuration for a client, the **chnamsv** command calls the **namerslv** low-level command to change the **resolv.conf** configuration file appropriately.

You can use the Network application in Web-based System Manager (wsm) to change network characteristics. You could also use the System Management Interface Tool (SMIT) **smit namerslv** fast path to run this command.

Flags

-A *FileName*
: Specifies name of file containing the named server initialization information.

-a*″Attribute=Value...″*
: Specifies a list of attributes and their corresponding values to be used for updating the named server initialization files in the database.

  Attributes can be either of the following:

  domain
  : The domain name of the changed named server

  nameserver
  : The Internet address of the changed name server

Examples

1. To update the named server initialization files, enter the command in the following format:

   ```
   chnamsv    -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 update name server initialization files according to information in another file, enter the command in the following format:

   ```
   chnamsv    -A   namsv.file
   ```

   In this example, the file that contains the updated information is `namsv.file`.

Files

/etc/resolv.conf Contains DOMAIN name server information for local resolver routines.

Related Information

The **namerslv** command.

TCP/IP name resolution in *Networks and communication management*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

TCP/IP reference in *Networks and communication management*.

---

chndaf Command

Purpose

Changes the configuration of the AIX Network Data Administration Facility (NDAF).

Syntax

/usr/sbin/chndaf [**-I** | **-B** | **-N**] [*parameter*=*value*]

Description

The **chndaf** command modifies the parameters used by the **dms** and the **dmadm** daemons. Depending on the flags that you specify, the changes take place at different times. You can save the changes in the **/etc/rc.ndaf** startup script for subsequent restarts.

Flags

-B Temporarily stops the daemons currently running on the system, modifies the **/etc/rc.ndaf** startup script with the new parameters, and restarts the daemons with the indicated parameters. This flag is the default.

-I Modifies the **/etc/rc.ndaf** script so that the specified parameters run when the daemons restart.

-N Temporarily stops the daemons currently running on the system and restarts the daemons with the indicated parameters.

Parameters

You can specify one or more of the following optional parameter values:

-admin_serv=yes | **no**

yes Specifies an NDAF administration server. Both the **dms** and **dmadm** daemons are started.

no Specifies an NDAF data server. Only the **dms** daemon is started.

-rpc_timeout=*val* Sets the timeout for an RPC connect or call. The default value is 300 seconds.

-log_level=*val* Sets the level of logging for the log files. The default value is **notice**. You can specify the following values:

- **critical**
- **error**
- **warning**
- **notice**
- **information**

-security=val Sets the type of security method that is used. The default value is **krb5**. You can specify the following values:

auth_sys User ID and group ID (UID/GID) authentication

krb5 Kerberos authentication

krb5i Kerberos integrity authentication

krb5p Kerberos privacy authentication

-krb5_principal=*val* Sets the Kerberos principal that is used for the **kinit** command, with which you can renew your credentials.

-admin_port=*val* Sets the **dmadm** port waiting for the **dmf** remote procedure call (RPC). The default value is 28000.

-serv_port=*val* Sets the **dms** port waiting for the **dmadm** RPC. The default value is 28001.

-admin_cb_port=*val* Sets the **dmadm** port waiting for the **dms** RPC callbacks. The default value is 28002.

-serv_serv_port=*val* Sets the **dms** port waiting for the **dms** RPC. The default value is 28003.

-ndaf_dir=*val* Sets the base directory for NDAF. By default, it contains databases, logs and also the data sets and replicas that are created with no specific path. If you do not specify the **-ndaf_dataset_default** value, data sets are placed here by default. If you do not specify the **-ndaf_replica_default** value, replicas are placed here by default. The following subdirectories are created:

${ndaf_dir}/dsets
If you do not specify the **-ndaf_dataset_default** parameter, the base directory contains data sets that are created without a path where they must be placed.

${ndaf_dir}/replicas
If you do not specify the **-ndaf_replica_default** parameter is not specified, the base directory contains replicas that are created without a path where they must be placed.

${ndaf_dir}/log
If you do not specify the **-ndaf_log_dir** parameter, the base directory contains log files for the **dms** and the **dmadm** daemons.

${ndaf_dir}/admin
The base directory contains the administration databases.

${ndaf_dir}/server
The base directory contains the data server databases.

Requirement: You must specify the **-ndaf_dataset_default** and **-ndaf_replica_default** parameters or you must specify the **-ndaf_dir** parameter. You must have previously enabled the creation of cells, data sets, and replicas, using the **dms_enable_fs** command, on the file systems containing the specified directories to store the data sets and replicas.

-ndaf_dataset_default=*val* Sets the default directory for data sets. The default is **${ndaf_dir}/dsets**.

Requirement: You must specify the **-ndaf_dataset_default** and **-ndaf_replica_default** parameters or you must specify the **-ndaf_dir** parameter. You must have previously enabled the creation of cells, data sets, and replicas, using the **dms_enable_fs** command, on the file systems containing the specified directories to store the data sets and replicas.

-ndaf_replica_default=*val* Sets the default directory for replicas. The default is **${ndaf_dir}/replicas**.

Requirement: You must specify the **-ndaf_dataset_default** and **-ndaf_replica_default** parameters or you must specify the **-ndaf_dir** parameter. You must have previously enabled the creation of cells, data sets, and replicas, using the **dms_enable_fs** command, on the file systems containing the specified directories to store the data sets and replicas.

-ndaf_log_dir=*val* Sets the directory for log files. The default is **${ndaf_dir}/log**.

-krb5_keytab=*val* Indicates the Kerberos keytab path. If you do not specify the parameter and the system resource controller (SRC) is not in use, the keytab is defined either by the KRB5_KTNAME environment variable, or by the default as specified in the **/etc/krb5/krb5.conf** file (when the KRB5_KTNAME variable is not set). If you do not specify the parameter but the SRC is in use, the keytab is always the default as specified in the **/etc/krb5/krb5.conf** file.

-nfs_args=*val* Specifies arguments to be used when data sets are exported by NDAF using NFS. The NFS arguments are formatted exactly as they are for the **exportfs** command. For more information about how to use the parameter, see 2 on page 445.

Examples

1. To configure this system as an NDAF administration server, use the following command:

   ```
   chndaf -admin_serv=yes -ndaf_dir=/var/dmf \
   -ndaf_dataset_default=/ndafpool/dset \
   -ndaf_replica_default=/ndafpool/replica -I
   ```

   This change is made for the restart of the next daemons.

   For an example of how to configure a data server, refer to Configuring an NDAF data server.

2. To restrict exports to **krb5p** only, use the following command:

   ```
   chndaf -nfs_args=sec=krb5p
   ```

   If you specify the NFS version number in the *vers*=*stanza* form, you must include version 4; otherwise, NDAF does not work correctly. If you do not specify the version number, the file systems are exported for NFSv4 only. To export for versions 3 and 4, use the following command:

   ```
   chndaf -nfs_args=vers=3:4
   ```

   .

Security

RBAC Disclaimer

Related Information

The **mkndaf**, **rmndaf**, **lsndaf**, and **dms** commands.

The **dmadm** daemon.

The NDAF installation and configuration in AIX Network Data Administration Facility.

The System Resource Controller in *Operating system and device management*.

---

chnlspath Command

Purpose

Modify the value of the secure **NLSPATH** system configuration variable.

Syntax

chnlspath [**-p**] *NlspathValue*

Description

The **chnlspath** command is used to modify the secure **NLSPATH** system configuration variable.

Flags

-p *NlspathValue*	Specifies the path that the secure **NLSPATH** system configuration variable is set to. In this flag, the **-p** flag is optional.

Security

RBAC Disclaimer

Related Information

The **lsnlspath** command.

---

chnfs Command

Purpose

Changes the configuration of the system to invoke a specified number of **nfsd** daemons or to change NFS global configuration values.

Syntax

chnfs [**-b** *NumberofBiod*] [**-n** *NumberOfNfsd*] [**-l** *NumberOfLockd*] [**-I** | **-B** | **-N**] [**-s** | **-S**] [**-v** | **-V**] [**-r** *v4_root_node*] [**-p** *v4_public_node*] [**-L** *v4_lease_time*] [**-R** {**on**|**off**|*host*[+*host*]}] [**-g on** | **off**] [**-x** *xtend_cnt*] [**-P** *SS_pathname*] [**-G add** | **remove**]

Description

The **chnfs** command invokes the number of **nfsd** daemons specified. The **chnfs** command does this by changing the objects in the SRC database. The **chnfs** command also is used to enable or disable the use of advanced security methods by NFS or to enable or disable the use of NFS Version 4. These changes take place at different times depending on the flags chosen.

Note: The **chnfs** command does not change the number of **biod** threads. To change the number of **biod** threads, use the NFS-specific **-o biods=***n* option of the **mount** command. For example, to specify that an NFS mount use 16 **biod** threads, type:

```
mount -obiods=16 server:/tmp /mnt
```

By default, a v2 **mount** uses 7 **biod** threads, and a v3 **mount** and a v4 **mount** use 32 **biod** threads.

Flags

-B	Temporarily stops the daemons currently running on the system, modifies the SRC database code to reflect the new number, and restarts the daemons indicated. This flag is a default.
-b *NumberofBiod*	Specifies the number of biod threads on the client. This option has no effect and should not be used.
-G add | **remove**	Controls the NFSv4 Grace Period bypass. When `add` is specified as the value of this flag, the grace period is bypassed regardless of how the **-g** option is specified and the **-gpbypass** flag is added to the **nfsd** argument. When `remove` is specified as the value of this flag, the **-gpbypass** flag is removed from the **nfsd** argument.
-g onloff	Controls the NFSv4 Grace Period enablement. The possible values are `on` or `off`. When no **-g** option is specified, the grace period is disabled by default.
-I	Changes the objects in the SRC database so that the number of daemons specified will be run during the next system restart.
-L *v4_lease_time*	Specifies the lease time that the state manager uses when granting a lock to a client. This flag sets the NFS Version 4 lease time in seconds. The lease time also affects the length of the grace period, the time when a client is deemed dead or expired, and the duration of time that a client has before getting timed out. The valid range is from 10 to 600 seconds. The default value is 120 seconds. This flag is valid only for NFS Version 4.
-l *NumberOfLockd*	Specifies the number of **lockd** daemons to run on the system.
-N	Temporarily stops the daemons currently running on the system and restarts the number of daemons indicated.
-n *NumberOfNfsd*	Specifies the number of **nfsd** daemons to run on the system.

P *SS_pathname*	Specifies the location for stable storage. If grace period is enabled, the state manager begins logging in state information in this pathname. If the filesystem is small, the state manager also allocates space initially. The default location for the stable storage pathname is **/var/adm/nfsv4**.
-p *v4_public_node*	Changes the NFS Version 4 public directory to the specified directory. The directory must be a subdirectory of the root directory. The public directory cannot be changed if any directories are currently exported for Version 4 use.
-R {**on**|**off**|*host*[+*host*]}	Enables or disables NFS Version 4 replication. If replication is enabled, replica locations can be specified for Version 4 exports. If replication is not enabled, attempts to export a directory with replica locations will fail. If any directories are exported for NFS Version 4 use, the replication mode cannot be changed. Changing the replication mode of the NFS server can cause errors on clients holding filehandles issued under the previous replication mode. If the *host*[+*host*] form is used, replication is enabled and the host list is used as the replica locations for the **nfsroot**.
-r *v4_root_node*	Changes the NFS Version 4 root location to the specified directory. Version 4 clients that mount **/** will see the specified directory as the server's root. The public directory cannot be changed if any directories are currently exported for Version 4 use.
-S	Enable **RPCSEC_GSS**. This enables NFS to use the enhanced security offered by **RPCSEC_GSS**, such as Kerberos 5.
-s	Disable **RPCSEC_GSS**. This disables the use of **RPCSEC_GSS** methods by NFS.
-V	Enable NFS Version 4.
-v	Disable NFS Version 4.
-x *xtend_cnt*	Controls the NFSv4 Grace Period automatic extension. The *extend_cnt* parameter specifies the total number of automatic extensions allowed for the grace period. If no **-x** 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.

Examples

To set the number of **nfsd** daemons to 10, enter:

```
chnfs    -n    10    -I
```

This change will be made for the next system restart.

Security

RBAC Disclaimer

Related Information

The **exportfs** command, **mknfs** and **rmnfs** commands.

The **nfsd**, **nfsrgyd**, and **gssd** daemons.

Network File System (NFS) Overview for System Management in *Networks and communication management*.

NFS Installation and Configuration in *Networks and communication management*.

List of NFS commands in *Networks and communication management*.

System Resource Controller in *Operating system and device management*.

chnfsdom Command

Purpose
Displays or changes the local NFS domain.

Syntax
chnfsdom [*LocalDomain*]

Description
The **chnfsdom** command changes the local NFS domain of the system. The local NFS domain is stored in the **/etc/nfs/local_domain** file. If no argument is specified, the command displays the current local NFS domain.

Parameters
LocalDomain The new domain name.

Security
Users must have root authority.

RBAC Disclaimer

Files
/etc/nfs/local_domain Stores the local NFS domain name.

Related Information
The **nfsrgyd** command, the **chnfsrtd** command, and the **chnfssec** command.

chnfsexp Command

Purpose
Changes the options used to export a directory to NFS clients.

Syntax
/usr/sbin/chnfsexp -d *Directory* [**-V** *ExportedVersion*] [**-f** *Exports_file*] [**-e** *ExternalName*] [**-t** { **rw** | **ro** | **remove** } {**rm -h** *HostName* [**,***HostName ...*] }] [**-a** *UID*] [**-r** *HostName* [**,** *HostName ...*]] [**-c** *HostName* **,** *HostName ...*]] [**-D** {yes | no}] [**-s** | **-n**] [**-S** *flavor*] [**-G** *rootpath@host*[**+***host*][**:***rootpath@host*[**+***host*]]] [**-g** *rootpath@host*[**+***host*][**:***rootpath@host*[**+***host*]]] [**-o** *Ordering*] [**-x**] [**-X**] [**-I** | **-B** | **-N**] [**-P** | **-p**] [**-v** *number* [**,** *number ...*]]

Description
The **chnfsexp** command takes as a parameter a directory that is currently exported to NFS clients and changes the options used to export that directory. The options specified on the command line will replace those currently being used.

Flags

-a *UID*
Uses the *UID* parameter 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. The *UID* parameter can be either uid or username.

-B
Updates the entry in the **/etc/exports** file and the **exportfs** command is executed to again export the directory immediately.

-c *HostName* [**,** *HostName*] ...
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 *Directory*
Specifies the exported directory that is to be 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 *ExternalName*
Exports the directory specified by the *ExternalName* parameter. The external name must begin with the **nfsroot** name. This option is useful if you have run the **chnfs -r** command to change root to something other than **/**. See the description of the **/etc/exports** file for a description of the **nfsroot** name. This option applies only to directories exported for access by the NFS version 4 protocol.

-f *Exports_file*
Specifies the full path name of the exports file to use if other than the **/etc/exports** file.

-G
rootpath@host[+*host*][:*rootpath@host*[+*host*]]
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 **nfsroot**. The name `localhost` cannot be used as a *hostname*. The **-G** 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 **-G** option is available only on AIX 5L Version 5.3 with the 5300-03 Recommended Maintenance package 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.

-g
rootpath@host[+*host*][:*rootpath@host*[+*host*]]
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 *rootpaths* 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.

-h *Hostname* [**,** *HostName*] ...
Specifies which hosts have read-write access to the directory. This option is valid only when the directory is exported read-mostly.

-I
Adds an entry in the **/etc/exports** file so that the next time the **exportfs** command is run, usually during system restart, the directory will be exported.

-N
Does not modify the entry in the **/etc/exports** file but the **exportfs** command is run with the correct parameters so that the export is changed.

-n
Does not require client to use the more secure protocol. This flag is the default.

-o *Ordering*
Defines how the alternate locations list is generated from the servers that are specified on the **refer** or **replicas** option of the **exportfs** command. The option applies only to directories exported for access by NFS version 4 protocol. The *Ordering* parameter has the following values:

full All of the servers are scattered to form the combinations of alternate locations.

partial The first location of all combinations is fixed to the first server that is specified on the **refer** or **replicas** option of the **exportfs** command. The remaining locations besides the first location are scattered as if they are scattered using the `scatter=full` method.

none No scatter is to be used. The value can also be used to disable scattering if you previously enabled it.

-P
Specifies that the exported directory is to be a public directory.

-p
Specifies that the exported directory is not a public directory.

-r *HostName* [**,** *HostName*] ...
Gives root users on 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.

-S *flavor*
May be used in conjunction with the **-c**, **-t**, or **-r** options to specify which occurrence of the option to change. 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. If the entry in **/etc/exports** specified by the **-d** option contains a clause of the specified flavor, then that clause is updated to reflect the new parameters. Otherwise, a new **sec=** clause with the specified parameters will be appended to the current options list.

Allowable flavor values are:

sys UNIX authentication.

dh DES authentication.

none Use the anonymous ID if it has a value other than `-1`. Otherwise, a weak auth error is returned.

krb5 Kerberos. Authentication only.

krb5i Kerberos. Authentication and integrity.

krb5p Authentication, integrity, and privacy.

-t *Type*

Specifies one of the following types of mount access allowed to clients:

rw Exports the directory with read-write permission. This is the default.

ro Exports the directory with read-only permission.

remove
You must specify the **-t remove** option with the **-S** *flavor* option. Both the security flavor and the type of mount access (**rw**, **ro**, or **rm**) from the existing NFS export for the specified security flavor are removed.

rm Exports the directory with read-mostly permission. If this type is chosen, the **-h** flag must be used to specify hosts that have read-write permission.

-v *number* [**,** *number ...*]

The directory specified by the **-d** option is made available to clients using the specified NFS versions. Valid values are 2, 3, or 4.

-V *ExportedVersion*

Specifies the version of the exported directory that is to be changed. Valid version numbers are 2, 3 and 4.

-x

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 hostnames are in the replica list, NFSv4 clients might treat the location information as faulty and discard it.

-X

Enables the primary host name to be automatically inserted into the replica list. If you do not specify the primary host name of the server in the replica list, the host name is added as the first replica location.

Examples

1. To change the list of hosts that have access to an exported directory and to make this change occur immediately and upon each subsequent system restart, enter:

   ```
   chnfsexp     -d     /usr     -t     rw     -c     host1,host3,host29,grp3,grp2     -B
   ```

   In this example, the `chnfsexp` command changes the attributes of the `/usr` directory to give read and write permission to the `host1`, `host3`, and `host29` hosts, and the `grp3` and `grp2` netgroups.

2. To change the list of hosts that have access to an exported directory, to specify the path name of the exports file, and to make this change occur immediately and upon each subsequent system restart, enter:

   ```
   chnfsexp     -d     /usr     -t     rw     -c     host1,host3,host29,grp3,grp2
                -f     /etc/exports.other     -B
   ```

   In this example, the `chnfsexp` command changes the attributes of the `/usr` directory to give read and write permission to the `host1`, `host3`, and `host29` hosts: the `grp3` and `grp2` netgroups; and specifies the path name of the exports file as `/etc/exports.other`.

3. To change the version accessibility of the **/common/documents** directory to allow access only to clients using NFS version 4 protocol, enter:

   ```
   chnfsexp -d /common/documents -v 4
   ```

4. To change the root access of the **/common/documents** directory to `client1` and `client2` for clients using krb5 access, enter:

   ```
   chnfsexp -d /common/documents -S krb5 -r client1,client2
   ```

5. To change the options for the **/common/documents** directory that is exported only as version 3, enter the following command:

   ```
   chnfsexp -d /common/documents -V 3 -S krb5
   ```

6. To do a full scatter for the alternate locations specified in **refer** or **replicas** option for the **/common/documents** directory, enter the following command:

   ```
   chnfsexp -d /common/documents -o full
   ```

7. To add a list of alternate replica locations and do a partial scatter for the **/common/doc** directory, enter the following command:

   ```
   chnfsexp -d /common/doc -g /common/doc@s1:/common/doc@s2:/common/doc@s3 -o partial
   ```

Files

/etc/exports	Lists directories the server can export.

Security

RBAC Disclaimer

Related Information

The **exportfs** command, **mknfsexp** command, **rmnfsexp** command.

Network File System (NFS) Overview for System Management in *Networks and communication management*.

List of NFS commands in *Networks and communication management*.

---

chnfsim Command

Purpose

Changes NFS foreign identity mappings.

Syntax

For user and group related foreign identity mappings

chnfsim -a | **-l** | **-s** | **-x -u** | **-g** [**-i** *Identity*] [**-n** *name* **-d** *domain*]

For realm-to-domain mappings

chnfsim -a | **-l** | -x [**-r** *realm* **-d** *domain*]

To configure a system to use EIM

chnfsim -c -a | **-l** | **-x** [**-t** *type* **-h** *hostname*[:*port*] **-e** *EIMdomain* **-f** *EIMsuffix* **-b** *admin_DN* **-w** *admin_password* **-W** *access_password*]

To remove EIM configuration from a system

chnfsim -C

Description

The **chnfsim** command administers NFS foreign identity mappings using the Enterprise Identity Mapping (EIM) layer of an LDAP server. To use this command, the **bos.eim.rte** and **ldap.client** filesets must be installed. Additionally, if the machine is to be the EIM LDAP server, the **ldap.server** fileset must also be installed.

After changing identity mappings on the system, run the **nfsrgyd -f** command to flush the systems' identity cache.

You must first configure a system to use EIM with the **-c** and the **-a** flags before attempting to use any other function. All mapping data are stored and retrieved from the EIM LDAP server.

The **chnfsim** command is used to add, list, and remove an EIM configuration for NFS. The **chnfsim** command is then used to add and remove owner and owner group strings to user and group identities. It can list the identity mappings associated with a user or group, and can search for the mapping identity associated with a name and domain.

The **chnfsim** command is also used to add and remove Kerberos realm to NFS domain mappings, and can list the current realm to domain mappings.

Flags

-a Add operation.
-b Specifies the LDAP administrator distinguished name. The default value is `admin`.
-c Configure operation.
-C Remove EIM configuration.
-d Specify the NFS domain part of a NFS V4 owner string.
-e Specify the EIM domain of the EIM LDAP server used for NFS mapping.
-f Specify the EIM directory suffix of the EIM LDAP server used for NFS mapping.
-g Specify a group-based operation.
-h Specify the hostname and port of the EIM LDAP server used for NFS mapping.
-i Specify the mapping identity. This is a unique string that describes a particular owner or owner group.
-l List operation.
-n Specify the owner or owner group name of a NFS V4 owner string.
-r Specify the Kerberos realm.
-s Search operation.
-t Specify the type of EIM LDAP server.

p | P Primary LDAP server.

s | S Secondary (default) LDAP server.

-u Specify a user-based operation.
-w Specify the EIM administrator password.
-W Specify the EIM access-only user password.
-x Remove operation.

Action Matrix

Operation	Flags (Optional flags in parentheses)
-c	Displays current EIM configuration of the system.
	-a -t -h -e -f -w (-b -W) Configures the system for EIM use. The **-w** flag is required if the specified *hostname* is the local system. If the *hostname* is not the local system, at least one of the **-w** or the **-W** flag must be specified. The NFS client or server can be configured for more than one EIM LDAP replica server.
	-l -h Lists the configuration details of the server *hostname*[:*port*] from the configuration file.
	-x -h Deletes the configuration details of the server *hostname*[:*port*] from the configuration file.

-a	**-u -i (-n -d)** Adds the user mapping identity. If the **-n** and **-d** flags are specified, that identity mapping is associated to the user mapping identity. **-g -i (-n -d)** Adds the group mapping identity. If the **-n** and **-d** flags are specified, that identity mapping is associated to the group mapping identity. **-r -d** Adds a realm-to-domain mapping.
-x	**-u -i (-n -d)** Removes the user mapping identity. If the **-n** and **-d** flags are specified, only that identity mapping is removed from the user mapping identity **-g -i (-n -d)** Removes the group mapping identity. If the **-n** and **-d** flags are specified, only that identity mapping is removed from the group mapping identity **-r -d** Removes a realm-to-domain mapping.
-l	Lists all realm-to-domain mappings. **-u -i** Lists all identity mappings associated with the specified user mapping identity. **-g -i** Lists all identity mappings associated with the specified group mapping identity.
-s	**-u -n -d** Searches for user mapping identities associated with the specified identity mapping. **-g -n -d** Searches for group mapping identities associated with the specified identity mapping.
-C	Removes all of the EIM LDAP server entries from the configuration file.

Exit Status

0 Request was successful.

EACCES
Not enough permissions to access data.

ENOENT
The mapping identity, name, domain, or realm was not found in the database; or the configuration file was not found.

EBUSY
EIM server is unable to allocate internal objects.

ECONVERT
Data conversion error.

EINVAL
Input parameter was not valid.

ENOMEM
Unable to allocate memory.

ENOTCONN
LDAP connection has not been made.

EUNKNOWN
Unknown exception occurred.

Examples

1. To display the current EIM configuration for NFS, use the following command:

   ```
   chnfsim -c
   ```

2. To configure a system to use EIM for NFS foreign identity mapping, use the following command:

   ```
   chnfsim -c -a -t P -h foos.com -e nfs -f nfseim -w mypasswd -W access_passwd
   ```

   Note: If the *hostname* specified is the local system, the **chnfsim** command also sets up an LDAP server to run EIM.

3. To configure a client system to use EIM for NFS foreign identity mapping, use the following command:

   ```
   chnfsim -c -a -t P -h foos.com -e nfs -f nfseim -W access_passwd
   ```

   Note: This configures the client with the primary LDAP server (for read-only access). Here, the specified host name is not the local system.

4. To list the configuration details of a server from the configuration file, use the following command:

   ```
   chnfsim -c -l -h foos.com:1080
   ```

5. To delete the configuration details of a server from the configuration file, use the following command:

   ```
   chnfsim -c -x -h foos.com:1080
   ```

6. To add a user identity mapping that specifies ″John Doe″ to ″jdoe@com.com″, use the following command:

   ```
   chnfsim -a -u -i "John Doe" -n jdoe -d com.com
   ```

   Note: This command will create an EIM identity for ″John Doe″ if one does not already exist.

7. To remove the user identity mapping that specifies ″John Doe″ to ″jdoe@com.com″, use the following command:

   ```
   chnfsim -x -u -i "John Doe" -n jdoe -d com.com
   ```

8. To remove all identity mappings for the user ″John Doe″, use the following command:

   ```
   chnfsim -x -u -i "John Doe"
   ```

9. To list all identity mappings for the user ″John Doe″, use the following command:

   ```
   chnfsim -l -u -i "John Doe"
   ```

10. To add a realm-to-domain mapping that specifies ″realm1″ maps to ″domain1″, use the following command:

    ```
    chnfsim -a -r realm1 -d domain1
    ```

11. To remove the realm-to-domain mapping that specifies ″realm1″ maps to ″domain1″, use the following command:

    ```
    chnfsim -x -r realm1 -d domain1
    ```

12. To list all realm-to-domain mappings, use the following command:

    ```
    chnfsim -l
    ```

13. To search for the user mapping identity associated with ″jdoe@com.com″, use the following command:

    ```
    chnfsim -s -u -n jdoe -d com.com
    ```

14. To remove all EIM configuration from a system, use the following command:

    ```
    chnfsim -C
    ```

    Note: This does not remove the underlying LDAP database or entries.

Files

/usr/sbin/chnfsim Location of the **chnfsim** command.

Security

RBAC Disclaimer

Related Information

The **nfsrgyd** command, and the **chnfsrtd** command.

The **/etc/nfs/realm.map** file.

---

chnfsmnt Command

Purpose

Changes the options used to mount a directory from an NFS server.

Syntax

/usr/sbin/chnfsmnt -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**] [**-l** | **-B** | **-N**] [**-r** *TimesToRetry*] [**-R** *NumRetrans*] [**-b** *ReadBufferSize*] [**-c** *WriteBufferSize*] [**-o** *TimeOut*] [**-P** *PortNumber*] [**-u** *AcRegMin*] [**-U** *AcRegMax*] [**-v** *AcDirMin*] [**-V** *AcDirMax*] [**-T** *AcTimeO*] [**-p** *NumBiods*] [**-K** { *any* | *2* | *3* }] [**-k** { *any* | *tcp* | *udp* }] [**-M** *security_methods*] [**-i** { *dio* | *cio* [,*cior*] }]

Description

The **chnfsmnt** command changes the mount options of a currently mounted file system. However, before you can change the attributes of a mount, the **/etc/filesystems** file must contain an entry for the file system. This command unmounts the directory, changes the specified options, and mounts the directory with the new options.

Flags

-A	The **/etc/filesystems** entry for this file system will specify 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.
-B	Modifies the entry in the **/etc/filesystems** file and remounts the file system using the flags and parameters specified. This flag is the default.
-b *ReadBufferSize*	Indicates the size of the read buffer in *N* bytes.
-c *WriteBufferSize*	Indicates the size of the write buffer in *N* bytes.
-d *RemoteDirectory*	Specifies the directory that will be mounted on the path name specified.
-E	Allows keyboard interrupts on hard mounts.
-e	Prevents keyboard interrupts on hard mounts. This flag is the default.
-f *PathName*	Specifies the mount point for the 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.
-H	Makes the mount a hard mount, which causes the client to continue trying until the server responds.
-h *RemoteHost*	Specifies the NFS server that is exporting the directory.
-l	Changes the entry in the **/etc/filesystems** file but does not remount the directory.

-i Specifies I/O mode for the mount. The options are:

dio Specifies direct I/O mode.

cio Specifies concurrent I/O mode.

cior Specifies concurrent I/O with read-only mode.

-J Indicates that **acl**s are used on this mount.

-j Indicates that **acl**s are not used on this mount. This is the default.

-K Specifies the NFS version used for this NFS mount. This flag only applies to AIX 4.2.1 or later. The options are:

any Uses the mount command to determine the correct match, first attempting the highest NFS version available.

2 Specifies NFS Version 2.

3 Specifies NFS Version 3.

-k Specifies the transport protocol used for the mount. The options are:

any Uses the mount command to select the protocol to use. TCP protocol is the preferred protocol.

tcp Specifies the TCP protocol.

udp Specifies the UDP protocol.

-M *security_methods* A list of security methods to use when attempting the mount. A comma separated list of the values `sys`, `dh`, `krb5`, `krb5i`, `krb5p`, which correspond to Unix, DES, Kerberos 5, Kerberos 5 with integrity, and Kerberos 5 with privacy. Multiple values are allowed, but are only meaningful with NFS version 4 mounts. If multiple methods are given for a version 2 or 3 protocol mount, the first method will be used. For a NFS version 4 mount, the methods will be tried in listed order.

-m *MountTypeName* Corresponds to the *type* field in the stanza of the entry in the **/etc/filesystems** file. When the **mount -t** command *MountTypeName* is issued, all of the currently unmounted file systems with a field type equal to the string are mounted.

-N Prevents modification of the corresponding entry in the **/etc/filesystems** file if it exists. If the directory is currently mounted, it is unmounted and then mounted again with the flags and parameters specified.

-n Instructs the mount not to use a more secure protocol. This flag is the default.

-o *TimeOut* Indicates the length of the NFS time out in *N* tenths of a second.

-P *PortNumber* Indicates the IP port number for the server.

-p *NumBiods* Specifies the number of **biod** daemons that are allowed to work on a particular file system. The default 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.

-r *TimeToRetry* Indicates the number of times to retry a mount. The default is 1000.

-R *NumRetrans* Specifies, for a soft mount, the number of times that a request is to be transmitted if it is not acknowledged by the server. If the request goes unacknowledged after *NumRetrans* transmissions, the client gives up on the request. If this flag is not specified, the default value of 3 is used.

-S Makes the mount a soft mount, which means that the system returns an error if the server does not respond.

-s Instructs the mount to use a more secure protocol.

-T*AcTimeO* Sets minimum and maximum time allowed for regular files and directories to *AcTimeO* seconds. If this option is specified, the other cached attribute times are overridden.

-t	Specifies whether the directory will be mounted as 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 *AcRegMax*	Holds cached attributes for no more than *AcRegMax* seconds after file modification.
-u *AcRegMin*	Holds cached attributes for at least *AcRegMin* seconds after file modification.
-V *AcDirMax*	Holds cached attributes for no more than *AcDirMax* seconds after directory update.
-v *AcDirMin*	Holds cached attributes for at least *AcDirMin* seconds after directory update.
-w { **fg** | **bg** }	Indicates whether the mount should be attempted in the foreground (fg) or background (bg). If **bg** is specified and the attempt to mount the directory fails, the mount will be tried again in the background. The **fg** parameter is the default.
-X	Specifies that the server does support long device numbers. This is the default.
-x	Specifies that the server does not support long device numbers.
-Y	Indicates that the execution of **suid** and **sgid** programs are allowed in this file system. This is the default.
-y	Indicates that the execution of **suid** and **sgid** programs is not allowed in this file system.
-Z	Indicates that device access through this mount is allowed. This is the default.
-z	Indicates that device access through this mount is not allowed.

Examples

To change a mount to read-only, enter:

```
chnfsmnt -f /usr/man -d /usr/man -h host1 -t ro
```

In this example, the `chnfsmnt` command changes the attributes of the mounted directory to read-only.

Files

/etc/filesystems	Lists the remote file systems to be mounted during the system restart.

Security

RBAC Disclaimer

Related Information

The **mknfsmnt** command, **mount** command, **rmnfsmnt** command.

How to Mount an NFS File Explicitly in *Networks and communication management*.

List of NFS commands in *Networks and communication management*.

Network File System (NFS) Overview for System Management in *Networks and communication management*.

chnfsrtd Command

Purpose

Changes the local NFS realm-to-domain mappings.

Syntax

chnfsrtd [**-a** *RealmDomain*] [**-e** *OldRealm OldDomain NewRealm NewDomain*] [**-r** *RealmDomain*]

Description

The **chnfsrtd** command administers the local realm-to-domain mappings of the system. The local realm-to-domain mappings are stored in the **/etc/nfs/realm.map** file.

Note: Use the **chnfsdom** command to list the current realm-to-domain mappings.

Flags

-a *RealmDomain*	Adds a new realm-to-domain mapping.
-e *OldRealm OldDomain NewRealm NewDomain*	Edits an existing realm-to-domain mapping.
-r *RealmDomain*	Removes a realm-to-domain mapping.

Security

Users must have root authority to use the **chnfsrtd** command.

RBAC Disclaimer

Examples

1. To add a new realm-to-domain mapping, type:

   ```
   chnfsrtd -a realm1 domain1
   ```

   This commands appends `realm1 domain1` to the **/etc/nfs/realm.map** file.
2. To remove a realm-to-domain mapping, type the following:

   ```
   chnfsrtd -r realm2 domain2
   ```

   This command removes `realm2 domain2` from the **/etc/nfs/realm.map** file, if that mapping exists.
3. To edit an existing realm-to-domain mapping, type:

   ```
   chnfsrtd -e realm3 domain3 realm4 domain4
   ```

   This command changes the `realm3 domain3` mapping to `realm4 domain4` in the **/etc/nfs/realm.map** file, if that mapping exists.

Files

/etc/nfs/realm.map	Stores the local realm-to-domain mappings.

Related Information

The **nfsrgyd** command, the **chnfsdom** command, and the **chnfssec** command.

---

chnfssec Command

Purpose

Changes the default security flavor used by the NFS client

Syntax

chnfssec [**-a** | **-r**] *comma-separated-list*

Description

The **chnfssec** command administers the default security flavors used by the NFS client. These defaults are stored in the **/etc/nfs/security_default** file. Use the **chnfssec** command (without flags) to list the current security flavors.

The valid security flavors available are:

```
sys          Unix style (uids, gids)
dh           DES style (encrypted timestamps)
krb5         Kerberos 5, no integrity or privacy
krb5i        Kerberos 5, with integrity
krb5p        Kerberos 5, with privacy
```

Flags

-a	Sets a new list of security flavors.
-r	Removes a set of security flavors.

Parameters

comma-separated-list	sys, dh, krb5, krb5i, krb5p are the available flavors.

Security

Users must have root authority to use the **chnfssec** command.

RBAC Disclaimer

Examples

1. To add a list of security flavors, type:

   ```
   chnfssec -a krb5,krb5i,sys
   ```

   This command tells the NFS client to first use `krb5`, then `krb5i`, and lastly `sys` security.
2. To remove a security flavor, type the following:

   ```
   chnfssec -r krb5,sys
   ```

   This command removes `krb5` and `sys` from the list of security flavors the NFS client will use.

Files

/etc/nfs/security_default	Stores the default NFS security flavors.

Related Information

The **nfsrgyd** command, the **chnfsdom** command, and the **chnfsrtd** command.

chown Command

Purpose

Changes the owner or group associated with a file.

Syntax

chown [**-f**] [**-h**] [**-R**] *Owner* [**:***Group*] { *File ...* | *Directory ...* }

chown -R [**-f**] [**-H** | **-L** | **-P**] *Owner* [**:***Group*] { *File ...* | *Directory ...* }

Description

The **chown** command changes the owner of the file or directory specified by the *File* or *Directory* parameter to the user specified by the *Owner* parameter. The value of the *Owner* parameter can be a user name from the user database or a numeric user ID. Optionally, a group can also be specified. The value of the *Group* parameter can be a group name from the group database or a numeric group ID.

Only the root user can change the owner of a file. You can change the group of a file only if you are a root user or if you own the file. If you own the file but are not a root user, you can change the group only to a group of which you are a member.

Although the **-H**, **-L** and **-P** flags are mutually exclusive, specifying more than one is not considered an error. The last flag specified determines the behavior that the command will exhibit.

When a symbolic link is encountered and you have not specified the **-h** flag, the **chown** command changes the ownership of the file or directory pointed to by the link and not the ownership of the link itself.

If you specify the **-h** flag, the **chown** command has the opposite effect and changes the ownership of the link itself and not that of the file or directory pointed to by the link.

If you specify the **-R** flag, the **chown** command recursively descends the specified directories.

If you specify both the **-h** flag and the **-R** flag, the **chown** command descends the specified directories recursively, and when a symbolic link is encountered, the ownership of the link itself is changed and not that of the file or directory pointed to by the link.

Flags

-f Suppresses all error messages except usage messages.

-h Changes the ownership of an encountered symbolic link and not that of the file or directory pointed to by the symbolic link.

-H If the **-R** option is specified and a symbolic link referencing a file of type directory is specified on the command line, the **chown** command shall change the user ID (and group ID, if specified) of the directory referenced by the symbolic link and all files in the file hierarchy below it.

-L If the **-R** option is specified and a symbolic link referencing a file of type directory is specified on the command line or encountered during the traversal of a file hierarchy, the **chown** command shall change the user ID (and group ID, if specified) of the directory referenced by the symbolic link and all files in the file hierarchy below it.

-P If the **-R** option is specified and a symbolic link is specified on the command line or encountered during the traversal of a file hierarchy, the **chown** command shall change the owner ID (and group ID, if specified) of the symbolic link if the system supports this operation. The **chown** command shall not follow the symbolic link to any other part of the file hierarchy.

-R Descends directories recursively, changing the ownership for each file. When a symbolic link is encountered and the link points to a directory, the ownership of that directory is changed but the directory is not further transversed. If the **-h**, **-H**, **-L** or **-P** flags are not also specified, when a symbolic link is encountered and the link points to a directory, the group ownership of that directory is changed but the directory is not traversed further.

Exit Status

This command returns the following exit values:

0 The command executed successfully and all requested changes were made.
>0 An error occurred.

Security

Access Control

This program should be installed as a normal user program in the Trusted Computing Base.

RBAC Disclaimer

Examples

1. To change the owner of the file `program.c`:

   ```
   chown jim program.c
   ```

   The user access permissions for `program.c` now apply to `jim`. As the owner, `jim` can use the **chmod** command to permit or deny other users access to `program.c`.
2. To change the owner and group of all files in the directory `/tmp/src` to owner `john` and group `build`:

   ```
   chown -R john:build /tmp/src
   ```

Files

/usr/bin/chown	The **chown** command
/etc/group	File that contains group IDs
/etc/passwd	File that contains user IDs

Related Information

The **chgrp** command, **chmod** command.

The **chown** subroutine, **fchown** subroutine.

The File ownership and user groups in *Operating system and device management* introduces file ownership and permissions to access files and directories.

Security describes system security.

---

chpasswd Command

Purpose

Changes password for users.

Syntax

chpasswd [**-R** *load_module*] [**-e**] [**-f** *flags* | **-c**]

Description

The **chpasswd** command administers users' passwords. The root user can supply or change users' passwords specified through standard input. Each line of input must be of the following format.

```
username:password
```

Only root users can set passwords with this command.

By default, the **chpasswd** command sets the ADMCHG flag for the users. The **-f** option may be used with other valid flags to override the default. The **-c** option clears all password flags.

The password field can be cleartext or a value encrypted with the crypt algorithm. The **-e** option indicates that passwords are of encrypted format. Please note that all passwords in a batch must conform to the same format.

Flags

-c	Clears all password flags.
-e	Specifies that the passwords are of encrypted format.
-f *flags*	Specifies the comma separated list of password flags to set. Valid flag values are: ADMIN, ADMCHG, and/or NOCHECK. Refer to the **pwdadm** command documentation for details about these values.
-R *load_module*	Specifies the loadable I&A module used to change users' passwords.

Security

Access Control

Only root users should have execute (x) access to this command. The command should have the trusted computing base attribute.

RBAC Disclaimer

Examples

1. To set passwords for users from the command line, type:

   ```
   chpasswd
   ```

   Followed by entering `username:password` pairs, one pair per line. Enter CTRL+D when finished.

   ```
   user1:passwd1
   user2:passwd2
   CTRL+D
   ```

2. To set passwords for users contained in a file named **mypwdfile**, type the following:

   ```
   cat mypwdfile | chpasswd
   ```

   Note that **mypwdfile** must contain `username:password` pairs; one pair per line. For example:

   ```
   user1:passwd1
   user2:passwd2
   ...
   ```

Files

Mode	File	Description
	/etc/user/bin/chpasswd	Location of the **chpasswd** command.
rw	**/etc/passwd**	
rw	**/etc/security/passwd**	
r	**/etc/security/user**	

Related Information

The **passwd** and the **pwdadm** commands.

---

chpath Command

Purpose

Changes the operational status of paths to an MultiPath I/O (MPIO) capable device, or changes an attribute associated with a path to an MPIO capable device.

Syntax

chpath -l *Name* **-s** *OpStatus* [**-p** *Parent*] [**-w** *Connection*]

chpath -l *Name* **-p** *Parent* [**-w** *Connection*] [**-P**] **-a** *Attribute=Value* [**-a** *Attribute=Value* ...]

chpath -h

Description

The **chpath** command either changes the operational status of paths to the specified device (the **-l** *Name* flag) or it changes one, or more, attributes associated with a specific path to the specified device. The required syntax is slightly different depending upon the change being made.

The first syntax shown above changes the operational status of one or more paths to a specific device. The set of paths to change is obtained by taking the set of paths which match the following criteria:

- The target device matches the specified device.
- The parent device matches the specified parent (**-p** *Parent*), if a parent is specified.
- The connection matches the specified connection (**-w** *Connection*), if a connection is specified.
- The path status is **PATH_AVAILABLE**.

The operational status of a path refers to the usage of the path as part of MPIO path selection. The value of **enable** indicates that the path is to be used while **disable** indicates that the path is not to be used. It should be noted that setting a path to **disable** impacts future I/O, not I/O already in progress. As such, a path can be disabled, but still have outstanding I/O until such time that all of the I/O that was already in progress completes. As such, if **-s disable** is specified for a path and I/O is outstanding on the path, this fact will be output.

Disabling a path affects path selection at the device driver level. The **path_status** of the path is not changed in the device configuration database. The **lspath** command must be used to see current operational status of a path.

The second syntax shown above changes one or more path specific attributes associated with a particular path to a particular device. Note that multiple attributes can be changed in a single invocation of the **chpath** command; but all of the attributes must be associated with a single path. In other words, you

cannot change attributes across multiple paths in a single invocation of the **chpath** command. To change attributes across multiple paths, separate invocations of **chpath** are required; one for each of the paths that are to be changed.

Flags

-a *Attribute*=*Value*
: Identifies the attribute to change as well as the new value for the attribute. The *Attribute* is the name of a path specific attribute. The *Value* is the value which is to replace the current value for the *Attribute*. More than one instance of the **-a** *Attribute*=*Value* can be specified in order to change more than one attribute.

-h
: Displays the command usage message.

-l *Name*
: Specifies the logical device name of the target device for the path(s) affected by the change. This flag is required in all cases.

-p *Parent*
: Indicates the logical device name of the parent device to use in qualifying the paths to be changed. This flag is required when changing attributes, but is optional when change operational status.

-P
: Changes the path's characteristics permanently in the ODM object class without actually changing the path. The change takes affect on the path the next time the path is unconfigured and then configured (possibly on the next boot).

-w *Connection*
: Indicates the connection information to use in qualifying the paths to be changed. This flag is optional when changing operational status. When changing attributes, it is optional if the device has only one path to the indicated parent. If there are multiple paths from the parent to the device, then this flag is required to identify the specific path being changed.

-s *OpStatus*
: Indicates the operational status to which the indicated paths should be changed. The operational status of a path is maintained at the device driver level. It determines if the path will be considered when performing path selection.The allowable values for this flag are:

  enable Mark the operational status as **enabled** for MPIO path selection. A path with this status will be considered for use when performing path selection. Note that enabling a path is the only way to recover a path from a **failed** condition.

  disable
  Mark the operational status as **disabled** for MPIO path selection. A path with this status will not be considered for use when performing path selection.

  This flag is required when changing operational status. When used in conjunction with the **-a** *Attribute*=*Value* flag, a usage error is generated.

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	The **chpath** command line.

Examples

1. To disable the paths between **scsi0** and the **hdisk1** disk device, enter:

   ```
   chpath -l hdisk1 -p scsi0 -s disable
   ```

   The system displays a message similar to one of the following:

   ```
   paths disabled
   ```

   or

   ```
   some paths enabled
   ```

   The first message indicates that all PATH_AVAILABLE paths from **scsi0** to **hdisk1** have been successfully enabled. The second message indicates that only some of the PATH_AVAILABLE paths from **scsi0** to **hdisk1** have been successfully disabled.

Files

/usr/sbin/chpath	Contains the **chpath** command.

Related Information

The **lspath** command, **mkpath** command, **rmpath** command.

---

chprtsv Command

Purpose

Changes a print service configuration on a client or server machine.

Syntax

chprtsv -c | **-s** [**-d** | **-i**] [**-h***"HostName..."* | **-H** *FileName*] [**-x***"HostName..."* | **-X** *FileName*] [**-q***"QEntry"* **-v** *DeviceName* **-a***"Attribute* **=***Value..."* **-b***"Attribute* **=***Value..."* | **-A** *FileName*]

Description

The **chprtsv** high-level command changes print service configuration on a client or server machine.

To change print service for a client, the **chprtsv** command does the following:

1. Disables the client spool queue with the **chque** and **chquedev** commands.
2. Changes the appropriate entries in the **/etc/qconfig** file with the **chque** and **chquedev** commands.
3. Enables the client spool queue with the **chque** and **chquedev** commands.

To change print service for a server, the **chprtsv** command does the following:

1. Calls the **ruser** low-level command to change remote users configured on the print server, if necessary.
2. Calls the **chque** and **chquedev** commands to change the print queues and entries in the **qconfig** file, if necessary.
3. Calls the SRC **refresh** command to restart the **lpd** and **qdaemon** servers.

If you want to change the attributes of a queue, you must specify the queue name and the attributes associated with the queue. If you want to change the attributes of the queue device, you must specify queue name, queue device name, and the attributes associated with the queue device.

The changes you make with the **chprtsv -i** command go into effect on the system database and on the current active system.

If you want the changes you make to go into effect at system startup time without affecting the current system, use the **chprtsv -d** command to change only TCP/IP and its associated network interfaces in the system database only.

Flags

-A *FileName*
Specifies the name of the file containing **qconfig** command-related entries.

-a *"Attribute* **=***Value..."*
Specifies a list of attributes with corresponding values to be used for updating the spooler's **qconfig** file or object class. The list should be enclosed in quotes. Valid attribute types follow:

acctfile (true/false)
Identifies the file used to save **print** accounting information. The default value of **false** suppresses accounting. If the named file does not exist, no accounting is done.

device Identifies the symbolic name that refers to the device stanza.

discipline
Defines the queue-serving algorithm. The default, **fcfs**, means first come, first served. A value of **sjn** means shortest job next.

host Specifies the name of the host from which to print. (The name of this host must be the same as the name specified by the *HostName* variable.)

l_statfilter
Translates long queue-status information from non-AIX format to AIX format.

s_statfilter
Translates short queue-status information from non-AIX format to AIX format.

up (true/false)
Defines the state of the queue. The default **true** indicates that it is running. A value of **false** indicates that it is not.

-b *"Attribute =Value..."* Specifies a list of attributes with corresponding values for device stanza corresponding values to be used for updating the spooler's **qconfig** file or object class. The list should be enclosed in quotes. Valid attribute types follow:

access (write/both)
Specifies the type of access the backend has to the file specified by the **file** field. The **access** file 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**.

align (true/false)
Specifies whether the backend sends a form-feed control before starting the job if the printer has been idle. The default is **false**.

backend
Specifies the full path name of the backend, optionally followed by the flags and parameters to be passed to it.

feed Specifies the number of separator pages to print when the device becomes idle, or takes a value of **never**, which indicates that the backend is not to print separator pages.

file Identifies the special file where the output of the backend is to be redirected. The default values of **false** indicates no redirection. In this case, the backend opens the output file.

header (never/always/group)
Specifies whether a header page prints before each job or group of jobs. The default is a value of **never** which indicates no header page. To produce a header page before each job, specify a value of **always**. To produce a header before each group of jobs for the same user, specify a value of **group**.

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 a value of **always**. To produce a trailer after each group of jobs for the same user, specify a value of **group**.

-c Specifies to the **chprtsv** command to reconfigure print service for a client machine.

-d Specifies that changes be reflected in the system database only, so that they can take effect at the next system startup.

-H *FileName* Specifies the name of a file containing a list of host names to be included.

-h*"HostName..."* Specifies a list of host names to be included on the current list of remote users who can use the print server. Note that the queuing system does not support multibyte host names.

-i Specifies that the change be reflected not only in the database, but also in the current running system.

-q*"QEntry"* Specifies a **qconfig** file entry to be removed.

-s Specifies that print service reconfiguration is to be performed for a server machine.

-v *DeviceName* Specifies a list of device stanzas to be changed.

-X *FileName* Specifies the name of a file containing a list of host names to be excluded.

-x*"HostName..."* Specifies a list of host names to be excluded on the current list of remote users who can use the print server.

Examples

To reconfigure a print server, specify that the changes will take effect at the next startup, specify the file containing the host names, and then exclude some of those hosts, enter:

```
chprtsv -s -d -H ruser.inc -x "host1,host2,host3"
```

Files

/etc/qconfig	Contains configuration information for the printer queuing system.
/etc/hosts.lpd	Specifies foreign hosts that can print on the local host.

Related Information

The **chque** command, **chquedev** command, **ruser** command.

The **lpd** daemon, **qdaemon** daemon.

TCP/IP reference in *Networks and communication management*.

TCP/IP daemons in *Networks and communication management*.

---

chps Command

Purpose

Changes the attributes of a paging space.

Syntax

chps [**-t** *ps_helper*] [**-s** *LogicalPartitions* | **-d** *LogicalPartitions*] [**-f**] [**-c** *ChecksumSize*] [**-a** { **y** | **n** }] *PagingSpace*

Description

The **chps** command changes the attributes of a paging space. The *PagingSpace* parameter specifies the name of the paging space to be changed.

To change the size of a Network File System (NFS) paging space, the size of the file that resides on the server must first be changed and then the **swapon** command used to notify the client of the change in size of the paging space.

Note: There is a paging space limit of 64 GB per device.

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 **-c** flag to specify **chps** command. The /etc/swapspaces path will be modified accordingly if the helper executable returns zero. The helper executable must change the attributes. If the helper program doesn't exist in the /sbin/helpers/pagespace path, the **chps** 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 Web-based System Manager Devices application (**Devices** fast path) to change device characteristics. You could also use the System Management Interface Tool (SMIT) **smit chps** fast path to run this command.

> **Note:** The primary paging space is hardcoded in the boot record. Therefore, the primary paging space will always be activated when the system is restarted. The **chps** command is unable to deactivate the primary paging space.

Flags

-a Specifies to use a paging space at the next system restart.

- **y** Specifies that the paging space is active at subsequent system restarts.
- **n** Specifies that the paging space is inactive at subsequent system restarts.

-d *LogicalPartitions* Specifies the number of logical partitions to subtract.

-c *ChecksumSize* Specifies the size of the checksum to use for the paging space, in bits. Valid options are 0 (checksums disabled), 8, 16 and 32. If **-c** is not specified, it will default to 0. The **chps** command with this option will fail on a swapped on paging space unless **-f** is used.

-f Specifies that the checksum size set by **-c** will be used for the next swapon of the paging space. This option has no effect if **-c** is not used or if the paging space is not swapped on.

-s *LogicalPartitions* Specifies the number of logical partitions to add.

-t Specifies to use the helper program under /sbin/helpers/pagespace directory.

ps_helper
Name of the helper program for a third party device.

Security

RBAC Disclaimer

Examples

1. To change the size of the `myvg` paging space, enter:

   ```
   chps  -s 4 myvg
   ```

   This adds four logical partitions to the `myvg` paging space.
2. To define the `PS02` paging space as configured and active at subsequent system restarts, enter:

   ```
   chps  -a y PS02
   ```

   This specifies that the `PS02` paging space is to be active at subsequent system restarts.
3. To set the checksum size of the `myvg` paging space to 1 byte, enter:

   ```
   chps  -c 8 mypg
   ```

   This sets the myvg paging space checksum size to 8 bits, if it is not swapped on.
4. To change the size of the `myvg` paging space using helper program foo enter:

   ```
   chps  -t foo –s4 myps
   ```

   This adds four logical partitions to myps by calling the helper program foo.

Files

/etc/swapspaces Specifies the paging space devices and their attributes.

Related Information

The **lsps** command, **mkps** command, **rmps** command, **swap** command, **swapon** command, **swapoff** command.

The Paging space in *Operating system and device management* explains paging space and its allocation policies.

The File systems in *Operating system and device management* provides information on working with files.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

The System Management Interface Tool (SMIT) in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* explains the structure, main menus, and tasks that are done with SMIT.

---

chpv Command

Purpose

Changes the characteristics of a physical volume in a volume group.

Syntax

chpv [**-h** *hotspare*] [**-a** *allocation*] [**-v** *availability*] [**-c**] [**-p** *mirrorpool*] [**-P**] [**-m** *mirrorpool*] *physicalvolume ...* [**-C** *hdiskname*]

Description

Attention: This command is not allowed if the volume group is varied on in concurrent mode.

The **chpv** command changes the state of the physical volume in a volume group by setting allocation permission to either allow or not allow allocation and by setting the availability to either available or removed. This command can also be used to clear the boot record for the given physical volume. Characteristics for a physical volume remain in effect unless explicitly changed with the corresponding flag.

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

You can use the Volumes application in Web-based System Manager (wsm) to change volume characteristics. You can also use the System Management Interface Tool (SMIT) **smit chpv** fast path to run this command.

Flags

-a *allocation*
: Sets the allocation permission for additional physical partitions on the physical volume specified by the *physicalvolume* parameter. Either allows (yes) the allocation of additional physical partitions on the physical volume, or prohibits (no) the allocation of additional physical partitions on the physical volume. The *allocation* variable can be either:

  y Allows the allocation of additional physical partitions on the physical volume.

  n Prohibits the allocation of additional physical partitions on the physical volume. The logical volumes that reside on the physical volume can still be accessed.

-c
: Clears the boot record of the given physical volume.

-C *hdiskname*
: Clears the owning volume manager from a disk. This flag is only valid when running as the root user. This command will fail to clear LVM as the owning volume manager if the disk is part of an imported LVM volume group.

-h *hotspare*
Sets the sparing characteristics of the physical volume so that the physical volume can be used as a hot spare. Also sets the allocation permission for physical partitions on the physical volume specified by the *physicalvolume* parameter. This flag has no meaning for non-mirrored logical volumes. The *hotspare* variable can be either:

y Marks the disk as a hot spare disk within the volume group it belongs to and prohibits the allocation of physical partitions on the physical volume. The disk must not have any partitions allocated to logical volumes to be successfully marked as a hot spare disk.

n Removes the disk from the hot spare pool for the volume group in which it resides and allows allocation of physical partitions on the physical volume.

Note: This flag is not supported for the concurrent capable volume groups.

-m *mirrorpool*
Changes the name of the mirror pool that is assigned to the specified disk to the value of the *mirrorpool* parameter.

-p *mirrorpool*
Assigns the physical volume to a mirror pool. The name of a mirror pool can be up to 15 characters in length. After mirror pools are enabled in a volume group, the volume group can no longer be imported into a version of AIX (before AIX Version 6.1) that does not support mirror pools.

-P
Removes the physical volume from the mirror pool that is being assigned. The physical volume can only be removed from the mirror pool if it has partitions that are allocated to a logical volume where mirror pools are enabled.

-v *availability*
Sets the availability of the physical volume. If you set the availability to closed, logical input and output to the physical volume are stopped. You should close a physical volume when the physical volume is removed from operation. Access to physical volume data by the file system or the virtual memory manager is stopped, but you can continue to use the system management commands. The *availability* variable can be either:

a Makes a physical volume available for logical input and output.

r Makes a physical volume unavailable (removed) for logical input and output. If the physical volume is required in order to maintain a volume group quorum, an error occurs and the physical volume remains open.

Examples

1. To close physical volume `hdisk3`, enter:

   ```
   chpv  -v r hdisk3
   ```

   The physical volume is closed to logical input and output until the **-v a** flag is used.
2. To open physical volume `hdisk3`, enter:

   ```
   chpv  -v a hdisk3
   ```

   The physical volume is now open for logical input and output.
3. To stop the allocation of physical partitions to physical volume `hdisk3`, enter:

   ```
   chpv  -a n hdisk3
   ```

   No physical partitions can be allocated until the **-a y** flag is used.
4. To clear the boot record of a physical volume `hdisk3`, enter:

   ```
   chpv -c hdisk3
   ```

Files

/usr/sbin	Directory where the **chpv** command resides.
/tmp	Directory where temporary files are stored while the command is running.

Related Information

The **lspv** command.

The Logical volume storage in *Operating system and device management* explains the Logical Volume Manager, physical volumes, logical volumes, volume groups, organization, ensuring data integrity, and understanding the allocation characteristics.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

The System Management Interface Tool (SMIT) in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* explains the structure, main menus, and tasks that are done with SMIT.

---

chque Command

Purpose

Changes the queue name.

Syntax

chque -q *Name* [**-a** '*Attribute*=*Value*' ...]

Description

The **chque** command changes the queue name by changing the stanza in the **qconfig** file specified by the **-q** flag. Within that stanza, each attribute that matches one of the *Attribute* **=** *Value* pairs given on the command line will be replaced by the one on the command line. If no match is found, the *Attribute* **=** *Value* pair is added to the end of the stanza. The device attribute cannot be changed.

You can use the Printer Queues application in Web-based System Manager (wsm) to change printer queue characteristics. You could also use the System Management Interface Tool (SMIT) **smit chque** fast path to run this command.

Recommendation: To edit the **/etc/qconfig** file, use the **chque**, **mkque**, **rmque**, **chquedev**, **mkquedev**, and **rmquedev** commands or SMIT. Further, it is recommended to run these commands during slow or off-peak time.

If manual editing of the **/etc/qconfig** file is necessary, you can first issue the **enq -G** command to bring the queuing system and the **qdaemon** to a halt after all jobs are processed. Then you can edit the **/etc/qconfig** file and restart the **qdaemon** with the new configuration.

Flags

-a '*Attribute* **=** *Value*'	Specifies the '*Attribute* **=** *Value*' to be added or replaced by the one entered on the command line. For a list of valid attributes, refer to the **/etc/qconfig** file.
-q *Name*	Specifies the current *Name* of the queue and of the stanza in the **qconfig** file that is to be changed.

Security

RBAC Disclaimer

Examples

To change the name of the host to `fred` for queue `lp0`, enter:

```
chque  -qlp0  -a 'host = fred'
```

Files

/usr/bin/chque	Contains the **chque** command.
/etc/qconfig	Contains the configuration file.

Related Information

The **chquedev** command, **lsque** command, **mkque** command, **rmque** command.

The **qconfig** file.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

Changing / Showing Queue Characteristics in *Printers and printing*.

Printing administration in *Printers and printing*.

Printer-specific information in *Printers and printing*.

Installing support for additional printers in *Printers and printing*.

Print spooler in *Printers and printing*.

Virtual printer definitions and attributes in *Printers and printing*.

Printer colon file conventions in *Printers and printing*.

---

chquedev Command

Purpose

Changes the printer or plotter queue device names.

Syntax

chquedev -q*Name* **-d***Name* [**-a**'*Attribute* **=** *Value*'...]

Description

The **chquedev** command changes the printer or plotter queue device names by changing the device stanza in the **qconfig** file specified by the **-d**, and **-q** flags. Within that stanza, each attribute that matches one of the '*Attribute* **=** *Value*' flags given on the command line is replaced by the one entered on the command line. If no match is found, '*Attribute* **=** *Value*' is added to the end of the stanza.

You can use the Printer Queues application in Web-based System Manager (wsm) to change printer queue characteristics. You could also use the System Management Interface Tool (SMIT) **smit chquedev** fast path to run this command.

Recommendation: To edit the **/etc/qconfig** file, use the **chque**, **mkque**, **rmque**, **chquedev**, **mkquedev**, and **rmquedev** commands or SMIT. Further, it is recommended to run these commands during slow or off-peak time.

If manual editing of the **/etc/qconfig** file is necessary, you can first issue the **enq -G** command to bring the queuing system and the **qdaemon** to a halt after all jobs are processed. Then you can edit the **/etc/qconfig** file and restart the **qdaemon** with the new configuration.

Flags

-a '*Attribute* **=** *Value*'	Specifies the stanza lines to change or add. For a list of valid attributes, see the **qconfig** file.
-d *Name*	Specifies the device *Name* in the queue to be changed.
-q *Name*	Specifies the queue *Name* in which to change the device stanza.

Security

RBAC Disclaimer

Examples

To change the `ps` device stanza on the `lp0` queue to contain the line `'backend = piobe -x -y'`, enter:

```
chquedev  -qlp0  -d ps  -a 'backend        =        piobe -x -y'
```

Note: The **-x** flag and the **-y** flag in this example are flags for the **piobe** command.

Files

/usr/bin/chquedev	Contains the chquedev command.
/etc/qconfig	Contains the configuration file.

Related Information

The **chque** command, **lsquedev** command, **mkquedev** command, **rmquedev** command, **piobe** command.

The **qconfig** file.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

Changing / Showing Queue Characteristics in *Printers and printing*.

Printing administration in *Printers and printing*.

Printer-specific information in *Printers and printing*.

Installing support for additional printers in *Printers and printing*.

Print spooler in *Printers and printing*.

Printer colon file conventions in *Printers and printing*.

chresponse Command

Purpose

Adds or deletes the actions of a response or renames a response.

Syntax

To add an action to a response:

chresponse –a –n *action* [**–d** *days_of_week*[**,***days_of_week...*]]
[**–t** *time_of_day*[**,***time_of_day...*]] [**–s** *action_script*] [**–r** *return_code*]
[**–e a** | **r** | **b**] [**–o**] [**–E** *env_var*=*value*[**,***env_var*=*value...*]]
[**–u**] [**–h**] [**–TV**] *response*[**:***node_name*]

To delete an action from a response:

chresponse –p –n *action* [**–h**] [**–TV**] *response*[**:***node_name*]

To rename a response:

chresponse –c *new_response* [**–h**] [**–TV**] *response*[**:***node_name*]

To unlock or lock a response:

chresponse {**-U** | **-L**} [**–h**] [**–TV**] *response*[**:***node_name*]

Description

The **chresponse** command adds an action to a response or deletes an action from a response. Actions define commands to be run when the response is used with a condition and the condition occurs. The **chresponse** command can also be used to rename a response.

If a particular response is needed for system software to work properly, it may be locked. A locked response cannot be modified or removed until it is unlocked. If the response you specify on the **chresponse** command is locked, it will not be modified; instead an error will be generated informing you that the response is locked. To unlock a response, you can use the **-U** flag. However, since a response is typically locked because it is essential for system software to work properly, you should exercise caution before unlocking it. To lock a response so it cannot be modified, use the **-L** flag.

Flags

–a Adds the action specification to *response*.

–p Deletes *action* from *response*.

–c *new_response*
Specifies a new name to assign to the response. The new name must not already exist. The new name replaces the current name. The *new_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.

–n *action*
Specifies the name of the action. When the **–a** flag is used, this is the name of the action being defined. When the **–p** flag is used, this is the name of the action to be deleted. Action names must be unique within a response. Only one action can be defined at a time.

–d *days_of_week***[,***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

–t *time_of_day***[,***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 is **0000-2400**. If no value is specified but a comma is entered, the default value is used.

–s *action_script*
Specifies the fully-qualified path for the script or command to run for the action being defined. See the **displayevent**, **logevent**, **notifyevent**, and **wallevent** commands for descriptions of predefined response scripts that are provided with the application.

–r *return_code*
Specifies the expected return code for *action_script*. 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 **–r** flag is not specified, the actual return code is written to the audit log, and no comparison is performed.

–e a | **r** | **b**
Specifies the type of event that causes the action being defined to run:

a Specifies an event. This is the default.

r Specifies a rearm event.

b Specifies both an event and a rearm event.

–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.

–E *env_var***=***value***[,***env_var***=***value...***]**
Specifies any environment variables to be set before *action_script* is run. If multiple *env_var*=*value* variables are specified, they must be separated by commas.

–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.

–U Unlocks a response so it can be modified or removed. If a response is locked, this is typically because it is essential for system software to work properly. For this reason, you should exercise caution before unlocking it. When unlocking a response using the **-U** flag, no other operation can be preformed by this command.

–L Locks a response so it cannot be modified or removed. When locking a response using the **-L** flag, no other operation can be performed by this command.

Parameters

response Specifies the name of the response to be changed.

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 of the **chresponse** command needs write permission to the **IBM.EventResponse** resource class on the node where the response is defined. 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. In this example, the action named ″E-mail root″ cannot be the only action. To delete ″E-mail root″ from the response named ″E-mail root anytime″, run this command:

   ```
   chresponse -p -n "E-mail root" "E-mail root anytime"
   ```

2. In this example, the action named ″E-mail root″ will be used Monday through Friday from 8 AM to 6 PM, will use the command **/usr/sbin/rsct/bin/notifyevent root**, will save standard output in the audit log, and will expect return code 5 from the action. To add ″E-mail root″ to the response named ″E-mail root anytime″, run this command:

   ```
   chresponse  -a  -n "E-mail root" -d 2-6 -t 0800-1800 \
   -s "/usr/sbin/rsct/bin/notifyevent root" -o  -r 5  \
   "E-mail root anytime"
   ```

3. To rename the response ″E-mail root anytime″ to ″E-mail root and admin anytime″, run this command:

   ```
   chresponse  -c "E-mail root and admin anytime" "E-mail root anytime"
   ```

These examples apply to management domains:

1. To delete the action named ″E-mail root″ from the response named ″E-mail root anytime″ that is defined on the management server, run this command on the management server:

   ```
   chresponse -p -n "E-mail root" "E-mail root anytime"
   ```

2. In this example, the action named ″E-mail root″ will be used Monday through Friday from 8 AM to 6 PM, will use the command **/usr/sbin/rsct/bin/notifyevent root**, will save standard output in the audit log, and will expect return code 5 from the action. To add ″E-mail root″ to the response ″E-mail root anytime″ that is defined on the management server, run this command on the management server:

   ```
   chresponse  -a  -n "E-mail root" -d 2-6 -t 0800-1800 \
   -s "/usr/sbin/rsct/bin/notifyevent root" -o  -r 5  \
   "E-mail root anytime"
   ```

3. To delete the action named ″E-mail root″ from the response named ″E-mail root anytime″ that is defined on the managed node **nodeB**, run this command on the management server:

   ```
   chresponse -p -n "E-mail root" "E-mail root anytime":nodeB
   ```

These examples apply to peer domains:

1. In this example, the action named ″E-mail root″ will be used Monday through Friday from 8 AM to 6 PM, will use the command **/usr/sbin/rsct/bin/notifyevent root**, will save standard output in the audit log, and will expect return code 5 from the action. To add ″E-mail root″ to the response ″E-mail root anytime″ that is defined on node **nodeA** in the domain, run this command on any node in the domain:

```
chresponse  -a  -n "E-mail root" -d 2-6 -t 0800-1800 \
-s "/usr/sbin/rsct/bin/notifyevent root" -o  -r 5  \
"E-mail root anytime":nodeA
```

2. To delete the action named ″E-mail root″ from the response named ″E-mail root anytime″ that is defined on node **nodeA** in the domain, run this command on any node in the domain:

```
chresponse -p -n "E-mail root" "E-mail root anytime":nodeA
```

Location

/usr/sbin/rsct/bin/chresponse

Related Information

Books: *RSCT: Administration Guide*, for more information about ERRM operations

Commands: **lscondresp**, **lsresponse**, **mkcondresp**, **mkresponse**, **rmresponse**

Information Files: **rmccli**

---

chrole Command

Purpose

Changes role attributes. This command applies only to AIX 4.2.1 and later.

Syntax

chrole [**-R** *load_module*] *Attribute*=*Value ... Name*

Description

The **chrole** command changes attributes for the role identified by the *Name* parameter. The role name must already exist. To change an attribute, specify the attribute name and the new value with the *Attribute*=*Value* parameter.

If you specify a single incorrect attribute or attribute value with the **chrole** command, the command does not change any attribute.

You can use the Users application in Web-based System Manager (wsm) to change user characteristics. You could also use the System Management Interface Tool (SMIT) **smit chrole** fast path to run this command.

If the system is configured to use multiple domains for the role database, role modification is performed according to the order specified by the **secorder** attribute of the roles database stanza in the **/etc/nscontrol.conf** file. Only the first matching role is modified. Duplicate roles from the remaining domains are not modified. Use the **-R** flag to modify the role from a specific domain.

When the system is operating in enhanced Role Based Access Control (RBAC) mode, modifications made to the role database are not used for security considerations until the database is sent to the kernel security tables through the **setkst** command.

Flags

-R *load_module* Specifies the loadable module to use for the role modification.

Attributes

If you have the proper authority, you can set the following user attributes:

auth_mode Specifies the authentication that is required to assume the role when the **swrole** command is used. You can specify the following values:

NONE No authentication is required.

INVOKER
The invoker of the **swrole** command is required to enter their own password to assume the role. The **INVOKER** value is the default value.

authorizations List of additional authorizations required for this role beyond those defined by the roles in the **rolelist** attribute. The *Value* parameter is a list of authorization names, separated by commas.

dfltmsg Contains the default role-description text to use if message catalogs are not in use.

groups List of groups to which a user should belong, in order to effectively use this role. This attribute is for information only and does not automatically make the user a member of the list of groups. The *Value* parameter is a list of group names, separated by commas.

id Specifies the unique numeric ID for the role. You must specify the **id** attribute.

Attention: Do not modify the attribute value after the role is assigned to a user.

msgcat Contains the file name of the message catalog that holds the one-line descriptions of system roles. The *Value* parameter is a character string.

msgnum Contains the index into a message catalog for a description of the role. The *Value* parameter is an integer.

msgset Contains the message set that includes the role description in the message catalog.

rolelist Lists the roles implied by this role. The *Value* parameter is a list of role names, separated by commas.

screens Lists the SMIT screen identifiers allowing roles to be mapped to various SMIT screens. The *Value* parameter is a list of SMIT screen identifiers, separated by commas.

visibility Specifies the role's visibility status to the system. The *Value* parameter is an integer. Possible values are:

1 The role is enabled, displayed, and selectable. Authorizations contained in this role are applied to the user. If the attribute does not exist or has no value, the default value is 1.

0 The role is enabled and displayed as existing, but *not* selectable through a visual interface. Authorizations contained in this role are applied to the user.

-1 The role is disabled. Authorizations contained in this role are *not* applied to the user.

Security

The **chrole** command is a privileged command. You must assume a role that has the following authorization to run the command successfully.

aix.security.role.change Required to run the command.

Auditing Events

Event	Information
ROLE_Change	role, attribute

Files Accessed

Mode	File
rw	/etc/security/roles

Mode	File
r	/etc/security/user.roles

RBAC Disclaimer

Examples

1. To change the authorizations of the `ManagePasswds` role to `aix.security.passwd`, use the following command:

   ```
   chrole authorizations=aix.security.passwd ManagePasswds
   ```

2. To change the authorizations of the `ManagePasswds` role in LDAP to `aix.security.passwd`, use the following command:

   ```
   chrole -R LDAP authorizations=aix.security.passwd ManagePasswds
   ```

Files

/etc/security/roles	Contains the attributes of roles.
/etc/security/user.roles	Contains the role attribute of users.

Related Information

The **lsrole** command, **mkrole** command, **rmrole** command, **chuser** command, **lsuser** command, **mkuser** command, and **setkst** command.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

Securing the network in *Security*.

RBAC in *Security*.

---

chroot Command

Purpose

Changes the root directory of a command.

Syntax

chroot *Directory Command*

Description

> **Attention:** If special files in the new root directory have different major and minor device numbers than the real root directory, it is possible to overwrite the file system.

The **chroot** command can be used only by a user operating with root user authority. If you have root user authority, the **chroot** command changes the root directory to the directory specified by the *Directory* parameter when performing the *Command*. The first / (slash) in any path name changes to *Directory* for the specified *Command* and any of its children.

The *Directory* path name is always relative to the current root. Even if the **chroot** command is in effect, the *Directory* path name is relative to the current root of the running process.

A majority of programs may not operate properly after the **chroot** command runs. For example, the commands that use the shared libraries are unsuccessful if the shared libraries are not in the new root file system. The most commonly used shared library is the **/usr/ccs/lib/libc.a** library.

The **ls -l** command is unsuccessful in giving user and group names if the current root location makes the **/etc/passwd** file beyond reach. In addition, utilities that depend on localized files (**/usr/lib/nls/***) may also be unsuccessful if these files are not in the new root file system. It is your responsibility to ensure that all vital data files are present in the new root file system and that the path names accessing such files are changed as necessary.

Note: Ensure that the **/usr/sbin/execerror** command is available on the new root file system so that descriptive error messages are returned in the event of a **chroot** failure. Otherwise, if there is an error, **chroot** returns `Killed` and nothing more.

Parameters

Command	Specifies a command to run with the **chroot** command.
Directory	Specifies the new root directory.

Security

RBAC Disclaimer

Examples

Attention: The commands in the following examples may depend on shared libraries. Ensure that the shared libraries are in the new root file system before you run the **chroot** command.

1. To run the **pwd** command with the **/usr/bin** directory as the root file system, enter:

   ```
   mkdir /usr/bin/lib

   cp /usr/ccs/lib/libc.a /usr/bin/lib

   cp /usr/lib/libcrypt.a /usr/bin/lib

   chroot /usr/bin pwd
   ```

2. To run a Korn shell subshell with another file system as the root file system, enter:

   ```
   chroot /var/tmp /usr/bin/ksh
   ```

   This makes the directory name / (slash) refer to the `/var/tmp` for the duration of the `/usr/bin/ksh` command. It also makes the original root file system inaccessible. The file system on the `/var/tmp` file must contain the standard directories of a root file system. In particular, the shell looks for commands in the `/bin` and `/usr/bin` files on the `/var/tmp` file system.

   Running the `/usr/bin/ksh` command creates a subshell that runs as a separate process from your original shell. Press the END OF FILE (Ctrl-d) key sequence to end the subshell and go back to where you were in the original shell. This restores the environment of the original shell, including the meanings of the . (current directory) and the / (root directory).

3. To create a file relative to the original root, not the new one, enter:

   ```
   chroot directory Command > file
   ```

Files

/etc/passwd	Specifies file that contains basic user attributes.
/usr/ccs/lib/libc.a	Specifies the standard I/O library and the standard C library.
/usr/ccs/lib/libcurses.a	Specifies the curses library.
/usr/lib/liblvm.a	Specifies the LVM (Logical Volume Manager) library.

/usr/ccs/lib/libm.a	Specifies the math library.
/usr/lib/libodm.a	Specifies the ODM (Object Data Manager) library.
/usr/sbin/chroot	Contains the **chroot** command.

Related Information

The **ksh** command, **ls** command.

The **chdir** subroutine, **chroot** subroutine.

The File systems in *Operating system and device management* explains file system types, management, structure, and maintenance.

---

chrsrc Command

Purpose

Changes the persistent attribute values of a resource or a resource class.

Syntax

To change the persistent attribute values of a *resource,* using data that is...

- entered on the command line:

  chrsrc –s *"selection_string"* [**–a** | **–N** { *node_file* | *"-"* }] [**–v**] [**–h**] [**–TV**] *resource_class attr*=*value*...

  chrsrc –r [**–v**] [**–h**] [**–TV**] *resource_handle attr*=*value*...
- predefined in an input file:

  chrsrc –f *resource_data_input_file* **–s** *"selection_string"* [**–a** | **–N** { *node_file* | *"-"* }] [**–v**] [**–h**] [**–TV**] *resource_class*

  chrsrc –f *resource_data_input_file* **–r** [**–v**] [**–h**] [**–TV**] *resource_handle*

To change the persistent attribute values of a *resource class,* using data that is...

- entered on the command line:

  chrsrc { **–c** | **–C** *domain_name*... } [**–v** [**–a**] [**–h**] [**–TV**] *resource_class attr*=*value*...
- predefined in an input file:

  chrsrc –f *resource_data_input_file* { **–c** | **–C** *domain_name*... } [**–v**] [**–a**] [**–h**] [**–TV**] *resource_class*

Description

The **chrsrc** command changes the persistent attribute values of a resource or a resource class. By default, this command changes the persistent attribute values of a *resource.* Use the **-r** flag to change only the persistent attribute values of the resource that is linked with *resource_handle*. Use the **-s** flag to change the persistent attribute values of all of the resources that match *selection_string*. To change the persistent attributes of a *resource class,* use the **-c** flag.

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.

The **chrsrc** command cannot change dynamic attributes, nor can it change persistent attributes that are designated as **read_only**. 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 *not* designated as **read_only**, use the **-v** flag. When the **chrsrc** command is run with the **-v** flag, the specified attributes are not changed, but are instead merely verified to be persistent and not designated as **read_only**. Once you run **chrsrc -v** to verify that the attributes that are specified on the command line or in *resource_data_input_file* are valid, you can issue the **chrsrc** command without the **-v** flag to actually change the attribute values.

Note, however, that just because an attribute ″passes″ when **chrsrc -v** is run does not ensure that the attribute can be changed. The underlying resource manager that controls the specified resource determines which attributes can be changed by the **chrsrc** command. After **chrsrc** is run without the **-v** flag, an error message will indicate whether any specified attribute could not be changed.

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 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**.

–c Changes the persistent attribute values for *resource_class*.

–C *domain_name...*
Changes 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 change 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**.

–f *resource_data_input_file*
Specifies the name of the file that contains resource attribute information.

–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**.

–r Changes the persistent attribute values for the specific resource that matches *resource_handle*.

–s ″*selection_string*″
Changes the persistent attribute values for all of the resources that match *selection_string*. *selection_string* must be enclosed within either double or single quotation marks. If *selection_string* contains double quotation marks, enclose it in single quotation marks, for example:

```
-s 'Name == "testing"'
```

```
-s 'Name ?= "test"'
```

Only persistent attributes can be listed in a selection string. For information on how to specify selection strings, see the *RSCT: Administration Guide*.

–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 *not* designated as **read_only**. The **chrsrc** command does *not* change any persistent attribute values 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

attr=*value*...
Specifies one or more pairs of attributes and their associated values. *attr* is any defined persistent attribute name. Use the **lsrsrcdef** command to display a list of the defined persistent attributes and their datatypes for the specified resource. The value specified must be the appropriate datatype for the associated attribute. For example, if **NodeNumber** is defined as a **Uint32** datatype, enter a positive numeric value.

Do not specify this parameter if you run **chrsrc** with the **-f** flag.

resource_class
Specifies a resource class name. Use the **lsrsrcdef** command to display a list of defined resource class names.

resource_handle
Specifies a resource handle that is linked with the resource that you want to change. Use the **lsrsrc** command to display a list of valid resource handles. The resource handle must be enclosed within double quotation marks, for example:

```
"0x4017 0x0001 0x00000000 0x0069684c 0x0d4715b0 0xe9635f69"
```

Security

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

Exit Status

0 The command has run successfully.

1 An error occurred with RMC.

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

3 An incorrect flag was specified on the command line.

4 An incorrect parameter was specified on the command line.

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

6 No resources were found that match the selection string.

Environment Variables

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command

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

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

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

0 Specifies *local* scope.

1 Specifies *local* scope.

2 Specifies *peer domain* scope.

3 Specifies *management domain* scope.

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

Implementation Specifics

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

Standard Output

When the **-h** flag is specified, this command's usage statement is written to standard output. All verbose messages are written to standard output.

Standard Error

All trace messages are written to standard error.

Examples

1. To change the **Int32**, **Uint32** and **SD** persistent resource attributes in resource class **IBM.Foo** for the resources that have a **Name** equal to **c175n05**, enter:

```
chrsrc -s 'Name == "c175n05"' IBM.Foo \
Int32=-9999 Uint32=9999\
SD='["testing 1 2 3",1,{2,4,6}]'
```

2. To change the **Int32**, **Uint32** and **SD** resource attributes in resource class **IBM.Foo** for the resource that has a **Name** starting with **c175n**, using *resource_data_input_file* with the following contents:

```
PersistentResourceAttributes::
resource 1:
     Int32  = -9999
     Uint32 = 9999
     SD     = ["testing 1 2 3",1,{2,4,6}]
```

   enter:

```
chrsrc -f /tmp/IBM.Foo.chrsrc \
-s 'Name ?= "c175n"' IBM.Foo
```

3. To change the **Name** persistent resource attribute for the resource that has a resource handle equal to ″0x0001 0x4005 0x35ae868c 0x00000000 0xfeef2948 0x0d80b827″, enter:

```
chrsrc -r "0x0001 0x4005 0x35ae868c 0x00000000 0xfeef2948 0x0d80b827" Name="c175n05"
```

4. To change the **Int32**, **Uint32** and **SD** persistent resource attributes in resource class **IBM.Foo** for the resources that have a **Name** equal to **Test_Name** on nodes **node1.linwood.com** and **node2.linwood.com** in the cluster, using the **/u/joe/common_nodes** file:

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

   as input, enter:

   ```
   chrsrc -s 'Name == "Test_Name"' -N /u/joe/common_nodes IBM.Foo \
   Int32=-9999 Uint32=9999 \
   SD='["testing 1 2 3",1,{2,4,6}]'
   ```

Location

/usr/sbin/rsct/bin/chrsrc

Related Information

Books:

- *CSM: Administration Guide,* for information about node groups
- *CSM: Command and Technical Reference,* for information about the **nodegrp** command
- *RSCT: Administration Guide,* for information about RMC operations and about how to use expressions and selection strings

Commands: **lsrsrc**, **lsrsrcdef**, **mkrsrc**, **nodegrp**, **rmrsrc**

Information Files: **rmccli**, for general information about RMC commands

---

chsec Command

Purpose

Changes the attributes in the security stanza files.

Syntax

chsec [**-f***File*] [**-s** *Stanza*] [**-a** *Attribute* **=** *Value* ...]

Description

The **chsec** command changes the attributes stored in the security configuration stanza files. These security configuration stanza files have attributes that you can specify with the *Attribute* **=** *Value* 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**

When modifying attributes in the **/etc/security/environ**, **/etc/security/lastlog**, **/etc/security/limits**, **/etc/security/passwd**, and **/etc/security/user** files, the stanza name specified by the *Stanza* parameter must either be a valid user name or `default`. When modifying attributes in the **/etc/security/group** file, the stanza name specified by the *Stanza* parameter must either be a valid group name or `default`. When modifying attributes in the **/usr/lib/security/mkuser.default** file, the *Stanza* parameter must be either `admin` or `user`. When modifying attributes in the **/etc/security/portlog** file, the *Stanza* parameter must be a valid port name. When modifying attributes in the **/etc/security/login.cfg** file, the *Stanza* parameter must either be a valid port name, a method name, or the **usw** attribute.

When modifying attributes in the **/etc/security/login.cfg** or **/etc/security/portlog** file in a stanza that does not already exist, the stanza is automatically created by the **chsec** command.

You cannot modify the **password** attribute of the **/etc/security/passwd** file using the **chsec** command. Instead, use the **passwd** command.

Only the root user or a user with an appropriate authorization can change administrative attributes. For example, to modify administrative group data, the user must be root or have GroupAdmin authorization.

Flags

-a *Attribute* **=** *Value*	Specifies the attribute to modify and the new value for that attribute. If you do not specify the value, the attribute is removed from the given stanza.
-f *File*	Specifies the name of the stanza file to modify.
-s *Stanza*	Specifies the name of the stanza to modify.

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** command to allow the root user to access the security databases.

On a Trusted AIX system, only users with the **aix.mls.clear.write** authorization can modify clearance attributes. Only users with the **aix.mls.tty.write** authorization can modify the port attributes.

Auditing Events

Event	Information
USER_Change	user name, attribute
GROUP_Change	group name, attribute
PORT_Change	port, attribute

Files Accessed

Mode	File
rw	**/etc/security/environ**

Mode	File
rw	/etc/security/group
rw	/etc/security/audit/hosts
rw	/etc/security/lastlog
rw	/etc/security/limits
rw	/etc/security/login.cfg
rw	/usr/lib/security/mkuser.default
rw	/etc/nscontrol.conf
rw	/etc/security/passwd
rw	/etc/security/portlog
rw	/etc/security/pwdalg.cfg
rw	/etc/security/roles
rw	/etc/security/smitacl.user
rw	/etc/security/smitacl.group
rw	/etc/security/user
rw	/etc/security/user.roles

RBAC Disclaimer 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**

Examples

1. To change the **/dev/tty0** port to automatically lock if 5 unsuccessful login attempts occur within 60 seconds, enter:
   ```
   chsec -f /etc/security/login.cfg -s /dev/tty0 -a logindisable=5 -a logininterval=60
   ```
2. To unlock the **/dev/tty0** port after it has been locked by the system, enter:
   ```
   chsec -f /etc/security/portlog -s /dev/tty0 -a locktime=0
   ```
3. To allow logins from 8:00 a.m. until 5:00 p.m. for all users, enter:
   ```
   chsec -f /etc/security/user -s default -a logintimes=:0800-1700
   ```
4. To change the CPU time limit of user `joe` to 1 hour (3600 seconds), enter:
   ```
   chsec -f /etc/security/limits -s joe -a cpu=3600
   ```

Files

/usr/bin/chsec	Specifies the path to the **chsec** command.
/etc/security/environ	Contains the environment attributes of users.
/etc/security/group	Contains extended attributes of groups.
/etc/security/audit/hosts	Contains host and processor IDs.
/etc/security/lastlog	Defines the last login attributes for users.
/etc/security/limits	Defines resource quotas and limits for each user.
/etc/security/login.cfg	Contains port configuration information.
/usr/lib/security/mkuser.default	Contains the default values for new users.
/etc/nscontrol.conf	Contains the configuration information of some name services.
/etc/security/passwd	Contains password information.
/etc/security/portlog	Contains unsuccessful login attempt information for each port.
/etc/security/pwdalg.cfg	Contains the configuration information for loadable password algorithms (LPA).
/etc/security/roles	Contains a list of valid roles.

/etc/security/smitacl.user	Contains user ACL definitions.
/etc/security/smitacl.group	Contains group ACL definitions.
/etc/security/user	Contains the extended attributes of users.
/etc/security/user.roles	Contains a list of roles for each user.
/etc/security/enc/LabelEncodings	Contains label definitions for the Trusted AIX system.

Related Information

The **chgroup** command, **chuser** command, **grpck** command, **login** command, **lsgroup** command, **lssec** command, **lsuser** command, **mkgroup** command, **mkuser** command, **passwd** command, **pwdck** command, **rmgroup** command, **rmuser** command, **su** command, **usrck** command.

The **getgroupattr** subroutine, **getportattr** subroutine, **getuserattr** subroutine, **getuserpw** subroutine, **putgroupattr** subroutine, **putportattr** subroutine, **putuserattr** subroutine, **putuserpw** subroutine.

Trusted AIX in the *Security*.

---

chsensor Command

Purpose

Changes the attributes of a resource monitoring and control (RMC) sensor.

Syntax

chsensor [**-i** *seconds*] [**-a** | **-n** *host1* [**,** *host2* **,** ...] | **-N** { *node_file* | *"-"* }] [**-h**] [**-v** | **-V**] *sensor_name* *attr1*=*value1* [*attr2*=*value2* ...]

Description

The **chsensor** command changes the attributes of a resource monitoring and control (RMC) sensor. Use the *sensor_name* parameter to specify which sensor you are changing.

The **chsensor** command runs on any node. If you want **chsensor** to run on all of the nodes in a domain, use the **-a** flag. If you want **chsensor** to run on a subset of nodes in a domain, use the **-n** flag.

Flags

-a Changes 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, **chsensor -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.

-i *seconds*
Specifies the interval in which the sensor command is run to update the values of the sensor attributes. *seconds* is an integer value and must be greater than or equal to **10**. The sensor command is run at the specified interval only when a sensor resource is monitored. If the interval is set to **0**, the sensor command will not be automatically run. Using the **refsensor** command is independent of interval updates.

-n *host1***[,***host2***...]**
Specifies the node on which the sensor should be changed. By default, the sensor is changed on the local node. This flag is only appropriate in a management domain or a peer domain.

-N {*node_file* **| "-"}**
Specifies a file or standard input listing the nodes on which the sensor must be removed. This flag is only appropriate in a Cluster Systems Management (CSM) or a peer domain cluster.

-h Writes the command's usage statement to standard output.

-v | -V
Writes the command's verbose messages to standard output.

Parameters

sensor_name
Specifies the name of the sensor to change.

*attr1***=***value1* **[***attr2***=***value2* **...]**
Specifies one or more sensor attributes and the new values to which they will be set. You can change the values of these two attributes:

Name Specifies the new name of the sensor. If the new name is a string that contains spaces or special characters, it must be enclosed in quotation marks.

ControlFlags Specifies that special handling is required for this sensor instead of the default behavior. You can specify one of these values:

0 Indicates that no special handling is required. This is the default.

1 Indicates that the command in this sensor will be run any time, even at the initial stage (when **lssensor** is called or when monitoring is just started.) It is not recommanded that you specify this value, unless you expect the command to run very soon. Setting this value could block other requests to the sensor resource manager, so that those requests will not be processed until the command ends.

2 Indicates that output from the command in the **SavedData** field is not saved permanently to **SavedData** persistent respurce 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 command when monitoring is 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**.

UserName Specifies the name of a user whose privileges will be used to run the command. The user should already be defined on the system.

Description Provides a description of the sensor and what it is monitoring.

ErrorExitValue
Specifies which exit values will be interpreted as errors, as follows:

0 No exit values are interpreted as errors.

1 Exit values other than **0** are interpreted as errors.

2 An exit value of **0** is interpreted as an error.

If the exit value indicates an error as specified by this attribute, no dynamic attribute values (except **ExitValue**) are updated.

Security

The user needs write permission for the **IBM.Sensor** resource class in order to run **chsensor**. Permissions are specified in the access control list (ACL) file on the contacted system. See the *RSCT: Administration Guide* for details on the ACL file and how to modify it.

Exit Status

0 The command has run successfully.

1 An incorrect combination of flags and parameters has been entered.

6 No sensor resources were found.

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

Environment Variables

CT_CONTACT

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

CT_IP_AUTHENT

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

CT_MANAGEMENT_SCOPE

Determines the management scope that is used for the 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 change the **Name** attribute of the **SensorA** sensor to **Sensor1A**, enter:

   ```
   chsensor SensorA Name=Sensor1A
   ```

2. To change the update interval of the **SensorA** sensor to **10**, enter:

   ```
   chsensor -i 10 SensorA
   ```

Location

/usr/sbin/rsct/bin/chsensor

Related Information

Books: *RSCT: Administration Guide*, for information about the ACL authorization file

Commands: **lssensor**, **mksensor**, **refsensor**, **rmsensor**

Information Files: **rmccli**, for information about *attr*=*value* syntax

---

chserver Command

Purpose

Changes a subserver definition in the subserver object class.

Syntax

chserver -t *OldSubserver* [**-c** *CodePoint*] [**-s** *NewSubsystem*] [**-t** *NewSubserver*]

Description

The **chserver** command modifies an existing subserver definition in the subserver object class. It can change subserver types, the owning subsystem, or the subserver code point.

Flags

-c *CodePoint*
Specifies the *CodePoint* integer that identifies the subserver. This is the value used by the subsystem to recognize the subserver. The **chserver** command is unsuccessful if the *CodePoint* already exists for the existing subsystem name and no new subsystem name is entered. It is also unsuccessful if the *NewSubsystem* name and subserver *CodePoint* exist in the subserver object class. The limit for the *CodePoint* storage is the same as a short integer (1 through 32,768).

-s *NewSubsystem*
Specifies the name that uniquely identifies the *NewSubsystem* to the subserver it belongs to. The **chserver** command is unsuccessful if one of the following occurs:

- The *NewSubsystem* name is not known in the subsystem object class.
- The *NewSubsystem* name is known in the subsystem object class but uses signals as its communication method.
- The *NewSubsystem* name already exists with the existing subserver *CodePoint* value in the Subserver Type object class, and no subserver *CodePoint* value is entered.
- A new subserver *CodePoint* is entered, with the *NewSubsystem* name and subserver *CodePoint* already existing in the Subserver Type object class.

-t *NewSubserver*
Specifies the name that uniquely identifies the *NewSubserver*. The **chserver** command is unsuccessful if the *NewSubserver* type is already known in the subserver object class.

-t *OldSubserver*
Specifies the name that uniquely identifies the existing subserver. The **chserver** command is unsuccessful if the *OldSubserver* type is not known in the subserver object class.

Security

Auditing Events

If the auditing subsystem is properly configured and is enabled, the **chserver** command generates the following audit record (event) every time the command is run:

Event	Information
SRC_Chserver	Lists in an audit log the name of the subsystem and the fields that have been changed.

For more details about how to properly select and group audit events, and how to configure audit event data collection, see Setting up Auditing in *Security*.

RBAC Disclaimer

Examples

1. To change the subserver type, enter:

   ```
   chserver  -t old  -t new
   ```

   This changes the subserver type from the `old` subserver type to the `new` subserver type.
2. To change the owning subsystem, enter:

   ```
   chserver  -t old  -s srctest
   ```

   This changes the owning subsystem to `srctest`.
3. To change the subserver type, subsystem, and subserver code point, enter:

   ```
   chserver  -t old  -t new  -s srctest  -c 1234
   ```

   This changes the subserver type from the `old` to the `new` subserver type, the owning subsystem to `srctest`, and the subserver code point to `1234`.

Files

/etc/objrepos/SRCsubsys	Specifies the SRC Subsystem Configuration object class.
/etc/objrepos/SRCsubsvr	Specifies the SRC Subserver Configuration object class.

Related Information

The **auditpr** command, **mkserver** command, **rmserver** command, **startsrc** command, **stopsrc** command, **traceson** command, **tracesoff** command.

Auditing Overview in *Security*.

System Resource Controller in *Operating system and device management*.

Defining Your Subsystem to the SRC in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

System Resource Controller (SRC) Overview for Programmers in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

chservices Command

Purpose

Changes the contents of the **/etc/services** file.

Syntax

To Add or Activate an Entry:

chservices [**-a**] **-v** *ServiceName* **-p** *protocol* **-n** *port* [**-u** *"Alias ..."*]

To Change an Entry:

chservices -c -v *ServiceName* **-p** *protocol* **-n** *port* [**-V** *NewServiceName*] [**-P** *NewProtocol*] [**-N** *NewPort*] [**-u** *"Alias ..."*]

To Deactivate an Entry:

chservices -d -v *ServiceName* **-p** *protocol* **-n** *port* [**-V** *NewServiceName*] [**-u** *Alias ..."*]

Description

The **chservices** command adds, deletes, or changes entries in the **/etc/services** file. These entries are related to known services used in the DARPA Internet and also related to information used by the **inetd** server. The entries for the **inetd** server determine how the system handles Internet service requests.

The **chservices** command manipulates the following entries for known services:

- The official Internet service name specified by the *ServiceName* variable.
- The port number, specified by the *port* variable, used for the service.
- The transport protocol, specified by the *protocol* variable, used for the service.
- A list of unofficial names, specified by the *Alias* variable, used by the service.

Flags

-a	Adds or activates an entry in the **/etc/services** file. If the requested service exists in the file, the **-a** flag uncomments the line. If the line does not exist, the **-a** flag adds the line to the file. This is the default action.
-c	Changes an entry in the **/etc/services** file.
-d	Deactivates an entry in the **/etc/services** file by commenting the line in the file.
-N *NewPort*	Specifies a socket port number.
-n *port*	Specifies a socket port number.
-P *NewProtocol*	Specifies a new protocol name for a current protocol name.
-p *protocol*	Specifies the protocol.
-V *NewName*	Specifies a new service name.
-v *ServiceName*	Specifies the service name.
-u *"Alias..."*	Specifies a list of aliases.

Note: Adding or keeping comments on lines modified with the **chservices** command is not supported.

Security

Access Control: Only the root user and members of the system group have access to this command.

Examples

1. To add the service, `gregsapp`, as a `udp` service on port 1423, enter:

   ```
   chservices -a -v gregsapp -p udp -n 1423
   ```

2. To add the service, `gregsapp`, as a `udp` service on port 1423 with an alias of `fredsapp`, enter:

   ```
   chservices -a -v gregsapp -p udp -n 1423 -u
   "fredsapp"
   ```

3. To change the port of the service specified as `gregsapp` with a `udp` protocol to 1456, enter:

   ```
   chservices -c -v gregsapp -p udp -N 1456
   ```

4. To deactivate the `gregsapp` service on `udp` port 1456 by commenting it out, enter:
   ```
   chservices -d -v gregsapp -p udp -n 1456
   ```

Files

/usr/sbin/chservices	Contains the **chservices** command.
/etc/services	Contains services information for the **inetd** daemon.

Related Information

The **chsubserver** command.

The **inetd** daemon, **fingerd** daemon, **ftpd** daemon, **rexecd** daemon, **rlogind** daemon, **rshd** daemon, **syslogd** daemon, **talkd** daemon, **telnetd** daemon, **tftpd** daemon.

The **inetd.conf** file format, **protocols** file format, **services** file format.

TCP/IP daemons in *Networks and communication management*.

---

chsh Command

Purpose

Changes a user's login shell.

Syntax

chsh [**-R** *load_module*] [*Name* [*Shell*]]

Description

The **chsh** command changes a user's login **shell** attribute. The **shell** attribute defines the initial program that runs after a user logs in to the system. This attribute is specified in the **/etc/passwd** file. By default, the **chsh** command changes the login shell for the user who gives the command.

The **chsh** command is interactive. When you run the **chsh** command, the system displays a list of the available shells and the current value of the **shell** attribute. Then, the system prompts you to change the shell. You must enter the full path name of an available shell.

If you have execute permission for the **chuser** command, you can change the login shell for another user. To change the login shell for another user, specify a *Name* parameter. Valid shells are defined in the usw stanza of the **/etc/security/login.cfg** file. The default list of valid shells is: **/usr/bin/ksh**, **/usr/bin/sh**, **/usr/bin/bsh**, **/usr/bin/csh** but your system manager may have defined more.

For users that are created 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.

Flag

-R *load_module* Specifies the loadable I&A module used to change the user's shell.

Exit Status

This command returns the following exit values:

0	The command runs successfully and all requested changes are made.
>0	An error occurred. The printed error message gives further details about the type of failure.

Security

Access Control

All users should have execute (x) access to this command since the program enforces its own access policy. This command should be installed as a program in the trusted computing base (TCB). The command should be owned by the **security** group with the **setgid** (SGID) bit set.

Files Accessed

Mode	File
x	**/usr/bin/chuser**
r	**/etc/security/login.cfg**
rw	**/etc/passwd**

RBAC Disclaimer

Limitations

Changing a user's shell may not be supported by all loadable I&A modules. If the loadable I&A module does not support changing a user's shell, an error is reported.

Examples

1. To change the shell that runs after you log in to the system, type:

   ```
   chsh
   ```

   Information similar to the following appears:

   ```
   current available shells:
    /usr/bin/sh
    /usr/bin/bsh
    /usr/bin/csh
    /usr/bin/ksh:
   current login shell:
    /usr/bin/ksh
   change (y/n)? >
   ```

   Indicate that a change should be made by entering `y` after the change `(y/n)?` prompt. Then, add the name of the shell you want when the `to?` prompt appears, as in the following example:

   ```
   change (y/n)? > y
   to? > /usr/bin/csh
   ```

   The next time you log in, the **/usr/bin/csh** shell appears.
2. To change the shell to `/usr/bin/ksh` for `kim`, type:

   ```
   chsh kim /usr/bin/ksh
   ```
3. To change the shell for LDAP I&A load module defined user `davis`, type:

   ```
   chsh -R LDAP davis
   ```

Files

/usr/bin/chsh	Specifies the path to the **chsh** command.
/usr/bin/chuser	Changes user information.
/etc/passwd	Contains the basic user attributes.
/etc/security/login.cfg	Contains login configuration information.

Related Information

The **chgroup** command, **chgrpmem** command, **chuser** command, **lsgroup** command, **lsuser** command, **mkgroup** command, **mkuser** command, **passwd** command, **pwdadm** command, **rmgroup** command, **rmuser** command, **setgroups** command, **setsenv** command.

Security describes the identification and authentication of users, discretionary access control, the trusted computing base, and auditing.

---

chslave Command

Purpose

Re-executes the **ypinit** command to retrieve maps from a master server and re-starts the **ypserv** daemon to change the slave server.

Syntax

/usr/etc/yp/chslave [**-C** | **-c**] [**-O** | **-o**] [**-I** | **-B** | **-N**] *Master*

Description

The **chslave** command re-invokes the **ypinit** command to retrieve maps from the master server you specify on the command line. The **ypserv** daemon is re-started after the **ypinit** command has completed successfully. The *Master* parameter specifies the host name of the master server. The master server specified can be the master server currently in use or a new master server that is configured and running.

You can use the Network application in Web-based System Manager (wsm) to change network characteristics. You could also use the System Management Interface Tool (SMIT) **smit chslave** fast path to run this command.

Flags

-B Invokes the **ypinit** command and starts the **ypserv** daemon. If the **ypserv** daemon is already running, this flag will cause the **ypinit** command to kill the daemon and then restart it. This flag is the default.
-C Invokes the **ypinit** command with the **-n** flag. The **chslave** command continues on errors. This flag is the default.
-c Stops execution when errors occur.
-I Executes the **ypinit** command immediately but does not start or restart the **ypserv** daemon.
-O Overwrites any maps that exist in the domain.
-o Prevents the overwrite of maps that exist in the domain. This flag is the default.
-N Invokes the **ypinit** command and restarts the **ypserv** daemon.

Examples

To retrieve maps from the master server named `host91`, enter:

```
chslave -O -B host91
```

This will overwrite any existing maps for the current domain.

Files

/etc/rc.nfs	Contains the startup script for NFS and NIS daemons.
/var/yp/domainname	Contains the NIS maps for the NIS domain.

Related Information

The **chmaster** command, **mkclient** command, **mkslave** command, **rmyp** command, **smit** command, **ypinit** command.

The **ypbind** daemon, **yppasswdd** daemon, **ypserv** daemon, **ypupdated** daemon.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

The System Management Interface Tool (SMIT) in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* explains the structure, main menus, and tasks that are done with SMIT.

Network Information Service (NIS) in *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*.

NIS Reference.

---

chssys Command

Purpose

Changes a subsystem definition in the subsystem object class.

Syntax

chssys -s *OldSubsystem* [**-a** *Arguments*] [**-e** *StandardError*] [**-i** *StandardInput*] [**-o** *StandardOutput*] [**-p** *Path*] [**-s** *NewSubsystem*] [**-t** *Synonym*] [**-u** *UserID*] [**-O** | **-R**] [**-d** | **-D**] [**-q** | **-Q**] [**-K** | [**-I** *MessageQueue* **-m** *MessageMtype* | **-f** *StopForce* **-n** *StopNormal* **-S**] [**-E** *Nice*] [**-G** *Group*] [**-w** *Wait*]

Description

The **chssys** command modifies an existing subsystem definition in the subsystem object class. If a new subsystem name is entered, the Subserver Type object class and the Notify object class are modified to reflect the new subsystem name.

> **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

-a *Arguments*	Specifies any arguments that must be passed to the program executed as the subsystem. These command *Arguments* are passed by the SRC to the subsystem according to the same rules used by the shell. Quoted strings are passed as a single argument, and blanks outside a quoted string delimit arguments. Single and double quotes can be used.
-d	Specifies that an inactive subsystem is displayed when the **lssrc -a** command request (status all) or the **lssrc -g** command request (status group) is made.

-D	Specifies that an inactive subsystem is not displayed when status all or status group requests are made.
-e *StandardError*	Specifies where the subsystem standard error data is placed.
-E *Nice*	Specifies the *Nice* value. The *Nice* parameter changes the execution priority of the subsystem. The valid values are 0 through 39 (ordinary *Nice* values 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.
-f *StopForce*	Specifies the signal sent to the subsystem when a forced stop of the subsystem is requested. Use only when the subsystem uses signals for communication. The **chssys** command is unsuccessful if the *StopForce* parameter specifies an invalid signal. The **-n** and **-S** flags must follow this flag.
-G *Group*	Specifies that the subsystem belongs to the group specified by the *Group* parameter and responds to all group actions on the group.
-i *StandardInput*	Specifies where the subsystem *StandardInput* is routed. This field is ignored when the subsystem uses sockets for communication.
-K	Specifies that the subsystem uses sockets as its communication method.
-l *MessageQueue*	Specifies that the subsystem uses message queues as its communication method. The *MessageQueue* parameter 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. The **-m** flag must follow this flag.
-m *MessageMtype*	Specifies the *MessageMtype* key that the subsystem expects on packets sent to the subsystem by the SRC. Use only when the subsystem uses message queues for communication. The *MessageMtype* must be greater than 0. This flag must be preceded by the **-l** flag.
-n *StopNormal*	Specifies the signal sent to the subsystem when a normal stop of the subsystem is requested. Use only when the subsystem uses signals for communication. The **chssys** command is unsuccessful if the *StopNormal* parameter specifies an invalid signal. This flag must be preceded by the **-f** flag and followed by the **-S** flag.
-o *StandardOutput*	Specifies where the subsystem *StandardOutput* is placed.
-O	Specifies that the subsystem is not restarted if it stops abnormally.
-p *Path*	Specifies the absolute *Path* to the subsystem 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.
-R	Specifies that the subsystem is restarted if it stops abnormally.
-s *NewSubsystem*	Specifies the new name that uniquely identifies the subsystem. Any subservers or notify methods defined for the old subsystem's name are redefined for the *NewSubsystem* name. The **chssys** command is unsuccessful if the *NewSubsystem* name is already known in the subsystem object class.
-s *OldSubsystem*	Specifies the current name that uniquely identifies the subsystem. The **chssys** command is unsuccessful if the *OldSubsystem* name is not known in the subsystem object class.
-S	Specifies that the subsystem uses signals as its communication method. You cannot define subservers for the subsystem name when your communication method is signals. If a subserver is defined for the subsystem, the subserver definitions are deleted from the subserver object class. This flag must be preceded by the **-f** and **-n** flags.
-t *Synonym*	Specifies an alternate name for the subsystem. The **chssys** command is unsuccessful if the *Synonym* name is already known in the subsystem object class.
-u *UserID*	Specifies the user ID for the subsystem. The *UserID* that creates the subsystem is used for security auditing of that subsystem.
-w *Wait*	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 *Wait* value, it is not automatically restarted.

Security

Auditing Events

If the auditing subsystem is properly configured and is enabled, the **chssys** command generates the following audit record (event) every time the command is run:

Event	Information
SRC_Chssys	Lists in an audit log the name of the subsystem and the fields that have been changed.

For more information about properly selecting and grouping audit events, and configuring audit event data collection, see Setting up Auditing in *Security*.

RBAC Disclaimer

Examples

1. To change the subsystem name, enter:

   ```
   chssys  -s srctest  -s inetd
   ```

   This changes the subsystem name from `srctest` to `inetd`.
2. To change the communication type to sockets, enter:

   ```
   chssys  -s srctest  -K
   ```

   This changes the communication type for the subsystem to sockets.
3. To change the communication type to message queues, enter:

   ```
   chssys  -s srctest  -l 123456  -m 789
   ```

   This changes the communication type for the subsystem to message queues, with a message queue key of `123456` and a subsystem message type of `789`.
4. To change the communication type to signals, enter:

   ```
   chssys  -s srctest  -S  -n 30  -f 31
   ```

   This changes the communication type for the subsystem to signals, with a normal stop signal of `30` and a force stop signal of `31`.
5. To change the command arguments, enter:

   ```
   chssys  -s srctest  -a "-a 123 -b \"4 5 6\" -c '7 8 9'"
   ```

   This places `-a` as the first argument, `123` as the second, `-b` as the third, `4 5 6` as the fourth, `-c` as the fifth, and `7 8 9` as the sixth argument to the `srctest` subsystem.

Files

/etc/objrepos/SRCsubsys	Specifies the SRC Subsystem Configuration object class.
/etc/objrepos/SRCsubsvr	Specifies the SRC Subserver Configuration object class.
/etc/objrepos/SRCnotify	Specifies the SRC Notify Method object class.
/dev/SRC	Specifies the **AF_UNIX** socket file.
/dev/.SRC-unix	Specifies the location for temporary socket files.

Related Information

The **auditpr** command, **lssrc** command, **mkssys** command, **rmssys** command.

Auditing Overview in *Security*.

System Resource Controller in *Operating system and device management*.

Defining Your Subsystem to the SRC in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

System Resource Controller (SRC) Overview for Programmers in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

---

chsubserver Command

Purpose

Changes the contents of the /etc/inetd.conf file or similar system configuration file.

Syntax

To Add or Activate a Server or Subserver Entry:

chsubserver [**-a**] **-v** *ServiceName* **-p** *protocol* [**-t** *socket_type*][**-w** *WaitIndicator*] [**-u** *user*] [**-g** *program*] [**-r** *server*] [**-C** *ConfigFile*] [*program*] [*args*]

To Change a Server Entry:

chsubserver -c -v *ServiceName* **-p** *protocol* [**-t** *SocketType*] [**-w** *WaitIndicator*] [**-u** *user*] [**-g** *program*] [**-V** *NewServiceName*] [**-P** *NewProtocol*] [**-T** *NewSocketType*] [**-W** *NewWaitIndicator*] [**-U** *NewUser*] [**-G** *NewProgram*] [**-r** *server*] [**-C** *ConfigFile*] [*program*] [*args*]

To Deactivate a Server Entry or an inetd Subserver Entry:

chsubserver -d -v *ServiceName* **-p** *protocol* [**-t** *SocketType*] [**-w** *WaitIndicator*] [**-u** *user*] [**-g** *program*] [**-r** *server*] [**-C** *ConfigFile*] [*program*] [*args*]

Description

The **chsubserver** command adds, deletes, or changes entries in the **/etc/inetd.conf** system configuration file, which is the default, or a similar configuration file. These entries are related to known services used in the DARPA Internet and also related to information used by the **inetd** server. The entries for the **inetd** server determine how the system handles Internet service requests.

The **chsubserver** command also allows the user to refresh a server using the **-r** flag. The server specified is sent a **SIGHUP** signal to reread its configuration file. This allows you to edit the configuration file and have the changes take effect immediately.

Each service entry contains information about known services and information used by the **inetd** server. The **chsubserver** command manipulates the following entries for known services and for **inetd** server or other subserver information:

- The official Internet service name specified by the *ServiceName* variable.
- The transport protocol, specified by the *protocol* variable, used for the service.
- The type of socket, specified by the *SocketType* variable, associated with the service. The socket types associated with a service can be stream sockets or datagram sockets. Use only the **nowait** flag with stream sockets. Use either the **wait** or **nowait** flag with datagram sockets.

- A **wait** or **nowait** flag, specified by the *WaitIndicator* variable. The **wait** or **nowait** flag indicates whether the **inetd** server waits for a datagram server to release the socket before continuing to listen at the socket.
- The user name, specified by the *user* variable, that the **inetd** server uses to start a subserver.

You can use the System application in Web-based System Manager (wsm) to change system characteristics. You could also use the System Management Interface Tool (SMIT) **smit inetdconf** fast path to run this command.

Flags

-a	Adds or activates an entry in the configuration file. If the requested service exists in the configuration file, the **-a** flag uncomments the line. If the line does not exist, the **-a** flag adds the line to the configuration file. This is the default action.
-c	Changes an entry in the configuration file.
-C	Specifies a configuration file similar to **/etc/inetd.conf**.
-d	Deactivates an entry in the configuration file by commenting the line in the file.
-G *NewProgram*	Replaces the existing program to start.
-g *Program*	Specifies the program to start..
-P *NewProtocol*	Specifies a new protocol name for a current protocol name.
-p *protocol*	Specifies the protocol.
-r *server*	Sends a **SIGHUP** to the specified server.
-T *NewSocketType*	Replaces the existing type of socket, either a value of **stream** for stream sockets or a value of **dgram** for datagram sockets.
-t *SocketType*	Specifies a type of socket, either a value of **stream** for stream sockets or a value of **dgram** for datagram sockets.
-U *NewUser*	Replaces the existing user name.
-u *user*	Specifies a user name.
-V *NewName*	Specifies a new service name.
-v *ServiceName*	Specifies the service name.
-W *NewWaitIndicator*	Replaces the existing *WaitIndicator*.
-w *WaitIndicator*	Specifies either single-thread service with a value of **wait** or multithread service with a value of **nowait**.

Security

Access Control: Only the root user and members of the system group have access to this command.

Examples

1. To uncomment the uucp line in the /etc/inetd.conf file, enter:

   ```
   chsubserver -a -v uucp -p tcp
   ```

2. To add a line to the **/etc/inetd.conf** file that describes the gregserv service and runs the program `/usr/sbin/gregserv` as root over the udp protocol with stream sockets and arguments of ftpd, enter in one line:

   ```
   chsubserver -a -r inetd -v gregserv -p udp -t stream -w nowait -u
   root -g /usr/sbin/gregserv ftpd
   ```

   The **inetd** does not wait for confirmation. After adding the line to the file, the **inetd** program will be sent a **SIGHUP** signal.

3. To change the existing service from using stream sockets to using dgram sockets in the **/tmp/inetd.conf** file, enter in one line:

   ```
   chsubserver -c -v gregserv -p udp -t stream -T dgram -C /tmp/inetd.conf
   ```

4. To comment the gregserv service over udp in the **/etc/inetd.conf** file, enter:

```
chsubserver -d -v gregserv -p udp
```

Files

/usr/sbin/chsubserver	Contains the **chsubserver** command.
/etc/inetd.conf	Contains configuration information for the **inetd** daemon.

Related Information

The **chservices** command.

The **inetd** daemon, **fingerd** daemon, **ftpd** daemon, **rexecd** daemon, **rlogind** daemon, **rshd** daemon, **syslogd** daemon, **talkd** daemon, **telnetd** daemon, **tftpd** daemon.

The **inetd.conf** file format, **protocols** file format, **services** file format.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

TCP /IP daemons in *Networks and communication management*.

---

chtcb Command

Purpose

Changes or queries the **trusted computing base** attribute of a file.

Syntax

chtcb { **on** | **off** | **query** } *File* ...

Description

The **chtcb** command changes or queries the **trusted computing base** (TCB) attribute of the files you specify with the *File* parameter. The following alternatives are valid:

on	Enables the **trusted computing base** attribute.
off	Disables the **trusted computing base** attribute, if set.
query	Displays the value of the **trusted computing base** attribute.

This command should be executed on the trusted path.

Security

Access Control: This command should grant execute (x) access to the root user and members of the security group. The command should have the **trusted computing base** attribute.

Examples

1. To identify the `plans` file as part of the trusted computing base (TCB), set the **trusted computing base** attribute to the `on` value by entering the following:

   ```
   chtcb on plans
   ```

   The `plans` file now can be executed from the trusted path.
2. To query whether the `plans` file is part of the trusted computing base (TCB), enter:

   ```
   chtcb query plans
   ```

When the status appears, you know that the `plans` file is part of the trusted computing base if the TCB attribute is set to the **on** value.

3. To remove the `plans` file from the trusted computing base (TCB), enter:

```
chtcb off plans
```

Files

/usr/sbin/chtcb	Contains the **chtcb** command.

Related Information

The **tsh** command, **tsm** command, **tvi** command.

The **chmod** subroutine.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to *Security*.

chtun Command

Purpose

Changes a tunnel definition.

Syntax

chtun -t *tunnel_ID* **-v {4|6}** [**-s** *src_host_IP_address*] [**-d** *dst_host_IP_address*] [**-m** *pkt_mode*] [**-f** *fw_address* [**-x** *dst_mask*]] [**-e** *src_esp_algo*] [**-a** *src_ah_algo]*] [**-p** *src_policy*] [**-E** *dst_esp_algo*] [**-A** *dst_ah_algo]*] [**-P** *dst_policy*] [**-l** *lifetime*] [**-k** *src_esp_key*] [**-h** *src_ah_key*] [**-K** *dst_esp_key*] [**-H** *dst_ah_key*] [**-n** *src_esp_spi*] [**-u** *src_ah_spi*] [**-N** *dst_esp_spi*] [**-U** *dst_ah_spi*] [**-b** *src_enc_mac_algo*] [**-c** *src_enc_mac_key*] [**-B** *dst_enc_mac_algo*] [**-C** *dst_enc_mac_key*]

Description

Use the **chtun** command to change a definition of a tunnel between a local host and a tunnel partner host. If a flag is not specified, then the value given for the **gentun** command should stay the value for that field. It may also change the auto-generated filter rules created for the tunnel by the **gentun** command.

Flags

-A *dst_ah_algo]*	(**manual** tunnel only) Authentication algorithm, which is used by the destination for IP packet encryption. The valid values for **-A** depend on which authentication algorithms have been installed on the host. The list of all the authentication algorithms can be displayed by issuing the **ipsecstat -A** command.
-a *src_ah_algo]*	Authentication algorithm, used by source host for IP packet authentication. The valid values for **-a** depend on which authentication algorithms have been installed on the host. The list of all authentication algorithms can be displayed by issuing the **ipsecstat -A** command.
-B *dst_enc_mac_algo*	(**manual** tunnel only) Destination ESP Authentication Algorithm (New header format only). The valid values for **-B** depend on which authentication algorithms have been installed on the host. The list of all the authentication algorithms can be displayed by issuing the **ipsecstat -A** command.

-b *src_enc_mac_algo*
(**manual** tunnel only) Source ESP Authentication Algorithm (New header format only). The valid values for **-b** depend on which authentication algorithms have been installed on the host. The list of all the authentication algorithms can be displayed by issuing the **ipsecstat -A** command.

-C *dst_enc_mac_key*
(**manual** tunnel only) Destination ESP Authentication Key (New header format only). It must be a hexadecimal string started with ″0x″.

-c *src_enc_mac_key*
(**manual** tunnel only) Source ESP Authentication Key (New header format only). It must be a hexdecimal string started with ″0x″.

-d *dst_host_IP_address*
Destination Host IP address. For a host-host tunnel, this value is the IP address of the destination host interface to be used by the tunnel. For a host-firewall-host tunnel, this is the IP address of a destination host behind the firewall. A host name is also valid and the first IP address returned by the name server for the host name will be used.

-E *dst_esp_algo*
(**manual** tunnel only) Encryption algorithm, which is used by the destination for IP packet encryption. The valid values for **-E** depend on which encryption algorithms have been installed on the host. The list of all the encryption algorithms can be displayed by issuing the **ipsecstat -E** command.

-e *src_esp_algo*
Encryption algorithm, used by source host for IP packet encryption. The valid values for **-e** depend on which encryption algorithms have been installed on the host. The list of all encryption algorithms can be displayed by issuing the **ipsescstat -E** command.

-f *fw_address*
IP address of the firewall that is between source and destination hosts. A tunnel will be established between the source and the firewall. Therefore the corresponding tunnel definition must be made in the firewall host. A host name can also be specified with this flag, and the first IP address returned by name server for the host name will be used.

The **-m** flag is forced to use default value (**tunnel**) if **-f** is specified.

-H *dst_ah_key*
The Key String for destination AH. The input must be a hexdecimal string started with ″0x″.

-h *src_ah_key*
The Key String for source AH. The input must be a hexdecimal string started with ″0x″.

-K *dst_esp_key*
The Key String for destination ESP. The input must be a hexdecimal string started with ″0x″.

-k *src_esp_key*
The Key String for the source ESP. It is used by the source to create the tunnel. The input must be a hexdecimal string started with ″0x″.

-l *lifetime*
Key Lifetime, specified in minutes.

For **manual** tunnels, the value of this flag indicates the time of operability before the tunnel expires.

The valid values for **manual** tunnels are 0 - 44640. Value 0 indicates that the **manual** tunnel will never expire.

-m *pkt_mode*
Secure Packet Mode. This value must be specified as **tunnel** or **transport**.

-N *dst_esp_spi*
(**manual** tunnel only) Security Parameter Index for the destination ESP.

-n *src_esp_spi*
(**manual** tunnel only) Security Parameter Index for source ESP. This SPI and the destination IP address is used to determine which security association to use for ESP.

-P *dst_policy*
(**manual** tunnel only) Destination policy, identifies how the IP packet authentication and/or encryption is to be used by destination. If the value of this flag is specified as **ea**, the IP packet gets encrypted before authentication. If specified as **ae**, it gets encrypted after authentication, whereas specifying **e** or **a** alone corresponds to the IP packet being encrypted only or authenticated only.

-p *src_policy*
Source policy, identifies how the IP packet authentication and/or encryption is to be used by source. If the value of this flag is specified as **ea**, the IP packet gets encrypted before authentication. If specified as **ae**, it gets encrypted after authentication, whereas specifying **e** or **a** alone corresponds to the IP packet being encrypted only or authenticated only.

-s *src_host_IP_address*
Source Host IP address, IP address of the local host interface to be used by the tunnel. A host name is also valid and the first IP address returned by name server for the host name will be used.

-t *tunnel_ID*
The tunnel identifier (ID), a locally unique, numeric identifier for a particular tunnel definition. The value must match an existing tunnel ID.

-U *dst_ah_spi*
(**manual** tunnel only) Security Parameter Index for the destination AH.

-u *src_ah_spi*
(**manual** tunnel only) Security Parameter Index for source AH. This SPI and the destination IP address is used to determine which security association to use for AH.

-v
The IP version for which the tunnel is created. For IP version 4 tunnels, use the value of **4**. For IP version 6 tunnels, use the value of **6**.

-x *dst_mask*
This flag is used for host-firewall-host tunnels. The value is the network mask for the secure network behind a firewall. The Destination host specified with the **-d** flag is a member of the secure network. The combination of the **-d** and **-x** flags allows source host communications with multiple hosts in the secure network through the source-firewall tunnel, which must be in tunnel Mode.

This flag is valid only when **-f** is specified.

-y
(**manual** tunnel only) Replay prevention flag. Replay prevention is valid only when the ESP or AH header is using the new header format (see the **-z** flag). The valid values for the **-y** flag are Y (yes) and N (no).

-z
(**manual** tunnel only) New header format flag. The new header format reserves a field in ESP or AH header for replay prevention and also allows ESP authentication. The replay field is used only when the replay flag (**-y**) is set to Y. The valid values are Y (yes) and N (no).

Security

RBAC Disclaimer

Related Information

The **exptun** command, **gentun** command, **imptun** command, **lstun** command, **mktun** command, and **rmtun** command.

chtz Command

Purpose
Changes the *TimeZoneInfo* (TZ) environment variable in the **/etc/environment** file.

Syntax
chtz *TimeZoneInfo*

Description
The **chtz** command is a high-level shell command that changes the TZ environment variable in the **/etc/environment** file. The **chtz** command returns a value of `0` if successful and nonzero if unsuccessful.

Files

/etc/environment	Contains variables specifying the basic environment for all processes.

Related Information
The **date** command.

The **environment** file.

chuser Command

Purpose
Changes user attributes.

Syntax
chuser [**-R** *load_module*] *Attribute*=*Value ... Name*

Description
Attention: Do not use the **chuser** command if you have a Network Information Service (NIS) database installed on your system.

The **chuser** command changes attributes for the user identified by the *Name* parameter. The user name must already exist. To change an attribute, specify the attribute name and the new value with the *Attribute*=*Value* parameter. The following files contain local user attributes that are set by this command:

- **/etc/passwd**
- **/etc/security/environ**
- **/etc/security/limits**
- **/etc/security/user**
- **/etc/security/user.roles**
- **/etc/security/audit/config**
- **/etc/group**
- **/etc/security/group**

To change attributes for a user with an alternate Identification and Authentication (I&A) mechanism, the **-R** flag can be used to specify the I&A load module that user is defined under. If the **-R** flag is not specified, the **chuser** command treats the user as a local user. Load modules are defined in the **/usr/lib/security/methods.cfg** file.

If you specify a single incorrect attribute or attribute value with the **chuser** command, the command does not change any attribute.

You can use the Users application in Web-based System Manager (wsm) or the System Management Interface Tool (SMIT) **smit chuser** fast path to change user characteristics.

Changing the ID for an account can compromise system security and as a result one should not do so. However, when the ID is changed using the **chuser** command, ID collision checking is also controlled by the **dist_uniqid** attribute in the usw stanza of the **/etc/security/login.cfg** file. The behavior of ID collision control is the same as that described for the **mkuser** command.

Restrictions on Changing Users

To ensure the integrity of user information, some restrictions apply when using the **chuser** command. Only the root user or users with UserAdmin authorization can use the **chuser** command to perform the following tasks:

- Make a user an administrative user by setting the **admin** attribute to **true**.
- Change any attributes of an administrative user.
- Add a user to an administrative group.

An administrative group is a group with the **admin** attribute set to **true**. Members of the **security** group can change the attributes of non-administrative users and add users to non-administrative groups.

The **chuser** command manipulates local user data only. You cannot use it to change data in registry servers like NIS and DCE.

Flags

-R *load_module* Specifies the loadable I&A module used to change the user's attributes.

Attributes

If you have the proper authority, you can set the following user attributes:

account_locked Indicates if the user account is locked. Possible values include:

- **true** The user's account is locked. The values **yes**, **true**, and **always** are equivalent. The user is denied access to the system.
- **false** The user's account is not locked. The values **no**, **false**, and **never** are equivalent. The user is allowed access to the system. This is the default value.

admin Defines the administrative status of the user. Possible values are:

- **true** The user is an administrator. Only the root user can change the attributes of users defined as administrators.
- **false** The user is not an administrator. This is the default value.

admgroups Lists the groups the user administrates. The *Value* parameter is a comma-separated list of group names.

auditclasses Lists the user's audit classes. The Value parameter is a list of comma-separated classes, or a value of **ALL** to indicate all audit classes.

auth1
Lists the primary methods for authenticating the user. The *Value* parameter is a comma-separated list of *Method*;*Name* pairs. The *Method* parameter is the name of the authentication method. The *Name* parameter is the user to authenticate. If you do not specify a *Name* parameter, the name of the invoking login program is used.

Valid authentication methods are defined in the **/etc/security/login.cfg** file. By default, the SYSTEM method and local password authentication are used. The NONE method indicates that no primary authentication check is made.

auth2
Lists the secondary methods used to authenticate the user. The *Value* parameter is a comma-separated list of *Method*;*Name* pairs. The *Method* parameter is the name of the authentication method. The *Name* parameter value is the user to authenticate.

If this attribute is not specified, the default is NONE, indicating that no secondary authentication check is made. Valid authentication methods are defined in the **/etc/security/login.cfg** file. If you do not specify a *Name* parameter, the name of the invoking login program is used.

capabilities
Defines the system privileges (capabilities) which are granted to a user by the **login** or **su** commands. Valid **capabilities** are:

CAP_AACCT
Performed Advanced Accounting operations.

CAP_ARM_APPLICATION
A process has the ability to use the ARM (Application Response Measurement) services.

CAP_BYPASS_RAC_VMM
A process has the ability to bypass restrictions on VMM resource usage.

CAP_EWLM_AGENT
A process has the ability to use the EWLM (Enterprise Workload Manager™) AIXsystem services. This capability is typically only granted to the userid that runs the EWLM product's Managed Server Component.

CAP_NUMA_ATTACH
A process has the ability to bind to specific resources.

CAP_PROPAGATE
All capabilities are inherited by child processes.

The value is a comma-separated list of zero or more capability names.

core
Specifies the soft limit for the largest core file a user's process can create. The *Value* parameter is an integer representing the number of 512-byte blocks.

core_compress
Enables or disables core file compression. Valid values for this attribute are On and Off. If this attribute has a value of On, compression is enabled; otherwise, compression is disabled. The default value of this attribute is Off.

core_hard
Specifies the largest core file a user's process can create. The Value parameter is an integer representing the number of 512-byte blocks..

core_naming
Selects a choice of core file naming strategies. Valid values for this attribute are On and Off. A value of On enables core file naming in the form `core.pid.time`, which is the same as what the **CORE_NAMING** environment variable does. A value of Off uses the default name of **core**.

core_path
Enables or disables core file path specification. Valid values for this attribute are On and Off. If this attribute has a value of On, core files will be placed in the directory specified by **core_pathname** (the feature is enabled); otherwise, core files are placed in the user's current working directory. The default value of this attribute is Off.

core_pathname
Specifies a location to be used to place core files, if the **core_path** attribute is set to On. If this is not set and **core_path** is set to On, core files will be placed in the user's current working directory. This attribute is limited to 256 characters.

cpu
Identifies the soft limit for the largest amount of system unit time (in seconds) that a user's process can use. The *Value* parameter is an integer. The default value is -1 which turns off restrictions.

cpu_hard Identifies the largest amount of system unit time (in seconds) that a user's process can use. The Value parameter is an integer. The default value is -1 which turns off restrictions.

daemon Indicates whether the user specified by the Name parameter can run programs using the **cron** daemon or the **src** (system resource controller) daemon. Possible values are:

true The user can initiate **cron** and **src** sessions. This is the default.

false The user cannot initiate **cron** and **src** sessions.

data Specifies the soft limit for the largest data segment for a user's process. The *Value* parameter is an integer representing the number of 512-byte blocks. The minimum allowable value for this attribute is 1272.

data_hard Specifies the largest data segment for a user's process. The Value parameter is an integer representing the number of 512-byte blocks. The minimum allowable value for this attribute is 1272.

default_roles Specifies the default roles for the user. The *Value* parameter, a comma-separated list of valid role names, can only contain roles assigned to the user in the **roles** attribute. You can use the **ALL** keyword to signify that the default roles for the user are all their assigned roles.

dictionlist Defines the password dictionaries used by the composition restrictions when checking new passwords.

The password dictionaries are a list of comma-separated absolute path names, evaluated from left to right. All dictionary files and directories must be write protected from all users except root. The dictionary files are formatted one word per line. The word starts in the first column and terminates with a newline character. Only 7 bit ASCII words are supported for passwords. If you install text processing on your system, the recommended dictionary file is the **/usr/share/dict/words** file.

expires Identifies the expiration date of the account. The Value parameter is a 10-character string in the *MMDDhhmmyy* form, where *MM* = month, *DD* = day, *hh* = hour, *mm* = minute, and *yy* = last 2 digits of the years 1939 through 2038. All characters are numeric. If the *Value* parameter is 0, the account does not expire. The default is 0. See the **date** command for more information.

fsize Defines the soft limit for the largest file a user's process can create or extend. The *Value* parameter is an integer representing the number of 512-byte blocks. To make files greater than 2G, specify `-1` or `unlimited`. The minimum value for this attribute is 8192.

fsize_hard Defines the largest file a user's process can create or extend. The Value parameter is an integer representing the number of 512-byte blocks. To make files greater than 2G, specify `-1` or `unlimited`. The minimum value for this attribute is 8192.

gecos Supplies general information about the user specified by the Name parameter. The *Value* parameter is a string with no embedded : (colon) characters and cannot end with the characters '`#!`'.

groups Identifies the groups the user belongs to. The *Value* parameter is a comma-separated list of group names.

histexpire Defines the period of time (in weeks) that a user cannot reuse a password. The value is a decimal integer string. The default is 0, indicating that no time limit is set. Only an administrative user can change this attribute.

histsize Defines the number of previous passwords a user cannot reuse. The value is a decimal integer string. The default is 0. Only an administrative user can change this attribute.

home Identifies the home directory of the user specified by the Name parameter. The *Value* parameter is a full path name.

id Specifies the user ID. The *Value* parameter is a unique integer string. Changing this attribute compromises system security and, for this reason, you should not change this attribute.

login Indicates whether the user can log in to the system with the **login** command. Possible values are:

true The user can log in to the system. This is the default.

false The user cannot log in to the system.

loginretries Defines the number of unsuccessful login attempts allowed after the last successful login before the system locks the account. The value is a decimal integer string. A zero or negative value indicates that no limit exists. Once the user's account is locked, the user will not be able to log in until the system administrator resets the user's unsuccessful_login_count attribute in the **/etc/security/lastlog** file to be less than the value of loginretries. To do this, enter the following:

```
chsec -f /etc/security/lastlog -s username -a \
unsuccessful_login_count=0
```

logintimes

Defines the days and times that the user is allowed to access the system. The value is a comma-separated list of entries in one of the following formats:

```
[!]:<time>-<time>

[!]<day>[-<day>][:<time>-<time>]

[!]<month>[<daynum>][-<month>[<daynum>]][:<time>-<time>]
```

Possible values for <day> include mon, tues, w, THU, Friday, sat, and SUNDAY. Indicate the day value as any abbreviated day of the week; however, the abbreviation must be unique with respect to both day and month names. The range of days can be circular, such as Tuesday-Monday. Day names are case insensitive.

Possible values for <time> include times specified in 24-hour military format. Precede the time value with a : (colon) and specify a string of 4 characters. Leading zeros are required. Thus, 0800 (8am) is valid while 800 is not valid. An entry consisting of only a specified time period applies to every day. The start hour must be less than the end hour. The time period cannot flow into the next day.

Possible values for <month> include Jan, F, march, apr, and s. Indicate the month value as any abbreviated month; however, the abbreviation must be unique with respect to both day and month names. The range of months can be circular, such as September-June. Month names are case insensitive.

Possible values for <daynum> include days 1-31 of a month. This value is checked against the specified month. Specify the month value as either a 1 or 2 character string. A month specified without a daynum value indicates the first or last day of the month, depending on if the month is the start or end month specified, respectively.

Entries prefixed with ! (exclamation point) deny access to the system and are called DENY entries. Entries without the ! prefix allow access and are called ACCESS entries. The ! prefix applies to single entries and must prefix each entry. Currently, the system allows 200 entries per user.

This attribute is internationalized. Month and day names can be entered and are displayed in the language specified by the locales variables set for the system. The relative order of the month and day values are also internationalized; the <month><daynum> and <daynum><month> formats are accepted.

The system evaluates entries in the following order:

1. All DENY entries. If an entry matches the system time, the user is denied access and the ALLOW entries are not processed.
2. All ALLOW entries, if no DENY entries exist. If an ALLOW entry matches the system time, the user is allowed access. If an ALLOW entry does not match the system time, the user is denied access. If no ALLOW entry exists, the user is permitted to log in.

maxage

Defines the maximum age (in weeks) of a password. The password must be changed by this time. The value is a decimal integer string. The default is a value of 0, indicating no maximum age. Range: 0 to 52

maxexpired Defines the maximum time (in weeks) beyond the **maxage** value that a user can change an expired password. After this defined time, only an administrative user can change the password. The value is a decimal integer string. The default is -1, indicating restriction is set. If the **maxexpired** attribute is 0, the password expires when the **maxage** value is met. If the **maxage** attribute is 0, the **maxexpired** attribute is ignored. Range: 0 to 52 (a root user is exempt from **maxexpired**)

maxrepeats Defines the maximum number of times a character can be repeated in a new password. Since a value of 0 is meaningless, the default value of 8 indicates that there is no maximum number. The value is a decimal integer string. Range: 0 to 8

maxulogs Specifies the maximum number of concurrent logins per user. If the concurrent login number for a user exceeds the maximum number of allowed logins, the login is denied.

minage Defines the minimum age (in weeks) a password must be before it can be changed. The value is a decimal integer string. The default is a value of 0, indicating no minimum age. Range: 0 to 52

minalpha Defines the minimum number of alphabetic characters that must be in a new password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number. Range: 0 to 8

mindiff Defines the minimum number of characters required in a new password that were not in the old password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number. Range: 0 to 8

minlen Defines the minimum length of a password. The value is a decimal integer string. The default is a value of 0, indicating no minimum length. The maximum value allowed is 8. This attribute is determined by by **minlen** and/or **'minalpha + minother'**, whichever is greater. **'minalpha + minother'** should never be greater than 8. If **'minalpha + minother'** is greater than 8, then the effective value for **minother** is reduced to **'8 - minalpha'**.

minother Defines the minimum number of non-alphabetic characters that must be in a new password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number. Range: 0 to 8

nofiles Defines the soft limit for the number of file descriptors a user process may have open at one time. The **Value** parameter is an integer.

nofiles_hard Defines the hard limit for the number of file descriptors a user process may have open at one time. The **Value** parameter is an integer. The default value is -1, which sets the limit to the maximum allowed by the system.

nproc Defines the soft limit on the number of processes a user can have running at one time. The *Value* parameter is an integer equal to or greater than 1. The default value is -1, which sets the limit to the maximum allowed by the system.

nproc_hard Defines the hard limit on the number of processes a user can have running at one time. The *Value* parameter is an integer equal to or greater than 1. The default value is -1, which sets the limit to the maximum allowed by the system.

pgrp Identifies the user's primary group. The *Value* parameter must contain a valid group name and cannot be a null value.

projects Defines the list of projects to which the user's processes can be assigned. The value is a list of comma-separated project names and is evaluated from left to right. The project name should be a valid project name as defined in the system. If an invalid project name is found on the list, it will be reported as an error.

pwdchecks Defines the password restriction methods enforced on new passwords. The value is a list of comma-separated method names and is evaluated from left to right. A method name is either an absolute path name or a path name relative to **/usr/lib** of an executable load module.

pwdwarntime Defines the number of days before the system issues a warning that a password change is required. The value is a decimal integer string. A zero or negative value indicates that no message is issued. The value must be less than the difference of the **maxage** and **minage** attributes. Values greater than this difference are ignored and a message is issued when the **minage** value is reached.

rcmds Controls the remote execution of the **r**-commands (**rsh**, **rexec**, and **rcp**). Possible values are as follows:

allow Allows this user to perform remote command execution. This is the default value.

deny Denies this user the ability to use remote command execution.

hostlogincontrol
Specifies that the ability of remote command execution is determined by the **hostsallowedlogin** and **hostsdeniedlogin** attributes. The user is only allowed to execute remote commands on a target system if the user (or target user) is allowed to log in the target system. This value is typically used for users defined in a centralized user database, such as LDAP, where the user might be allowed to log in to some systems but not others.

Note: The **rcmds** attribute controls only remote command execution. It does not control **r**-command functionality to open a remote shell. Login functions such as this are controlled by the **rlogin**, **hostsallowedlogin**, and **hostsdeniedlogin** attributes.

Although the deprecated **ttys** attribute value `!rsh`, which is effectively the same as setting the **rcmds** attribute to **deny**, is still supported for purposes of backward compatibility, the **rcmds** attribute should be used instead to control the execution of **r**-commands.

rlogin Permits access to the account from a remote location with the **telnet** or **rlogin** commands. Possible values are:

true The user account can be accessed remotely. This is the default **rlogin** value.

false The user cannot be accessed remotely.

roles Lists the administrative roles for this user. The *Value* parameter is a list of role names, separated by commas.

rss The soft limit for the largest amount of physical memory a user's process can allocate. The *Value* parameter is a decimal integer string specified in units of 512-byte blocks. This value is not currently enforced by the system.

rss_hard The largest amount of physical memory a user's process can allocate. The Value parameter is a decimal integer string specified in units of 512-byte blocks. This value is not currently enforced by the system.

shell Defines the program run for the user at session initiation. The *Value* parameter is a full path name.

stack Specifies the soft limit for the largest process stack segment for a user's process. The *Value* parameter is an integer representing the number of 512-byte blocks to allot. The minimum allowable value for this attribute is 49.

stack_hard Specifies the largest process stack segment of a user's process. The *Value* parameter is an integer representing the number of 512-byte blocks to allot. The minimum allowable value for this attribute is 49. The largest allowable value for this parameter is 2147483647.

su Indicates whether another user can switch to the specified user account with the **su** command. Possible values are:

true Another user can switch to the specified account. This is the default.

false Another user cannot switch to the specified account.

sugroups Lists the groups that can use the **su** command to switch to the specified user account. The *Value* parameter is a comma-separated list of group names, or a value of **ALL** to indicate all groups. An ! (exclamation point) in front of a group name excludes that group. If this attribute is not specified, all groups can switch to this user account with the **su** command.

sysenv Identifies the system-state (protected) environment. The *Value* parameter is a set of comma-separated *Attribute*=*Value* pairs as specified in the **/etc/security/environ** file.

threads Specifies the soft limit for the largest number of threads that a user process can create. The *Value* parameter is an integer equal to or greater than 1, representing the number of threads each user process can create. This limit is enforced by both the kernel and the user space **pthread** library.

threads_hard Specifies the largest possible number of threads that a user process can create. The *Value* parameter is an integer equal to or greater than 1, representing the number of threads each user process can create. This limit is enforced by both the kernel and the user space **pthread** library.

tpath Indicates the user's trusted path status. The possible values are:

- **always** The user can only execute trusted processes. This implies that the user's initial program is in the trusted shell or some other trusted process.
- **no tsh** The user cannot invoke the trusted shell on a trusted path. If the user enters the secure attention key (SAK) after logging in, the login session ends.
- **nosak** The secure attention key (SAK) is disabled for all processes run by the user. Use this value if the user transfers binary data that may contain the SAK sequence. This is the default value.
- **on** The user has normal trusted path characteristics and can invoke a trusted path (enter a trusted shell) with the secure attention key (SAK).

ttys Lists the terminals that can access the account specified by the *Name* parameter. The *Value* parameter is a comma-separated list of full path names, or a value of **ALL** to indicate all terminals. An ! (exclamation point) in front of a terminal name excludes that terminal. If this attribute is not specified, all terminals can access the user account.

umask Determines file permissions. This value, along with the permissions of the creating process, determines a file's permissions when the file is created. The default is 022.

usrenv Defines the user-state (unprotected) environment. The *Value* parameter is a set of comma-separated *Attribute*=*Value* pairs as specified in the **/etc/security/environ** file.

efs_keystore_access Specifies the database type of the user keystore. You can specify the following values:

- **file** Creates the **/var/efs/users/***usrname***/keystore** keystore file associated with the user.
- **none** Keystore is not created. All the other keystore attributes have no effect.

The default value is **file**.

Restriction: The attribute is valid only when the system is EFS-enabled.

efs_adminks_access Represents the database type for the **efs_admin** keystore. The only valid value is **file**.

Restriction: The attribute is valid only when the system is EFS-enabled.

efs_initialks_mode

Specifies the initial mode of the user keystore. You can specify the following values:

admin Root or other security privileged system users can open the keystore using the admin key and reset the keystore password.

guard Root users cannot open the keystore using the admin key or reset the keystore password.

The default value is **admin**.

The attribute specifies the initial mode of the user keystore. You can use the attribute with the **mkuser** command. After the keystore has been created, changing the attribute value with the **chuser**, **chgroup**, or **chsec** command, or manual editing does not change the mode of the keystore unless the keystore is deleted and a new one is created. To change the keystore mode, use the **efskeymgr** command.

Restriction: The attribute is valid only when the system is EFS-enabled.

efs_allowksmodechangebyuser

Specifies whether the mode can be changed. You can specify the following values:

- **yes**
- **no**

The default value is **yes**.

Restriction: The attribute is valid only when the system is EFS-enabled.

efs_keystore_algo

Specifies the algorithm that is used to generate the private key of the user during the keystore creation. You can specify the following values:

- **RSA_1024**
- **RSA_2048**
- **RSA_4096**

The default value is **RSA_1024**.

You can use the attribute with the **mkuser** command. After the keystore has been created, changing the value of this attribute with the **chuser**, **chgroup**, or **chsec** command, or manual editing does not regenerate the private key unless the keystore is deleted and a new one is created. To change the algorithm for the keys, use the **efskeymgr** command.

Restriction: The attribute is valid only when the system is EFS-enabled.

efs_file_algo

Specifies the encryption algorithm for user files. You can specify the following values:

- **AES_128_CBC**
- **AES_128_ECB**
- **AES_192_CBC**
- **AES_192_ECB**
- **AES_256_CBC**
- **AES_256_ECB**

The default value is **AES_128_CBC**.

Restriction: The attribute is valid only when the system is EFS-enabled.

minsl

Defines the minimum sensitivity-clearance level that the user can have. **Note:** This attribute is valid only for Trusted AIX. See Trusted AIX in the *Security* for more information.The valid values are defined in the "Clearances" section of the **/etc/security/enc/LabelEncodings** file for the system. The value must be defined in quotation marks if it has white spaces. The **minsl** value must be dominated by the **defsl** value for the user.

maxsl Defines the maximum sensitivity-clearance level that the user can have. **Note:** This attribute is valid only for Trusted AIX. See Trusted AIX in the *Security* for more information.The valid values are defined in the "Clearances" section of the **/etc/security/enc/LabelEncodings** file. The value must be defined in quotation marks if it has white spaces. The **maxsl** value must dominate the **defsl** value for the user.

defsl Defines the default sensitivity level that the user is assigned during login. **Note:** This attribute is valid only for Trusted AIX. See Trusted AIX in the *Security* for more information.The valid values are defined in the "Clearances" section of the **/etc/security/enc/LabelEncodings** file. The value must be defined in quotation marks if it has white spaces. The **defsl** value must dominate the **minsl** value and be dominated by the **maxsl** value.

mintl Defines the minimum integrity clearance level that the user can have. **Note:** This attribute is valid only for Trusted AIX. See Trusted AIX in the *Security* for more information.The valid values are defined in the "Sensitivity labels" section of the **/etc/security/enc/LabelEncodings** file . If the optional "Integrity labels" section is defined in the **/etc/security/enc/LabelEncodings** file, the value must be from this section. The value must be defined in quotation marks if it contains white spaces. The **mintl** value must be dominated by the **deftl** value for the user.

maxtl Defines the maximum integrity clearance level that the user can have. **Note:** This attribute is valid only for Trusted AIX. See Trusted AIX in the *Security* for more information.The valid values are defined in the "Sensitivity labels" section of the **/etc/security/enc/LabelEncodings** file . If the optional "Integrity labels" section is defined in the **/etc/security/enc/LabelEncodings** file, the value must be from this section. The value must be defined in quotation marks if it contains white spaces. The **maxtl** value must dominate the **deftl** value for the user.

deftl Defines the default integrity clearance level that the user is assigned during login. **Note:** This attribute is valid only for Trusted AIX. See Trusted AIX in the *Security* for more information.The valid values are defined in the "Sensitivity labels" section of the **/etc/security/enc/LabelEncodings** file . If the optional "Integrity labels" section is defined in the **/etc/security/enc/LabelEncodings** file, the value must be from this section. The value must be defined in quotation marks if it contains white spaces. The **deftl** value must dominate the **mintl** value and be dominated by the **maxtl** value.

Security

Access Control

This command must grant execute (x) access only to the root user and the security group. This command must be installed as a program in the trusted computing base (TCB). The command must be owned by the root user with the **setuid** (SUID) bit set.

On a Trusted AIX system, only users with the aix.mls.clear.write authorization can modify the attributes **minsl, maxsl, defsl, mintl, maxtl** and **deftl**.

Auditing Events

Event	Information
USER_Change	user, attributes

Files Accessed

Mode	File
rw	**/etc/passwd**

Mode	File
rw	/etc/security/user
rw	/etc/security/user.roles
rw	/etc/security/limits
rw	/etc/security/environ
rw	/etc/security/audit/config
rw	/etc/group
rw	/etc/security/group
r	/etc/security/enc/LabelEncodings

RBAC Disclaimer 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**

Limitations

Changing a user's attributes may not be supported by all loadable I&A modules. If the loadable I&A module does not support changing a user's attributes, an error is reported.

Examples

1. To enable user `smith` to access this system remotely, type:
   ```
   chuser rlogin=true smith
   ```
2. To change the expiration date for the `davis` user account to 8 a.m., 1 May, 1995, type:
   ```
   chuser expires=0501080095 davis
   ```
3. To add `davis` to the groups `finance` and `accounting`, type:
   ```
   chuser groups=finance,accounting davis
   ```
4. To change the user `davis`, who was created with the LDAP load module, to not be allowed remote access, type:
   ```
   chuser -R LDAP rlogin=false davis
   ```

Files

/usr/bin/chuser	Contains the **chuser** command.
/etc/passwd	Contains the basic attributes of users.
/etc/group	Contains the basic attributes of groups.
/etc/security/group	Contains the extended attributes of groups.
/etc/security/user	Contains the extended attributes of users.
/etc/security/user.roles	Contains the administrative role attributes of users.
/etc/security/lastlog	Contains the last login attributes of users.
/etc/security/limits	Defines resource quotas and limits for each user.
/etc/security/audit/config	Contains audit configuration information.
/etc/security/environ	Contains the environment attributes of users.
/etc/security/enc/LabelEncodings	Contains the label definitions for the Trusted AIX system.

Related Information

The **chfn** command, **chgroup** command, **chgrpmem** command, **chsh** command, **lsgroup** command, **lsuser** command, **mkgroup** command, **mkuser** command, **passwd** command, **pwdadm** command, **rmgroup** command, **rmuser** command, **setgroups** command, **setsenv** command, **su** command.

For information on installing the Web-based System Manager, see Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

Securing the network in *Security*.

Trusted AIX in the *Security*.

chusil Command

Purpose
Changes an attribute of an existing user-specified installation location (USIL) instance.

Syntax
chusil -R *RelocatePath* **-c** *NewComments* [**-X**]

Description
The **chusil** command changes an attribute of an existing USIL instance.

Flags

-c *NewComments*	Specifies the new comments to include in the USIL definition (visible with the **lsusil** command).
-R *RelocatePath*	Specifies the path to an existing USIL location.
-X	Expands the space needed automatically.

Security
RBAC Disclaimer

Files

/usr/sbin/chusil	Contains the **chusil** command.

Related Information
The **mkusil**, **lsusil**, and **rmusil** commands.

chvfs Command

Purpose
Changes entries in the **/etc/vfs** file.

Syntax
chvfs *VFSEntry*

Description
The **chvfs** command changes **/etc/vfs** file entries by specifying the following fields within the *VFSEntry* parameter. The *VFSEntry* parameter is composed of the following fields:
VFSName:*VFSNumber*:*MountHelper*:*FileSystemHelper*.

Any of the entries in the *VFSEntry* can be null (empty), with the exception of the *VFSName* field, and the corresponding value will not be changed. If all of the arguments are satisfactory, the entry in the **/etc/vfs** file is changed.

Parameters

VFSEntry	A string in the following format: *VFSName*:*VFSNumber*:*MountHelper*:*FileSystemHelper*
VFSName	Specifies the name of a virtual file system type.
VFSNumber	Specifies the virtual file system type's internal number as known by the kernel.
MountHelper	Specifies the name of the backend used to mount a file system of this type.
FileSystemHelper	Specifies the name of the backend used by certain file system specific commands to perform operations on a file system of this type.

Security

RBAC Disclaimer

Examples

To change the *FileSystemHelper* for the vfs entry named `newvfs`, enter:

```
chvfs "newvfs:::/etc/helper/testhelper"
```

The missing parameters are left unchanged.

Files

/etc/vfs	Contains descriptions of virtual file system types.

Related Information

The **crvfs** command, **lsvfs** command, **mount** command, and **rmvfs** command.

The File systems in *Operating system and device management*explains file system types, management, structure, and maintenance.

---

chvg Command

Purpose

Sets the characteristics of a volume group.

Syntax

chvg [**-s** *Sync* { **y** | **n** }] [**-h** *Hotspare* {**y** | **Y** | **n** | **r** }] [**-a** *AutoOn* { **n** | **y** }] [**-c** | **-l**] [**-L** *LTGSize*] [**-Q** { **n** | **y** }] [**-u**] [**-x** { **n** | **y** }] [**-S** | **-R**] [**-t** [*factor*]] [**-B** | **-G**] [**-P**] [**-v**][**-C**] [**-f**] [**-g**] [**-b** { **n** | **y** }] [**-I**]*VolumeGroup*

Description

The **chvg** command changes the characteristics of a volume group.

You can use the Volumes application in Web-based System Manager to change volume characteristics. You could also use the System Management Interface Tool (SMIT) **smit chvg** fast path to run this command.

Note: On AIX 5.2 and later only Enhanced Concurrent Capable volume groups will be created when the **-c** or **-C** flags are specified.

Flags

Notes:

1. Only the **-a**, **-R**, **-S**, **-u**, and **-h** options are allowed on the volume group that has a snapshot volume group.
2. Only the **-a**, **-R**, **-S**, and **-u** options are allowed on the snapshot volume group.

-a *AutoOn*
Determines if the volume group is automatically activated during system startup. The *AutoOn* variable can be either of the following:

n The volume group is not automatically activated during system startup.

y The volume group is automatically activated during system startup.

-b
Sets the bad-block relocation policy of a volume group. The default value is `yes`.

y Will turn on the bad-block relocation policy of a volume group.

n Turns off the bad block relocation policy of a volume group.

-B
Changes the volume group to Big VG format. This can accommodate up to 128 physical volumes and 512 logical volumes.

Notes:

1. The **-B** flag cannot be used if there are any stale physical partitions.
2. Once the volume group is converted, it cannot be imported into AIX 4.3.1 or lower versions.
3. The **-B** flag cannot be used if the volume group is varied on in concurrent mode.
4. There must be enough free partitions available on each physical volume for the VGDA expansion for this operation to be successful.
5. Because the VGDA resides on the edge of the disk and it requires contiguous space for expansion, the free partitions are required on the edge of the disk. If those partitions are allocated for user usage, they will be migrated to other free partitions on the same disk. The rest of the physical partitions will be renumbered to reflect the loss of the partitions for VGDA usage. This will change the mappings of the logical to physical partitions in all the PVs of this VG. If you have saved the mappings of the LVs for a potential recovery operation, you should generate the maps again after the completion of the conversion operation. Also, if the backup of the VG is taken with the map option and you plan to restore using those maps, the restore operation may fail since the partition number may no longer exist (due to reduction). It is recommended that backup is taken before the conversion, and right after the conversion if the map option is utilized.
6. Because the VGDA space has been increased substantially, every VGDA update operation (creating a logical volume, changing a logical volume, adding a physical volume, and so on) may take considerably longer to run.

-c
Same as **-C** flag. In AIX 5.2 and later only Enhanced Concurrent Capable volume groups will be created.

-C Changes the volume group into an Enhanced Concurrent Capable volume group. Changes the volume group varied on in non-concurrent mode to Enhanced Concurrent Capable. This requires that the volume group be re-imported on all other nodes prior to activation in Enhanced Concurrent mode. Changes the volume group varied on in Concurrent mode to an Enhanced Concurrent mode volume group. Only use the **-C** flag with the HACMP™ ES. It has no effect on volume groups and systems not using the HACMP ES product.

Enhanced Concurrent volume groups use Group Services. Group Services ships with HACMP ES and must be configured prior to activating a volume group in this mode.

Use this flag to change a volume group into an Enhanced Concurrent Capable volume group.

Notes:

1. Enhanced Concurrent volume groups use Group Services. Group Services ships with HACMP ES and must be configured prior to 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.

-f 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.

-g Will examine all the disks in the volume group to see if they have grown in size. If any disks have grown in size attempt to add additional PPs to PV. If necessary will determine proper 1016 multiplier and conversion to big vg.

Notes:

1. The user might be required to execute **varyoffvg** and then **varyonvg** on the volume group for LVM to see the size change on the disks.
2. There is no support for re-sizing while the volume group is activated in classic or enhanced concurrent mode.
3. There is no support for re-sizing for the rootvg.

-G Changes the volume group to Scalable VG format. This can accommodate up to 1024 physical volumes and 4096 logical volumes.

Notes:

1. The **-G** flag cannot be used if there are any stale physical partitions.
2. Once the volume group is converted, it cannot be imported into AIX 5.2 or lower versions.
3. The **-G** flag cannot be used if the volume group is varied on.
4. There must be enough free partitions available on each physical volume for the VGDA expansion for this operation to be successful.
5. Since the VGDA resides on the edge of the disk and it requires contiguous space for expansion, the free partitions are required on the edge of the disk. If those partitions are allocated for user usage, they will be migrated to other free partitions on the same disk. The rest of the physical partitions will be renumbered to reflect the loss of the partitions for VGDA usage. This will change the mappings of the logical to physical partitions in all the PVs of this VG. If you have saved the mappings of the LVs for a potential recovery operation, you should generate the maps again after the completion of the conversion operation. Also, if the backup of the VG is taken with the map option and you plan to restore using those maps, the restore operation may fail since the partition number may no longer exist (due to reduction). It is recommended that backup is taken before the conversion, and right after the conversion if the map option is utilized.
6. Because the VGDA space has been increased substantially, every VGDA update operation (creating a logical volume, changing a log ical volume, adding a physical volume, and so on) may take considerably longer to run.

-h *Hotspare* Sets the sparing characteristics for the volume group specified by the *VolumeGroup* parameter. Either allows (**y**) the automatic migration of failed disks, or prohibits (**n**) the automatic migration of failed disks. This flag has no meaning for non-mirrored logical volumes

y Enhances the automatic migration of failed disks by permitting one for one migration of partitions from one failed disk to one spare disk. The smallest disk in the volume group spare pool that is big enough for one to one migration will be used.

Y Permits the automatic migration of failed disks and allows migration to the entire pool of spare disks, as opposed to a one for one migration of partitions to a spare.

n Prohibits the automatic migration of a failed disk. This is the default value for a volume group.

r Removes all disks from the *Hotspare* pool for the volume group.

Note: This flag is not supported for the concurrent capable volume groups.

-I Modifies the volume group so that it can be imported to AIX 5.1 and AIX 5.2. The *LTGSize* will behave as if the volume group had been created prior to AIX 5.3. This operation might fail if the volume group contains striped logical volumes whose strip size (a strip size multiplied by the number of disks in an array equals the stripe size) is larger than the supported strip size on AIX 5.1 or AIX 5.2. If logical volumes are later created with a strip size that is larger than the supported strip size on AIX 5.1 or AIX 5.2, then attempting to import the volume group back to AIX 5.1 or AIX 5.2 is not supported.

-l Changes the volume group into a Non-Concurrent Capable volume group. The volume group must be varied on in non-concurrent mode for this command to take effect.

-L For volume groups created on AIX 5.3, the **-L** flag is ignored. When the volume group is varied on, the logical track group size will be set to the common max transfer size of the disks.

For volume groups created prior to AIX 5.3, the **-L** flag changes the logical track group size, in number of kilobytes, of the volume group. The value of the *LTGSize* parameter must be `0`, `128`, `256`, `512`, or `1024`. In addition, it should be less than or equal to the maximum transfer size of all disks in the volume group. The default size is `128` kilobytes. An *LTGSize* of `0` will cause the next varyonvg to set the logical track group size to the common max transfer size of the disks.

-P *PhysicalPartitions* Increases the number of physical partitions a volume group can accommodate. Where the *PhysicalPartitions* variable is represented in units of 1024 partitions. Valid values are 64, 128, 256, 512 768, 1024 and 2048. The value should be larger than the current value or no action is taken. This option is only valid with Scalable-type volume groups. This option is not available for volume groups varied on in concurrent mode.

-Q Determines if the volume group is automatically varied off after losing its quorum of physical volumes. The default value is `yes`. The change becomes effective immediately.

n The volume group stays active until it loses all of its physical volumes.

y The volume group is automatically varied off after losing its quorum of physical volumes.

-R Resumes normal I/O operations for a volume group.

-s *Sync* Sets the synchronization characteristics for the volume group specified by the *VolumeGroup* parameter. Either permits (**y**) the automatic synchronization of stale partitions or prohibits (**n**) the automatic synchronization of stale partitions. This flag has no meaning for non-mirrored logical volumes. Automatic synchronization is a recovery mechanism that will only be attempted after the LVM device driver logs **LVM_SA_STALEPP** in the **errpt**. A partition that becomes stale through any other path (for example, **mklvcopy**) will not be automatically resynced.

y Attempts to automatically synchronize stale partitions.

n Prohibits automatic synchronization of stale partitions. This is the default for a volume group.

Note: This flag is not supported for the concurrent capable volume groups.

-S Drains I/O's for this volume group and suspends future I/O's.

-t [*factor*] Changes the limit of the number of physical partitions per physical volume, specified by *factor*. *factor* should be between 1 and 16 for 32 disk volume groups and 1 and 64 for 128 disk volume groups.

If *factor* is not supplied, it is set to the lowest value such that the number of physical partitions of the largest disk in volume group is less than *factor* x 1016.

If *factor* is specified, the maximum number of physical partitions per physical volume for this volume group changes to *factor* x 1016.

Notes:

1. This option is ignored for Scalable-type volume groups.
2. If the volume group is created in AIX 4.1.2 in violation of 1016 physical partitions per physical volume limit, this flag can be used to convert the volume group to a supported state. This will ensure proper stale/fresh marking of partitions.
3. *factor* cannot be changed if there are any stale physical partitions in the volume group.
4. Once volume group is converted, it cannot be imported into AIX 4.3 or lower versions.
5. This flag cannot be used if the volume group is varied on in concurrent mode.
6. The maximum number of physical volumes that can be included in this volume group will be reduced to (MAXPVS/factor).

-u Unlocks the volume group. This option is provided if the volume group is left in a locked state by abnormal termination of another LVM operation (such as the command core dumping, or the system crashing).
Note: Before using the **-u** flag, make sure that the volume group is not being used by another LVM command.

-v *LogicalVolumes* Increases the number of logical volumes that can be created. Valid values are 512, 1024, 2048 and 4096. The value should be larger than the current value or no action is taken. This option is only valid with Scalable-type volume groups. This option is not available for volume groups varied on in concurrent mode.

-x Changes the mode which the Concurrent Capable volume group is varied on. The volume group must be varied on in non-concurrent mode for this command to take effect.
Note: There is no auto on support for Enhanced Concurrent Capable volume groups. On AIX 5.2 and later only Enhanced Concurrent Capable volume groups will be created.

y autovaryon the volume group in concurrent mode.

n autovaryon the volume group in non-concurrent mode.

Note: If the volume group is not created Concurrent Capable, this command has no effect on the volume group.

In order for this auto-varyon into concurrency of the volume group to take effect, you must enter the following line into the **/etc/inittab** file:

```
rc_clvmv:2:wait:/usr/sbin/clvm_cfg 2>&1
```

Attention: This entry must be added after the entry used to initiate **srcmstr**.
—

Examples

1. To cause volume group `vg03` to be automatically activated during system startup, type:

   ```
   chvg  -a y vg03
   ```

2. To change the volume group `vg03` to a supported state if it is in violation of 1016 physical partitions per physical volume limit, type:

   ```
   chvg  -t  vg03
   ```

3. To change the maximum number of physical partitions per physical volume to 2032 and maximum number of physical volumes in volume group `vg03` to 16, type:

```
chvg  -t 2 vg03
```

Files

/usr/sbin Directory where the **chvg** command resides.

Related Information

Commands: **bosboot**, **lsvg**, **mkvg**, **savebase**, **varyonvg**.

The Logical volume storage in *Operating system and device management* explains the Logical Volume Manager, physical volumes, logical volumes, volume groups, organization, ensuring data integrity, and allocation characteristics.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

The System management interface tool in *Operating system and device management* explains the structure, main menus, and tasks that are done with SMIT.

---

chvirprt Command

Purpose

Changes the attribute values of a virtual printer.

Syntax

chvirprt -d *QueueDeviceName* **-q** *PrintQueueName* [**-a** *Attribute*=*Value* ...]

Description

The **chvirprt** command changes attribute values for the virtual printer assigned to *PrintQueueName* and *QueueDeviceName*.

> **Note:** Attribute names for default values of the **qprt** command line flags can be specified by entering the flag letters. For example, to change the default value for the **-w** flag (page width) to 132, enter `w=132`. All other attribute names must be 2 characters long.

You can use the Printer Queues application in Web-based System Manager (wsm) to change printer characteristics. You could also use the System Management Interface Tool (SMIT) **smit chvirprt** fast path to run this command.

Flags

-a *Attribute*=*Value*	Replaces the value for *Attribute* with *Value*. If *Value* contains one or more spaces, it must be surrounded by quotes ('*Value*'). be the last flag when entering the **chvirprt** command on the command line.
-d *QueueDeviceName*	Specifies the name of the queue device to which the virtual printer is assigned.
-q *PrintQueueName*	Specifies the name of the print queue to which the virtual printer is assigned.

Examples

To change the default page width to 132 characters (the **w** attribute) and specify that user mary receives the ″intervention required″ messages (the **si** attribute) for the virtual printer associated with the proq print queue and the mypro queue device, enter:

```
chvirprt  -q proq  -d mypro  -a si=mary w=132
```

Files

/etc/qconfig	Configuration file
/usr/sbin/chvirprt	Contains the **chvirprt** command.
/var/spool/lpd/pio/@local/custom/*	Virtual printer attribute files
/var/spool/lpd/pio/@local/ddi/*	Digested virtual printer attribute files.

Related Information

The **lsvirprt** command, **mkvirprt** command, **qprt** command, **rmvirprt** command, **smit** command.

The **qconfig** file.

Changing or Showing Characteristics of a Virtual Printer in *Printers and printing*.

Printer colon file conventions in *Printers and printing*.

Printer-specific information in *Printers and printing*.

Installing support for additional printers in *Printers and printing*.

Virtual printer definitions and attributes in *Printers and printing*.

Adding a printer using the printer colon file in *Printers and printing*.

Printer code page translation tables in *Printers and printing*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

---

chvmode Command

Purpose

Changes the current output device and viewport size of the X server.

> **Note:** This command is usable only while the X server is running.

Syntax

chvmode [{ **+** | **-** } **l**] [{ **+** | **-** } **c**] [**-vsize** *Width***x***Height* [@ VSync]

Description

The **chvmode** command changes the current output device and viewport size used by the X server.

Viewport size specification is usable only for a CRT display and its resolution has panning option.

You can use the Devices application in Web-based System Manager (wsm) to change device characteristics. You could also use the System Management Interface Tool (SMIT) to run this command.

Flags

+/-c	Enables or disables CRT output.
+/-l	Enables or disables LCD output.
-vsize *Width***x***Height*[@VSync]	Specifies viewport size of CRT display and the vertical synchronization (refresh rate in Hz). If **@***VSync* is not specified, the current vertical synchronization frequency is used.

Security

Access Control: Any User

Auditing Events: None

Exit Status

The following exit values are returned:

0	Successful completion.
>0	An error occurred.

Examples

1. To disable the LCD panel and enable the CRT display, enter:
   ```
   chvmode -l +c
   ```
2. To change the current CRT viewport to be 1024x768, enter:
   ```
   chvmode -vsize 1024x768
   ```
3. To specify VGA mode with high refresh rate of 75Hz, enter:
   ```
   chvmode -vsize 640x480@75
   ```

Files

/usr/bin/X11/chvmode	Contains the **chvmode** command.

Related Information

The **lsvmode** command.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

---

chwpar Command

Purpose

Changes the characteristics of a workload partition.

Syntax

/usr/sbin/chwpar [**-a**] [**-A**] [**-c**] [**-d** *directory*] [**-D** *attribute*=*value* ...] ... [**-F**] [**-h** *hostname*] [**-i**] [**-I** *attribute*=*value* ...] ... [**-n** *newname*] [**-N** *attribute*=*value* ...] ... [**-P**] [**-R** *attribute*=*value* ...] [**-S** *attribute*[+|-]=*value*...] [**-u** *userscript*] [**-v**] *wparname*

/usr/sbin/chwpar -K [**-A**] [**-c**] [**-D devname**=*devicepathname*] ... [**-F**] [**-i**] [**-I rtdest**=*destination* **rtgateway**=*gateway* [*attribute*=*value* ...]] ... [**-N address**=*A.B.C.D*] ... [**-R** [*attribute* ...]] [**-S**] [**-u**] [**-v**] *wparname*

Description

The **chwpar** command modifies the configuration options of the workload partition specified by the *wparname* parameter. You can change most options whether the workload partition is running. Some changes to the running workload partitions are detected and disallowed (see the **-d** and **-n** options). Other changes, such as unexporting a busy device or removing a mounted file system, might generate errors on a running workload partition, but you can make these changes.

Use the **-K** flag to remove characteristics from the configuration of a workload partition. For an attribute with a default option, removing the value for the attribute restores the default setting for the option.

Flags

-a
Automatically resolves conflicting static settings if required. Settings that can be resolved are hostname and network configuration.

-A
Configures the workload partition to be started at system boot through the **/etc/rc.wpars** command by setting the **auto** attribute value of the workload partition to **yes**. When you specify the **-A** flag with the **-K** flag, the **auto** attribute value is set to **no**. The **-A** flag takes effect the next time the global system boots. The **-A** flag is not valid for application workload partitions.

-c
The workload partition is enabled for checkpoint.
Note: The capability to enable a workload partition for checkpoint depends on additional software.

-d *directory*
Changes the base directory for the workload partition. The **-d** flag can not be used on a running workload partition. This flag is not valid for application workload partitions.

-D [**devname**=*device name* | **devid**=*device identifier*] [**rootvg**=*yes* | *no*] [**devtype**=[*clone* | *pseudo* | *disk* | *cdrom* | *tape*]]
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 is the full path to the device (i.e. /dev/pty10). 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 only applies to disk, cdrom, or tape type devices.

rootvg= [yes | no]
Used to indicate if the specified disk device is to be used as a rootvg workload partition device. If the **rootvg** attribute is not specified, the command will take the default of no.

devtype=[clone | pseudo | disk | cdrom | tape]
Specifies the device type of the device to allocate to the workload partition.

-F
Suppresses failures due to settings that are not valid.

-h *hostname*
Modifies the kernel host name of the workload partition.

-i Enables WPAR-specific routing for the workload partition. When WPAR-specific routing is enabled on a running workload partition, any explicit routing table entries that were configured using the **-I** flag with the **mkwpar**, **wparexec**, or **chwpar** command are added to the routing table of the workload partition. Running the **chwpar -i** command on a workload partition with enabled WPAR-specific routing refreshes the routing table of the workload partition. You can use the **-i** flag, for example, to restore the routing table after a global route flush. You can use the **-i** flag with the **-K** flag to disable WPAR-specific routing for the workload partition. For more information about the **-i** flag, see the description of the **-i** flag of the **mkwpar** command.

-I *attribute*=*value* ... Modifies explicit routing table entries. Entries are matched based on the combination of the **rtdest**, **rtgateway**, and **rtinterface** (if specified) attributes. If a matching entry is found, the remaining attributes are used to update that routing table entry. If no match is found, a new entry is created in the workload partition routing table. For more information about the **-I** flag, see the description of the **-i** flag and the **-I** flag of the **mkwpar** command. However, unlike the **mkwpar** command or the **wparexec** command, using the **-I** flag with the **chwpar** command does not change whether WPAR-specific routing is enabled or disabled. Use the **-i** flag (with or without the **-K** flag) to disable or enable WPAR-specific routing.

You can specify the following attributes with the **-I** flag:

rtdest=*destination* (Required) Identifies the host or network to which you are directing the route. You can specify the value using either a symbolic name or a numeric address. You can use the keyword **default** to specify a default route. For more information about the **rtdest** attribute, see the *Destination* parameter of the **route** command.

rtgateway=*gateway* (Required) Identifies the gateway to which packets are addressed. You can specify the value using either a symbolic name or a numeric address.

rtnetmask=*A.B.C.D* Specifies the network mask to the destination address.

rtprefixlen=*n* Specifies the length of a destination prefix, which is the number of bits in the netmask. The value must be a positive integer.

rttype={*net*|*host*} Forces the **rtdest** attribute to be interpreted as the specified type.

rtinterface=*if* Specifies the interface, for example, `en0`, to associate with the route so that packets are sent using the interface when the route is chosen.

-K Deletes the specified attributes from the configuration of the workload partition. You can use the **-K** flag with the following flags:

-A Changes the general **auto** option value of the workload partition to **no**, causing the workload partition not to be started when the **/etc/rc.wpars** command is running. This flag is not valid for application workload partitions.

-c The workload partition is disabled for checkpoint.

-D [**devname**=*device name* | **devid**=*device identifier*]
: Removes an explicit entry concerning an exported device, causing either a device that is not exported previously to be exported, or a previously exported device to be removed. This flag is not valid for application workload partitions.

  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 is the full path to the device (i.e. /dev/pty10). 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 only applies to disk, cdrom, or tape type devices. When the **devid** attribute is used, the **devtype** attribute must also be specified.

-i
: Disables WPAR-specific routing for the workload partition. Any explicit routing table directives that are supplied using the **-l** flag with the **mkwpar**, **wparexec**, or **chwpar** command are maintained (but inactive) in the configuration of the workload partition. The explicit entries are created automatically the next time WPAR-specific routing is enabled.

-l rtdest=*destination* **rtgateway**=*gateway* [*attribute*=*value* ...]
: Removes an explicit entry from the routing table of the workload partition. You must specify at least the **rtdest** attribute and the **rtgateway** attribute to identify the entry to be deleted.

-N address=*A.B.C.D*
: Removes the specified IPv4 address from the configuration of the workload partition.

-N address6=*S*:*T*:*U*:*V*:*W*:*X*:*Y*:*Z*
: Removes the specified IPv6 address from the configuration of the workload partition.

-R [*attribute* ...]
Removes specific fields from the resource control configuration of the workload partition. The **-R** flag can restore each field to its default state. For fields such as **totalProcesses**, the default state is **unlimited**. The following attributes can be restored to the default handling:

- **rset**
- **shares_CPU**
- **CPU**
- **shares_memory**
- **memory**
- **procVirtMem**
- **totalVirtMem**
- **totalProcesses**
- **totalThreads**
- **totalPTYs**
- **totalLargePages**
- **pct_msgIDs**
- **pct_semIDs**
- **pct_shmIDs**
- **pct_pinMem**

When no attributes are specified, the **-K** and **-R** flags restore the resource control profile of the workload partition to its default settings.

-S
Restores the security settings for the workload partition to default values.

-u
Disables the callout to the user script on administration events. (It not delete the script itself.)

-n *newname*
The new name for the workload partition. Do not specify the **-n** flag for a running workload partition.

-N *attribute*=*value* ...
Modifies the network configuration attributes. Entries are matched based on the **address** or **address6** attribute. Each entry must be specified per **-N** flag. You can specify more than one **-N** flags to reconfigure multiple IP addresses. You can modify the following network configuration attributes:

- **interface**=*if* or **interface**=*namemappedif*
- **address**=*A.B.C.D*
- **netmask**=*A.B.C.D*
- **broadcast**=*A.B.C.D*
- **address6**=*S*:*T*:*U*:*V*:*W*:*X*:*Y*:*Z*
- **prefixlen**=*n*

The value of the **prefixlen** attribute ranges from 0 through 128.

The name-mapped interface is in the **/etc/wpars/devmap** file. You can specify the mapping between the name-mapped interface and the system interface as follows:

```
# The comments start with '#'
# Each line contains a pair of name-mapped interface
# and real interface separated by tab or blank spaces.
foo en0
goo en1
soo en2
```

-P
Interactively sets the password for the root user in the workload partition. This flag is not valid for application workload partitions.

-R *attribute*=*value* ... Allows modification of resource control attributes. Most resource controls are similar to those used by Workload Manager. You can specify the following attributes:

active=**yes**|**no** If you specify **yes**, this attribute allows resource control definitions to be retained, but they are rendered inactive. If you specify **no**, performance metrics such as processor and memory usage might not be available through commands such as the **topas** and **wlmstat** commands, both inside and outside of the workload partition.

rset=*rset* Configures the workload partition to use a resource set that is created by the **mkrset** command.

shares_CPU=*n* Specifies the number of processor shares that are available to the workload partition. See Workload Manager shares File.

CPU=*m*%-*SM*%,*HM*% Specifies the percentage processor limits for the processes of the workload partition. See Workload Manager limits File.

shares_memory=*n* Specifies the number of memory shares that are available to the workload partition. See Workload Manager shares File.

memory=*m*%-*SM*%,*HM*% Specifies the percentage memory limits for the processes of the workload partition. See Workload Manager limits File.

procVirtMem=*n*[M|MB|G|GB|T|TB] 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 allowed is 1MB. 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 partition limits File.

totalVirtMem=*n*[M|MB|G|GB|T|TB] Specifies 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 will be terminated. The valid range and units are the same as for procVirtMem. If the value is set to -1 (no units), the limit is disabled. See workload partition limits File.

totalProcesses=*n* Specifies the total number of processes that are allowed in the workload partition. See workload partition limits File.

totalPTYs=*n* Specifies the total number of pseudo terminals that are allowed in the workload partition. See pty Special File.

totalLargePages=*n* Specifies the number of large pages that are allowed for the workload partition. See Large Pages.

pct_msgIDs=*n*% 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.

pct_semIDs=*n*% 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 partition limits File.

-S *attribute*[+|-]=*value*... Modifies the security settings for the workload partition. You can specify only one of the following forms of security changes:

secfile=*secAttrsFile*
Sets the privileges for the workload partition to the privileges listed in the specified file.

privs=*priv,priv,...*
Sets the privileges for the workload partition to the specified list of privileges.

privs+=*priv,priv,...*
Adds the specified list of privileges to the privilege set for the workload partition.

privs-=*priv,priv,...*
Removes the specified list of privileges from the privilege set for the workload partition.

Important: Do not change the security settings when a workload partition is active. For more information about privileges, see the AIX *Security* topic.

-u *userscript* Changes the path to the user script to be run on workload partition administration events. If no user script was configured, the specified script is added to the configuration. An RBAC user cannot run this flag for a WPAR that others own.

-v Enables verbose output.

Parameters

wparname The name of the system or application workload partition to be changed. The *wparname* parameter must be the last parameter on a command line.

Security

Access Control

Only the root user can run the command.

RBAC Disclaimer

Examples

1. To modify the host name of the workload partition called `roy`, enter the following command:
   ```
   chwpar -h roy.com roy
   ```
2. To remove a network address from the workload partition called `dale`, enter the following command:
   ```
   chwpar -K -N address=219.81.45.65 dale
   ```
3. To disable resource controls in the workload partition called `wayne` while retaining the settings for future use, enter the following command:
   ```
   chwpar -R active=no wayne
   ```
4. To unexport a device from a workload partition, enter the following command:
   ```
   chwpar -K -D devname=hdisk1 <wpar name>
   ```
5. To export a device, enter the following command:
   ```
   chwpar -D devname=hdisk1 devtype=disk <wpar name>
   ```
6. To rename the workload partition from `moore` to `hart`, enter the following command:
   ```
   chwpar -n hart moore
   ```

Files

/etc/wpars/devexports Default device export control file for workload partitions.

Related Information

The **clogin**, **lswpar**, **mkwpar**, **rc.wpars**, **rebootwpar**, **rmwpar**, **route**, **startwpar**, **stopwpar**, **syncwpar**, **syncroot**, and **wparexec** commands.

The **devexports** file.

---

chypdom Command

Purpose

Changes the current domain name of the system.

Syntax

/usr/sbin/chypdom [**-I** | **-B** | **-N**] *DomainName*

Description

The **chypdom** command will change the domain name of the system. The *DomainName* parameter specifies the new domain name for the system.

You can use the Network application in Web-based System Manager (wsm) to change network characteristics. You could also use the System Management Interface Tool (SMIT) **smit chypdom** fast path to run this command.

Flags

-I Specifies that the domain name should be changed in the **/etc/rc.nfs** file. With this flag, the domain name will be changed on the next system restart.

-B Specifies that the domain name should be changed now and the **/etc/rc.nfs** file should be updated to reflect the change.

-N Specifies that the domain name should be changed now. No change is made to the **/etc/rc.nfs** file. The **domainname** command is executed to change the domain name of the system.

Examples

To modify the **/etc/rc.nfs** file to set the domain name to `mydomain` on the next system restart, enter:

```
chypdom -I mydomain
```

Files

/etc/rc.nfs Contains the startup script for the NFS and NIS daemons.

Related Information

The **domainname** command, **mkclient** command, **mkmaster** command, **mkslave** command, **smit** command.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

The System Management Interface Tool (SMIT) in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* explains the structure, main menus, and tasks that are done with SMIT.

Network Information Service (NIS) in *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*.

NIS Reference.

ckauth Command

Purpose

Checks the current user session for an authorization.

Syntax

ckauth [**-A**] { *AuthName* [,*AuthName*] ... }

Description

The **ckauth** command determines whether the process that the **ckauth** command is invoked in has the authorizations specified by the *AuthName* parameter. The command is used in shell scripts that need to check for authorizations. With the **ckauth** command, you can specify a single authorization or multiple authorizations through a comma-separated list. The **ckauth** command returns 0 when the calling process has any of the listed authorizations. If you specify the **-A** option, the **ckauth** command returns 0 when the calling process has all of the listed authorizations. A nonzero value is returned for failures.

Flags

-A Checks whether the calling process has all of the listed authorizations.

Examples

1. To determine whether the existing user session has the `aix.fs.manage` authorization, use the following command:

   ```
   $ ckauth aix.fs.manage
   $ echo $?
   0
   ```

2. To determine whether the existing user session has both the `aix.security.user` and `aix.security.group` authorizations, use the following command:

   ```
   $ ckauth -A aix.security.user,aix.security.group
   $ echo $?
   0
   ```

Related Information

The **mkauth**, **lsauth**, **setkst** and **chauth** commands.

ckfilt Command

Purpose

Checks the syntax of filter rules.

Syntax

ckfilt [**-O**] [**-v 4** | **6**]

Description

The **ckfilt** command checks the syntax of the filter rules. IPsec stateful filter rules allow for actions such as IF, ELSE and ENDIF. Thus it is possible to have syntax errors in the rules set, such as IF with out and ENDIF, or an ELSE or ENDIF with out a preceding IF. The **ckfilt** command checks for such errors. Nesting of IF rules is permitted. The **ckfilt** command displays the filter rules, indenting the rules within IF statements in a scoping fashion. If the **-O** flag is used, filter rules and all of their attributes are displayed in a scoped fashion. IPsec filter rules for this command can be configured using the **genfilt** command, IPsec smit (IP version 4 or IP version 6), or Web-based System Manager in the Virtual Private Network submenu.

Flags

-O	Displays filter rule attributes.
-v 4 | **6**	Specifies IPv4 or IPv6.

Exit Status

This command returns the following exit values:

0	The command completed successfully.
non-zero	An error occurred.

Security

This command is only executable by root.

RBAC Disclaimer

Examples

1. To create a set of nested if-else-endif filter rules, use the **genfilt** command as follows:

```
genfilt -v4 -a I -s 192.168.100.101 -d 192.168.100.102 -c tcp -O eq -P 21 -D "IF ftp-cmd being used"
genfilt -v4 -a I -s 192.168.100.101 -d 192.168.100.102 -c tcp -O eq -P 1525 -D "IF 1525 port starts being used"
genfilt -v4 -a D -s 192.168.100.101 -d 192.168.100.102 -c tcp -O eq -P 37 -D "if scope: deny time"
genfilt -v4 -a L -s 192.168.100.101 -d 192.168.100.102 -c tcp -D "ELSE"
genfilt -v4 -a D -s 192.168.100.101 -d 192.168.100.102 -c tcp -O eq -P 13 -D "else scope: deny date"
genfilt -v4 -a E -s 192.168.100.101 -d 192.168.100.102 -c tcp -D "ENDIF"
genfilt -v4 -a L -s 192.168.100.101 -d 192.168.100.102 -c tcp -D "ELSE"
genfilt -v4 -a D -s 192.168.100.101 -d 192.168.100.102 -c tcp -O eq -P 20 -D "else scope: deny ftp-data"
genfilt -v4 -a E -s 192.168.100.101 -d 192.168.100.102 -c tcp -D "ENDIF"
```

The output of the **lsfilt** command will look similar to the following:

```
%lsfilt -v4 -O
1|permit|0.0.0.0|0.0.0.0|0.0.0.0|0.0.0.0|no|udp|eq|4001|eq|4001|both|both|no|all packets|0|all|0|||Default Rule
2|*** Dynamic filter placement rule for IKE tunnels ***|no
3|if|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|21|both|both|no|all packets|0|all|0|||IF ftp-cmd being used
4|if|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|1525|both|both|no|all packets|0|all|0|||IF 1525 port starts being used
5|deny|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|37|both|both|no|all packets|0|all|0|||if scope: de ny time
6|else|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|any|0|both|both|no|all packets|0|all|0|||ELSE
7|deny|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|13|both|both|no|all packets|0|all|0|||else scope: deny date
8|endif|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|any|0|both|both|no|all packets|0|all|0|||ENDIF
9|else|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|any|0|both|both|no|all packets|0|all|0|||ELSE
10|deny|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|20|both|both|no|all packets|0|all|0|||else scope: deny ftp-data
11|endif|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|any|0|both|both|no|all packets|0|all|0|||ENDIF
0|permit|0.0.0.0|0.0.0.0|0.0.0.0|0.0.0.0|yes|all|any|0|any|0|both|both|no|all packets|0|all|0|||Default Rule
```

The output of the **ckfilt** command will look similar to the following:

```
%ckfilt -v4
Beginning of IPv4 filter rules.
Rule 2
IF Rule 3
|    IF Rule 4
|    |    Rule 5
|    ELSE Rule 6
|    |    Rule 7
|    ENDIF Rule 8
```

```
ELSE Rule 9
|    Rule 10
ENDIF Rule 11
Rule 0
```

OR

```
%ckfilt -v4 -O
Beginning of IPv4 filter rules.
2|*** Dynamic filter placement rule for IKE tunnels ***|no
IF 3|if|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|21|both|both|no|all packets|0|all|0|||IF ftp-cmd being used
|    IF 4|if|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|1525|both|both|no|all packets|0|all|0|||IF 1525 port starts being used
|    |    5|deny|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|37|both|both|no|all packets|0|all|0|||if scope: deny time
|    ELSE 6|else|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|any|0|both|both|no|all packets|0|all|0|||ELSE
|    |    7|deny|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|13|both|both|no|all packets|0|all|0|||else scope: deny date
|    ENDIF 8|endif|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|any|0|both|both|no|all packets|0|all|0||| ENDIF
ELSE 9|else|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|any|0|both|both|no|all packets|0|all|0|||ELSE
|    10|deny|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|20|both|both|no|all packets|0|all|0|||else scope: deny ftp-data
ENDIF 11|endif|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|any|0|both|both|no|all packets|0|all|0|||ENDIF
0|all packets|0.0.0.0|0.0.0.0|0.0.0.0|0.0.0.0|no|0|???|0|???|0|?????|????????|no|???????|0||0|||
```

2. If incorrect if-else-endif rules are created, the **ckfilt** command will find and report the error as follows:

```
%lsfilt -v4 -O
1|permit|0.0.0.0|0.0.0.0|0.0.0.0|0.0.0.0|no|udp|eq|4001|eq|4001|both|both|no|all packets|0|all|0|||Default Rule
2|*** Dynamic filter placement rule for IKE tunnels ***|no
3|if|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|21|both|both|no|all packets|0|all|0|||IF ftp-cmd being used
4|deny|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|37|both|both|no|all packets|0|all|0|||if scope: deny time
5|else|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|any|0|both|both|no|all packets|0|all|0|||ELSE
6|deny|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|13|both|both|no|all packets|0|all|0|||else scope: deny date
7|endif|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|any|0|both|both|no|all packets|0|all|0|||ENDIF
8|else|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|any|0|both|both|no|all packets|0|all|0|||ELSE
9|deny|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|20|both|both|no|all packets|0|all|0|||else scope: deny ftp-data
10|endif|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|any|0|both|both|no|all packets|0|all|0|||ENDIF
0|permit|0.0.0.0|0.0.0.0|0.0.0.0|0.0.0.0|yes|all|any|0|any|0|both|both|no|all packets|0|all|0|||Default Rule

%ckfilt -v4
Beginning of IPv4 filter rules.
Rule 2
IF Rule 3
|    Rule 4
ELSE Rule 5
|    Rule 6
ENDIF Rule 7
No preceeding IF statement for filter rule 8.
The filter rules failed the syntax check.

%ckfilt -v4 -O
Beginning of IPv4 filter rules.
2|*** Dynamic filter placement rule for IKE tunnels ***|no
IF 3|if|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|21|both|both|no|all packets|0|all|0|||IF ftp-cmd being used
|    4|deny|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|37|both|both|no|all packets|0|all|0|||if scope: deny time
ELSE 5|else|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|any|0|both|both|no|all packets|0|all|0|||ELSE
|    6|deny|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|eq|13|both|both|no|all packets|0|all|0|||else scope: deny date
ENDIF 7|endif|192.168.100.101|255.255.255.255|192.168.100.102|255.255.255.255|yes|tcp|any|0|any|0|both|both|no|all packets|0|all|0|||ENDIF

No preceeding IF statement for filter rule 8.
The filter rules failed the syntax check.
```

Location

/usr/sbin/ckfilt

Files

/etc/security/ipsec_filter	This command reads the **/etc/security/ipsec_filter** ODM database. Rules are inserted and changed in this database using the **genfilt** and **chfilt** commands.

Related Information

"chfilt Command" on page 375.

The **genfilt** command in *AIX 5L Version 5.3 Commands Reference, Volume 2*.

The **lsfilt** and **mkfilt** commands in *AIX 5L Version 5.3 Commands Reference, Volume 3*.

The *Security*.

ckpacct Command

Purpose
Checks data file size for process accounting.

Syntax
/usr/sbin/acct/ckpacct [*BlockSize*]

Description
The **ckpacct** command checks the size of the active data file, **/var/adm/pacct**. Normally, the **cron** daemon runs this command. If the size of the active data file exceeds the number of blocks specified by the *BlockSize* parameter, the **ckpacct** command invokes the **turnacct switch** command to turn off process accounting. The default value for the *BlockSize* parameter is 1000.

If the number of free disk blocks in the **/var** file system falls below 500, the **ckpacct** command automatically turns off process accounting by invoking the **turnacct off** command. When 500 blocks are again available, accounting is reactivated. This feature is sensitive to how frequently the **ckpacct** command is run.

When the **MAILCOM** environment variable is set to **mail root adm**, a mail message is sent both to the **root** and **adm** groups if an error occurs.

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

Examples
To automatically check the size of the **/var/adm/pacct** data file, add the following to the **/var/spool/cron/crontabs/root** file:

```
5 * * * * /usr/sbin/acct/ckpacct
```

This example shows the instructions the **cron** daemon reads and acts upon. The **ckpacct** command runs at 5 minutes past every hour (`5 *`) every day. This command is only one of the accounting instructions normally given to the **cron** daemon. See ″Setting Up an Accounting System″ in *Operating system and device management* for more information on typical **cron** accounting entries.

Files

/usr/sbin/acct	The path to the accounting commands
/var/adm/pacct	Current file for process accounting.

Related Information
The **acctcom** command, **acctprc1, acctprc2,** or **accton** command, **turnacct** command.

The **cron** daemon.

The **acct** subroutine.

For more information about the accounting system, the preparation of daily and monthly reports, and the accounting files, see the System accounting in *Operating system and device management*.

Setting up an accounting subsystem in *Operating system and device management* explains the steps you must take to establish an accounting system.

ckprereq Command

Purpose

Verifies that all prerequisite software is available and at the appropriate revision levels.

Syntax

ckprereq [**-v**] [**-O** { **r** | **u** | **s** }] [**-f** *PrereqFile* | **-l** *FilesetName* [*Level*]]

Description

The **ckprereq** command determines whether the system level is compatible with the software product to be installed or updated.

The **ckprereq** command is designed to be used during the installation procedures of a software product.

When **ckprereq** is invoked with the **-f** flag, the *PrereqFile* parameter specifies a software prerequisite list file. Each record in this file contains information about a prerequisite fileset needed to complete the installation procedure.

When **ckprereq** is invoked with the **-l** flag, the prerequisite information is read from the *ProductName* information in the Software Vital Product Data (SWVPD) database.

If the *PrereqFile* parameter was given with the **-f** flag, then an output file is produced by the **ckprereq** command. The output file overwrites the input file and is a listing of the original input. Any failing lines are marked with a failure code in the first column. The **ckprereq** command ignores the failure codes if an output from a previous **ckprereq** call is used as input.

There are four possible requisite tests: **prereq**, **coreq**, **ifreq**, and **instreq**.

A **prereq** is a test to check that a fileset is installed and at a specified revision level. To be considered installed, the SWVPD entry for the software product must be in the APPLIED, APPLYING, COMMITTED, or COMMITTING state. A **prereq** requires that the fileset also be at the specified revision level before installing the independent fileset.

A **coreq** test is similar to a **prereq**, except that **coreq** tests can be installed in any order, but **prereq** tests require a specific order. If a corequisite software product is not yet installed, the test is ignored and the failure codes are not set because it is assumed that the software product will be installed. The **coreq** test is ignored by the **ckprereq** command. (It is not ignored by the installp command's requisite checking procedures.)

An **ifreq** test is identical to a **coreq**, except that it tests for the revision level only if the fileset is installed. If the fileset is not installed, the **ifreq** test is ignored.

An **instreq** test is treated like a **prereq** test by the **ckprereq** command. The special meaning of instreq is only used by the up-front requisite checks of the **installp** command.

The **installp** command checks corequisite and if-requisite file sets at the completion of an install set, and returns messages for any unsatisfied **coreq** or **ifreq** conditions. An if-requisite condition would be unsatisfied if the if-requisite product is installed, but does not match the revision level specified.

Flags

-f *PrereqFile*	Specifies the file name of a prerequisite list file.
-l *FilesetName*[*Level*]	Specifies the name of the fileset or fileset update under which to look for the prerequisite information from the SWVPD database.
-O {**r**|**u**|**s**}	Specifies the part of the file tree of the software product that is to be checked. If this flag is not specified, the **ckprereq** command uses the value of the **INUTREE** environment variable to determine which part to check. The **INUTREE** environment variable is set by the **installp** command. The **r** option indicates the / (root) part of the software product is checked. The **u** option indicates the **/usr** part of the software product is checked. The **s** option indicates the **/usr/share** part of the software product is checked. Only one part can be checked at a time.
-v	Displays a descriptive message to standard error for each failure in the prerequisite list file.

Return Values

The **ckprereq** command tests the current version, release, modification level, fix level, and fix ID found in the SWVPD and marks the first column in each failing line in the output file with one of the following codes if the test was unsuccessful:

f	The test for the fix (level) was unsuccessful.
m	The test for the modification level was unsuccessful.
n	The fileset is not installed or is set to **broken**.
p	The test for the fix ID was unsuccessful.
r	The test for the release was unsuccessful.
s	There is a syntax error in the *PrereqFile* parameter.
v	The test for the version was unsuccessful.

If a serious error occurs, such as an invalid command line or a syntax error in the prerequisite list file, the return code for the **ckprereq** command is 255. Otherwise, the return code is a number that represents the number of tests that failed.

Security

Access Control

You must have root authority to run this command.

RBAC Disclaimer

Examples

1. To check that the requisite specifications in the file **/tmp/prq.test**, that has the following contents:

   ```
   *prereq bos.rte 4.1.0.0
   *prereq X11.base.rte 4.1.0.0
   ```

   are satisfied, while reporting any failures, enter:

   ```
   ckprereq -vf /tmp/prq.test
   ```

2. To check all the requisite software listed in the **/usr/lpp/snaserv/prereq2** file for the root part, enter:

   ```
   ckprereq -f /usr/lpp/snaserv/prereq2 -Or
   ```

3. To check that the requisites of the installed fileset update bos.net.tcp.client at level 4.1.0.1 are met, enter:

```
ckprereq -l bos.net.tcp.client 4.1.0.1
```

Files

/etc/objrepos/product	Database containing information about the software installed in the **/root** part of the file system.
/**usr/lib/objrepos/product**	Database containing information about the software installed in the **/usr** part of the file system.
/usr/share/lib/objrepos/product	Database containing information about the software installed in the **/usr/share** part of the file system.

Related Information

The **installp** command.

---

cksum Command

Purpose

Displays the checksum and byte count of a file.

Syntax

cksum [*File* ...]

Description

The **cksum** command reads the files specified by the *File* parameter and calculates a 32-bit checksum Cyclic Redundancy Check (CRC) and the byte count for each file. If no files are specified, the **cksum** command reads standard input. The checksum, number of bytes, and file name are written to standard output. If standard input is used, the path name and leading space are omitted.

The **cksum** command can be used to compare a suspect file copied or communicated over noisy transmission lines against an exact copy of a trusted file. The comparison made by the **cksum** command may not be cryptographically secure. However, it is unlikely that an accidentally damaged file will produce the same checksum as the original file.

The **cksum** command uses a different algorithm to calculate the 32-bit checksum CRC than the **sum** command. The **cksum** command uses a CRC algorithm based on the Ethernet standard frame check. For more information on the Ethernet standard, see ″Understanding DLCETHER Protocol Support″ in *AIX 5L Version 5.3 Communications Programming Concepts*.

> **Note:** The **cksum** command is POSIX 1003.2 compliant and the checksum produced is guaranteed to be calculated the same on all POSIX 1003.2 compliant systems.

The following generating polynomial defines CRC checksum encoding:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

The following procedure mathematically defines the CRC value corresponding to a given file:

1. The *n* bits to be evaluated are considered to be the coefficients of a mod 2 polynomial M(x) of degree *n*-1. These *n* bits are the bits from the file. The most significant bit is the most significant bit of the first octet of the file. The last bit is the least significant bit of the last octet, padded with zero bits (if necessary) to achieve an integral number of octets, followed by one or more octets representing the length of the file as a binary value, least significant octet first. The smallest number of octets capable of representing this integer is used.

2. M(x) is multiplied by x32 (that is, shifted left 32 bits) and divided by G(x) using mod 2 division, producing a remainder R(x) of degree 31.
3. The coefficients of R(x) are considered to be a 32-bit sequence.
4. The bit sequence is complemented, and the result is the CRC.

Exit Status

This command returns the following exit values:

0 All files were processed successfully.
>0 An error occurred.

Examples

To display the checksum and the size, in bytes, of `file1` and `file2`, enter:

```
cksum file1 file2
```

If the checksum of the `file1` file is `3995432187` and contains `1390` bytes, and the checksum of the `file2` file is `3266927833` and contains `20912` bytes, the **cksum** command displays:

```
3995432187       1390    file1
3266927833       20912   file2
```

Files

/usr/bin/cksum Contains the **cksum** command.

Related Information

The **sum** command, **wc** command.

File systems in *Operating system and device management* gives an explanation of what a file system is and why to use one.

Understanding DLCETHER Protocol Support in *AIX 5L Version 5.3 Communications Programming Concepts* provides information on the Ethernet standard.

---

clear Command

Purpose

Clears the terminal screen.

Syntax

clear

Description

The **clear** command clears your screen, if possible. The **clear** command first checks the **TERM** environment variable for the terminal type. Next, the **/usr/share/lib/terminfo** directory, which contains terminal definition files, is checked to determine how to clear the screen. If the **TERM** environment variable is not set, the **clear** command exits without taking any action.

Examples

To clear your screen, enter:

```
clear
```

Files

/usr/share/lib/terminfo	Contains terminal information database.

Related Information

The **tput** command.

The Input and output redirection overview in *Operating system and device management* describes how the operating system processes input and output and how to use redirect and pipe symbols.

---

clogin Command

Purpose

Initiates a user session or runs a command within a workload partition.

Syntax

clogin *WparName* [**-l** *User*] [*command* [*args*]]

Description

The **clogin** command provides a mechanism for the root user to log in or run a command within a workload partition.

When you specify the **-l** flag, a session is initiated as if the session was started by the user specified using the *User* parameter in the workload partition. If a subsequent command is specified, the command runs as if it was launched as a parameter to the login shell associated with the *User*. The **clogin** command performs similar operations as the **su** command, so all of the functions that are associated with the **su** command apply to the **clogin** command.

Note: The pseudo-terminal on which the session is initiated belongs to the global environment, but the login shell running in the terminal belongs to the workload partition.

Flags

WparName	The name of the workload partition in which to log in.
-l *User*	Specifies the user name to log in the workload partition. Default is root. If you specify the *command* parameter, you must specify both the **-l** flag and the *User* parameter.
command	Specifies the command running within the workload partition. The command runs as a parameter to the login shell that is associated with the user.
args	Specifies optional parameters to use when the command that is specified by the *command* parameter runs.

Security

Access Control	Only the root user can run the command.

Examples

1. To log in to a workload partition named **bucko** as user **dan**, enter the following command:

   ```
   clogin bucko -l dan
   ```

2. To run the **/usr/bin/ps** command with the **-T 1** option as user root in a workload partition named **howdy**, enter the following command:

   ```
   clogin howdy -l root /usr/bin/ps -T 1
   ```

Related Information

The “chwpar Command” on page 529, **lswpar** command, **mkwpar** command, **rc.wpars** command, **rebootwpar** command, **rmwpar** command, **startwpar** command, **stopwpar** command, **syncwpar** command, **syncroot** command, and **wparexec** command.

The **devexports** file.

---

clsnmp Command

Purpose

TheAIX **clsnmp** command provides the SNMP manager function from the AIX shell to query SNMP agents for network management information.

Syntax

clsnmp [**-d** *DebugLevel*] [**-h** *TargetHost*] [**-c** *Community*] [**-t** *TimeOutValue*] [**-r** *RetryNumber*] [**-n** *NonRepeaters*] [**-m** *MaxRepetitions*] [**-p** *PortNumber*] [**-v**] [**-f** *ConfigurationFile*] [**-?**] *Function* [*MIBVariable* [*VariableType*] [*Value*] [...]]

Description

Use the **clsnmp** command to issue SNMP requests to agents and to process SNMP responses returned by agents. The AIX **clsnmp** commands supports issuance of SNMPv1, SNMPv2c, and SNMPv3 requests.

SNMP request types

findname
: Sends a request that a search be done to obtain the textual name, for a given *MIBVariable* input, whose internal ASN.1 value best matches the input ASN.1 value. The search first checks the **/etc/mib.defs** file, and if a matching textual name is not found, continue with the compiled MIB. Only one *MIBVariable* is allowed per **clsnmp findname** invocation.

get
: Sends a request to an SNMP agent for a specific management information base (MIB) variable. **clsnmp** then waits for a response or times out.

getbulk
: Obtains the value of the variables in the MIB tree specified by the OID or MIB variable name. A single **getbulk** performs the same function as a series of **getnext**s with fewer data exchanges between the **clsnmp** command and the SNMP agent.

getnext
: Sends a request to an SNMP agent for the next MIB variable that lexicographically follows the *MIBVariable* specified. **clsnmp** then waits for a response or times out.

set
: Sends a request to an SNMP agent to set a specific MIB variable. **clsnmp** then waits for a response or times out.

trap
: Listens for SNMP traps and displays **trap** information when they occur. Uses the default, well-known port 162 or the port number specified on the -p option. The **clsnmp trap** function continues to listen for traps until the process is killed or canceled.

walk Issues a **getnext** request for a specified prefix, then continues to issue **getnext** requests for as long as there are variables that match the specified prefix. A prefix can be any leading portion of the complete object identifier.

Usage

The **set** operation is not supported on all MIB objects. The **set** operation may be rejected if the agent or subagents managing the MIB object does not support SET.

getbulk is an SNMPv2 function. If the target agent only supports SNMPv1, the target agent ignores your request. As a result, your request times out.

The function keywords are not case sensitive. The flags, variable names and values are case sensitive.

In order to listen to traps from NetView SNMP and AIX **clsnmp** at the same time, use the **-p** *PortNumber* parameter on the **clsnmp** command. Only one management application at an IP address can listen on a port at a time. Specifying **-p** on the **clsnmp** trap command enables a port other than well-known port 162 to be used. Both ports must be configured as agent trap destinations.

An **clsnmp** command that is not authenticated (by using an acceptable community name or user name) will time out.

The **clsnmp** command uses two configuration files: **/etc/mib.defs** and **clsnmp.conf**. Sample files are shipped in the **/usr/samples/snmpdv3** directory.

The clsnmp command supports sending SNMPv1, SNMPv2c, and SNMPv3 requests. The file clsnmp uses to determine whether it should send an SNMPv1,SNMPv2c or SNMPv3 request is the **clsnmp.conf** file. If the target specified by way of the -h parameter matches a winSNMP name in the **clsnmp.conf** file, **clsnmp** sends the request using the parameters specified on the entry. If the **-h** parameter is not specified, then the request will be sent as an SNMPv1request.

Flags

-c *Community*
Specifies the community name used to access the specified variables at the destination SNMP agent. If you do not specify a community name, the default name is public. Community names are not required when using the user-based security model.
Note: Community names are case sensitive.

-d *DebugLevel*
Specifies the debug level. The default level is 0, which means no debug. The higher the debug level, the greater the number of messages that are displayed. The debug levels are 0-4.

-f *ConfigurationFile*
Specifies the full path and file name of the configuration file.

-h *TargetHost*
Specifies the target host to which you want to send a request. The host can be an IPv4 address, an IPv6 address, a host name, or a winSNMP name in the **clsnmp.conf** configuration file. If you do not specify a host, the default is your local host.

-m *MaxRepetitions*
Only applies to **getbulk**. This is ignored if the function request is not a **getbulk**. Maximum repetitions is the number of lexicographic successors to be returned for each variable binding pair after the first ″**-n** number″ successors. For example, starting with successor ″**-n** number″+1, return ″**-m** number″ of successors for each variable binding pair. The default is 10.

-n *NonRepeaters*
Only applies to **getbulk** requests. This is ignored if the function request is not a **getbulk**. *NonRepeaters* is the number of variable binding pairs (name/value), starting with the first, for which only a single successor is returned. The default is 0.

-p *PortNumber*
Specifies the number of the port that listens for traps. If a port number is not specified, the **clsnmp trap** function listens on the well-known port 162, the default port for **clsnmp** traps.

-r *RetryNumber*
Specifies the maximum number of times to retry the command if it timed out. The default is 2.

-t *TimeOutValue*
Specifies the amount of time (in seconds) that the **clsnmp** command waits for a reply from the SNMP agent. The default is 3.

-v
Specifies that the output from a request should be displayed using verbose output, for example, using the textual name instead of the MIB object identifier.

-?
Displays help information.

Parameters

Function
Specifies the SNMP function/operation to perform, which is one of the following: **get**, **getnext**, **getbulk**, **set**, **walk**, **trap**, **findname**.

MIBVariable
Specifies the Management Information Base (MIB) object, using its object descriptor (textual name), object identifier in ASN.1 notation, or a combination of the two. When used with **walk**, this is the MIB object prefix. A prefix can be any leading portion of the complete object identifier. When used with **findname**, this is the object identifier in ASN.1 notation.

Value
Specifies the value to be set by the SET function. If white space is needed in the value, you must enclose the value in double quotes (″). If you want to set a variable to a value that is also a type, you must specify the type.

VariableType

Specifies the type of value being set. To complete an SNMP SET request, the SMI_type must be known. If no type is specified, **clsnmp** searches first the **/etc/mib.defs** file and then the compiled MIB to determine the type. If the variable is not found, an error is returned. If a VariableType is specified, the VariableType takes precedence over any type that may be assigned in the MIB. The VariableType and value must be compatible. For example, if you specify a type of ″number″ and a value of ″foo,″ an error is returned because ″foo″ is not a number. *VariableType* is not case sensitive. Valid variable types are:

- bitstring
- counter
- counter32
- counter64
- display or displaystring
- gauge
- gauge32
- integer
- integer32
- ipaddress
- nsapaddress
- null
- objectidentifier or OID
- octetstring
- opaque
- opaqueascii
- timeticks
- uinteger

Limitation

Starting AIX 6.1 with the 6100-01 Technology Level and later, the **clsnmp** manager configured with **SNMPv1** type requests does not return all the supported clsnmp objects. The **snmpdv3** daemon stops dumping the MIB variables when the **snmpdv3** daemon encounters an MIB with a SMI-v2 data type while processing an **SNMPv1** type request from the **clsnmp** manager.

Work around

The **clsnmp** manager should be configured with **SNMPv2** type requests or **SNMPv3** type requests to dump all of the MIB variables with the **snmpdv3** daemon.

Examples

1. Getting the MIB variable.
   a. The following requests MIB object `sysName.0`:

      ```
      clsnmp get sysName.0
      ```

      The output from this command looks similar to:

      ```
      1.3.6.1.2.1.1.5.0 = hostname.austin.ibm.com
      ```

   b. The following requests MIB object `myName.0`, where `myName` is defined in the **/etc/mib.defs** file to be the same object identified by `sysName.0`:

```
clsnmp get myName.0
```

The output from this commands looks similar to:

```
1.3.6.1.2.1.1.5.0 = myhostname.austin.ibm.com
```

c. The following requests MIB object sysName.0 through an IPv6 address:

```
clsnmp -h 2000:1:1:1:209:6bff:feae:6d67 get sysName.0
```

The output from this command looks similar to:

```
1.3.6.1.2.1.1.5.0 = hostname.austin.ibm.com
```

2. Getting the next MIB variable.

a. The following requests the next logical MIB object:

```
clsnmp getnext udp
```

The output from this command looks similar to:

```
1.3.6.1.2.1.7.1.0 = 653
```

b. The following requests the next logical object, using the **-v** option to have value displayed with textual name instead of object identifier:

```
clsnmp -v getnext udp
```

The output from this command looks similar to:

```
udpInDatagrams.0 = 653
```

3. Setting the MIB variable.

a. The following sets MIB object sysName.0 to a value of 'hostname.austin.ibm.com':

```
clsnmp set sysName.0  "hostname.austin.ibm.com"
```

This command produces output similar to:

```
1.3.6.1.2.1.1.5.0 = hostname.austin.ibm.com
```

b. The value of MIB object sysName.0 can also be set using the *VariableType* parameter to specify the type of value being set, as in the following example:

```
clsnmp set sysName.0 displayname "hostname.austin.ibm.com"
```

This command produces output similar to:

```
1.3.6.1.2.1.1.5.0 = hostname.austin.ibm.com
```

4. Walking the MIB tree.

The following returns by name all objects beginning with the same object identifier prefix, but with fewer data packages to be exchanged between the clsnmp command and the SNMP agent:

```
clsnmp -h loopback -v -m 10 bulkwalk udp
```

The output of this command looks similar to the following:

```
clsnmp -v walk udp
udpInDatagrams.0 = 653
udpNoPorts.0 = 22
udpInErrors.0 = 0
udpOutDatagrams.0 = 678
udpLocalAddress.0.0.0.0.7 = 0.0.0.0
udpLocalAddress.0.0.0.0.9 = 0.0.0.0
udpLocalAddress.0.0.0.0.13 = 0.0.0.0
udpLocalAddress.0.0.0.0.19 = 0.0.0.0
udpLocalAddress.0.0.0.0.37 = 0.0.0.0
udpLocalAddress.0.0.0.0.161 = 0.0.0.0
udpLocalAddress.0.0.0.0.5020 = 0.0.0.0
udpLocalPort.0.0.0.0.7 = 7
udpLocalPort.0.0.0.0.9 = 9
udpLocalPort.0.0.0.0.13 = 13
```

```
udpLocalPort.0.0.0.0.19 = 19
udpLocalPort.0.0.0.0.37 = 37
udpLocalPort.0.0.0.0.161 = 161
udpLocalPort.0.0.0.0.5020 = 5020
```

5. Getting multiple MIB variables.

   The following requests multiple MIB objects using the **getbulk** request type. The **getbulk** request type returns the next logical object for one or more MIB objects listed on the command. In the following example, the **-n** option indicates that only one next logical object is requested for the first two variables (sysLocation and ifTable). For all the other objects in the list (tcp, udp, and icmp), the **-m** option indicates that 5 repetitions are requested.

   Note: The **getbulk** request type is an SNMPv2 function. The **-h** parameter identifies a host, loopback, defined in the **clsnmp.conf** file as an agent that supports SNMPv2 or SNMPv3.

   ```
   clsnmp -h loopback -v -n 2 -m 5 getbulk sysLocation ifTable tcp udp icmp
   ```

   This command produces output similar to the following:

   ```
   sysLocation.0 = Research Triangle Park, NC
   ifIndex.1 = 1
   tcpRtoAlgorithm.0 = 4
   udpInDatagrams.0 = 782
   icmpInMsgs.0 = 22
   tcpRtoMin.0 = 0
   udpNoPorts.0 = 22
   icmpInErrors.0 = 0
   tcpRtoMax.0 = 120
   udpInErrors.0 = 0
   icmpInDestUnreachs.0 = 22
   tcpMaxConn.0 = -1
   udpOutDatagrams.0 = 807
   icmpInTimeExcds.0 = 0
   tcpActiveOpens.0 = 1
   udpLocalAddress.0.0.0.0.7 = 0.0.0.0
   icmpInParmProbs.0 = 0
   ```

6. Finding the name of an ASN.1 variable.

   The following sends a request that a search be done to obtain the textual name, for a given *MIBVariable* input, whose internal ASN.1 value best matches the input ASN.1 value. The search begins with the **/etc/mib.defs** file and, if not found, continues with the compiled MIB. Only one *MIBVariable* is allowed per **clsnmp findname** invocation. For example, this can be done with a command similar to the following:

   ```
   clsnmp findname 1.3.6.1.2.1.6.13.1.2
   ```

   This command produces output similar to the following:

   ```
   1.3.6.1.2.1.6.13.1.2 found as: tcpConnLocalAddress
   ```

   A similar example is:

   ```
   clsnmp findname 1.3.6.1.2.1.6.13.1.2.0
   ```

   This command produces output similar to the following:

   ```
   1.3.6.1.2.1.6.13.1.2.0 found as: tcpConnLocalAddress.0
   ```

   Another similar example is:

   ```
   clsnmp findname 1.3.6.1.2.
   ```

   This command produces output similar to the following:

   ```
   1.3.6.1.2. found as: mgmt
   ```

7. Issuing an SNMPv3 request.

   a. If an winSnmpName entry is configured in **/etc/clsnmp.conf** file on the manager host with an entry like the following (all on one line):

```
target1   9.3.149.26 snmpv3 u1 -          - AuthNoPriv   HMAC-SHA
76784e5935acd6033a855df1fac42acb187aa867 -   -
```

and on the snmpd agent machine `9.3.149.26`, user `u1` is properly configured, then we can issue command on the manager host:

```
clsnmp -v -h target1 get sysName.0
```

This command will produce output similar to:

```
sysName.0 = somehostname.austin.ibm.com
```

b. It is simple to issue a trap command, as follows:

```
clsnmp trap
```

Note: If the security model of the trap received is SNMPv3, make sure on the manage station where is listens to the trap has the /**etc/clsnmp.conf** file properly configured in order to receive the trap.

Files

/etc/clsnmp.conf	Configuration file for the **clsnmp** command.
/etc/mib.defs	Defines the Management Information Base (MIB) variables the SNMP agent and manager should recognize and handle.

Related Information

The **pwchange** command, **pwtokey** command, **snmpdv3** daemon.

The **/etc/clsnmp.conf** file, **/etc/snmpdv3.conf** file.

---

cmp Command

Purpose

Compares the contents of two files and reports the first character that differs.

Syntax

cmp [**-l** | **-s**] *File1 File2*

Description

The **cmp** command compares files designated by the *File1* and *File2* parameters and writes the results to standard output. If you specify a **-** (minus sign) for either the *File1* or *File2* parameter, the **cmp** command reads standard input for that file. Only one file can be read from standard input. Under default conditions, the **cmp** command displays nothing if the files are the same. If they differ, the **cmp** command displays the byte and line number at which the first difference occurs. If the **-l** flag is specified and if one file is an initial subsequence of the other (that is, if the **cmp** command reads an end-of-file character in one file before finding any differences), the **cmp** command notes this. Normally, use the **cmp** command to compare non-text files and the **diff** command to compare text files.

Flags

-l (Lowercase L) Displays, for each difference, the byte number in decimal and the differing bytes in octal.

-s Returns only an exit value. A value of 0 indicates identical files; value of 1 indicates different files; a value of 2 indicates inaccessible file or a missing option.

Exit Status

This command returns the following exit values:

0 The files are identical.
1 The files are different. This value is given even if one file is an initial subsequence of the other (one file is identical to the first part of the other).
>1 An error occurred.

Examples

1. To determine whether two files are identical, enter:

   ```
   cmp prog.o.bak prog.o
   ```

   This compares `prog.o.bak` and `prog.o`. If the files are identical, then a message is not displayed. If the files differ, then the location of the first difference is displayed; for example:

   ```
   prog.o.bak prog.o differ: char 4, line 1
   ```

   If the message `cmp: EOF on prog.o.bak` is displayed, then the first part of `prog.o` is identical to `prog.o.bak`, but there is additional data in `prog.o`.
2. To display each pair of bytes that differ, enter:

   ```
   cmp  -l prog.o.bak prog.o
   ```

   This compares the files, and then displays the byte number (in decimal) and the differing bytes (in octal) for each difference. For example, if the fifth byte is octal 101 in `prog.o.bak` and 141 in `prog.o`, the **cmp** command displays:

   ```
   5 101 141
   ```
3. To compare two files without writing any messages, enter:

   ```
   cmp  -s prog.c.bak prog.c
   ```

   This gives an exit value of `0` if the files are identical, a value of `1` if different, or a value of `2` if an error occurs. This form of the command is normally used in shell procedures. For example:

   ```
   if cmp  -s prog.c.bak prog.c
   then
    echo No change
   fi
   ```

   This partial shell procedure displays `No change` if the two files are identical.

Files

/usr/bin/cmp Contains the **cmp** command.

Related Information

The **comm** command, **diff** command, **ksh** command.

Files in *Operating system and device management*.

Input and output redirection overview. in *Operating system and device management*.

col Command

Purpose

Filters for standard output text having reverse line feeds and forward/reverse half-line-feeds.

Syntax

col [**-b**] [**-f**] [**-p**] [**-x**] [**-T** *Name*] [**-l** *Number*]

Description

The **col** command reads a text file from standard input and writes to standard output. It performs the line overlays implied by the **flr** commands (reverse line feeds), as well as by the **hlf** and **hlr** commands (forward and reverse half-line-feed, respectively). The **nterm** file format document gives a description of these line-feed commands. Use the **col** command for filtering multicolumn output produced by the **nroff** command, the **.rt** request, and by output from the **tbl** command.

Use the **col** command as an **nroff** backend filter for devices that cannot handle reverse line motions (such as most impact printers). To print correctly, use the **col** command to process outputs from the **tbl** command, the **neqn** command, or explicit reverse motion request files (such as the **.sp -10V** file), or files with 2-column output. Do not process the **nroff** output targeted for the following devices with the **col** command:

- **hplj**
- **ibm4019**
- **ibm5577**
- **ibm5575**

Unless the **-x** flag is given, whenever possible, the **col** command converts white spaces to tabs upon output wherever possible to shorten printing time.

The **col** command, used with the **-T37** file, assumes the ASCII control characters, SO (\017) and SI (\016), begin and end text in an alternate character set. The **col** command remembers the character set each input character belongs to and upon output, generates SI and SO characters as appropriate to ensure that each character is printed in the correct character set.

Upon input, the **col** command accepts only the control characters for the Space, Backspace, Tab, and Return keys; the new-line character; the SI, SO (with the **-T37** file), and VT control characters; and the reverse line feed, forward half-line-feed and reverse half-line-feed characters. The VT control character (\013) is an alternate form of full reverse line feed, included for compatibility with some earlier programs of this type. The **col** command ignores all other nonprinting characters.

Normally, the **col** command ignores any escape sequences that are unknown to it and found in the input. However, the **-p** option can be used to cause the **col** command to output these sequences as regular characters, subject to overprinting from reverse line motions. The use of this option is highly discouraged unless the user is fully aware of the textual position of the escape sequences.

Notes:

1. If the output is being sent to a device that can interpret half-line motions, enter:
   ```
   nroff -Tppds File... | col -f -Tppds
   ```
   Otherwise, for example, enter:
   ```
   nroff -Tlp File... | col -Tlp
   ```
2. The maximum number of lines that can be backed up is 128.
3. No more than 800 characters, including backspaces, are allowed on a line.

4. Local vertical motions that would result in backing up over the first line are ignored. As a result, the first line must not contain any superscripts.

Flags

-b	Assumes that the output device in use is not capable of backspacing. In this case, if two or more characters are to be displayed in the same position, only the last one that is read is displayed in the output.
-f	Suppresses the default treatment of half-line motions in the input. Normally, the **col** command does not emit half-line motions on output, although it does accept them in its input. With this flag, output can contain forward half-line-feeds (**hlf**) but not reverse line feeds (**flr** or **hlr**).
-p	Displays unknown escape sequences as characters, subject to overprinting from reverse line motions. Normally, the **col** command ignores them.
-x	Converts tabs to white space.
-T*Name*	Uses the workstation specification indicated by the *Name* variable. *Name* varables for ″Terminal Names for Typewriter-like Devices and Line Printers″ are discussed in the **nroff** command **-T***Name* flag. The default is **37**.
-l *Number*	(lowercase L) Sends the specified number lines of text in memory to a buffer during processing.

Exit Status

The following exit values are returned:

0	Indicates successful completion.
>0	Indicates an error occurred.

Related Information

The **hplj** command, **mm** command, **nroff** command, **ps4014** command, **tbl** command.

The **nterm** file format.

---

colcrt Command

Purpose

Filters **nroff** command output for cathode ray tube (CRT) previewing.

Syntax

colcrt [**-**] [**-2**] [*File* ...]

Description

The **colcrt** command filters output from the **nroff** command so that the output can be previewed on a CRT. The **colcrt** command provides virtual half-line-feed and reverse line-feed sequences for terminals without these capabilities. The **colcrt** command changes underline characters to dashes and places these characters and the half-line characters on new lines between the normal output lines.

Notes:

1. Use this command with the **37** viewing device
2. The **-** (minus sign) flag removes all underlining; therefore, a true underline character does not show with the **-** (minus sign) flag.
3. It is not possible to back up more than 102 lines.
4. General overstriking is lost. As a special case, the | (vertical bar) overstruck with the - (dash) or the _ (underline) characters becomes the + (plus sign).

5. Lines are truncated to up to 132 characters.

Parameters

File Specifies a file processed by the **nroff** command for viewing on a CRT.

Flags

- Suppresses underlining. This flag is useful for previewing boxed tables from the **tbl** command.

-2 Causes all half-lines to be printed, effectively double-spacing the output. This is useful when printing output with subscripts and superscripts on a line printer, where half-lines normally are not displayed.

Examples

A typical use of the **colcrt** command is:

```
tbl exum2.n | nroff -ms -T37 | colcrt - | pg
```

Related Information

The **col** command, **nroff** command, **pg** command, **tbl** command, **troff** command, **ul** command.

---

colrm Command

Purpose

Extracts columns from a file.

Syntax

colrm *First* [*Last*]

Description

The **colrm** command removes selected columns from a file. Input is taken from standard input. Output is sent to standard output.

If called with one parameter, the columns of each line are removed starting with the specified column. If called with two parameters, the columns from the first column to the last column are removed.

Column numbering starts with column 1.

Examples

To remove columns from the `text.fil` file, enter:

```
colrm 6 < text.fil
```

If `text.fil` contains:

```
123456789
```

then the **colrm** command displays:

```
12345
```

Files

/usr/bin/colrm Contains the **colrm** command.

Related Information

The **cut** command.

Files in *Operating system and device management* introduces you to files and the way you can work with them.

Input and output redirection overview in *Operating system and device management* describes how the Operating System processes input and output.

---

comb Command (SCCS)

Purpose

Combines SCCS deltas.

Syntax

comb [**-o**] [**-s**] [**-c** *List* | **-p** *SID*] *File*

Description

The **comb** command writes to standard output a shell procedure that can combine specified SCCS deltas (SIDs) or all deltas into one delta. You can reduce the size of your Source Code Control System (SCCS) file by running the resulting procedure on the file. To see how much the file will be reduced, run the **comb** program with the **-s** flag. If you specify a directory for the *File* value, the **comb** command performs the requested actions on all SCCS files (that is, those having an **s.** prefix). If you specify a **-** (minus) for the *File* value, the **comb** command reads standard input and interprets each line as the name of an SCCS file. The **comb** command continues to take input until it reads an end-of-file character.

If you do not specify any flags, the **comb** command preserves only leaf deltas and the minimal number of ancestors needed to preserve the tree.

> **Note:** The **comb** command may rearrange the shape of the tree deltas. It may not save any space. In fact, it is possible for the reconstructed file to actually be larger than the original.

Flags

Note: Each flag or group of flags applies independently to each named file.

-c *List* Specifies a list of deltas (*SID*s) that the shell procedure will preserve (see the **get** command **-i** *List* flag). The procedure combines all other deltas.

-o Accesses the reconstructed file at the release of the delta to be created for each **get** command **-e** flag generated; otherwise, accesses the reconstructed file at the most recent ancestor. Using the **-o** flag may decrease the size of the reconstructed SCCS file. It may also alter the shape of the delta tree of the original file.

-p *SID* Specifies the *SID* of the oldest delta for the resulting procedure to preserve. All older deltas are combined in the reconstructed file.

-s Causes the **comb** command to generate a shell procedure that produces a report for each file listing: the file name, size (in blocks) after combining, original size (also in blocks), and percentage change computed by the formula:

```
100 * (original - combined) / original
```

You should run the **comb** program using this flag and then run its procedure before combining SCCS files in order to judge how much space will actually be saved by the combining process.

Examples

1. To generate a report on how much space will be saved by combining all deltas older than SID `1.4` for sccs file `s.test.c`, enter:

   ```
   comb -p1.4 -s s.test.c
   ```

   Run the report by piping the output of the above command to the **sh** command.
2. To actually perform the combination, enter:

   ```
   comb -p1.4 s.test.c
   ```

Files

s.COMB	The name of the reconstructed SCCS file.
comb*	Temporary files.

Related Information

The **admin** command, **delta** command, **get** command, **prs** command, **sccshelp** command, **sh** command.

The **sccsfile** file format.

List of SCCS Commands in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

Source Code Control System (SCCS) Overview in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

---

comm Command

Purpose

Selects or rejects lines common to two sorted files.

Syntax

comm [**-1** **-2** **-3**] *File1 File2*

Description

Note: If you specify - (minus) for one of the file names, the **comm** command reads standard input.

The **comm** command reads the *File1* and *File2* parameters and writes, by default, a three-column output to standard output. The columns consist of:

- Lines that are only in *File1*
- Lines that are only in *File2*
- Lines that are in both *File1* and *File2*.

Both *File1* and *File2* should be sorted according to the collating sequence specified by the current National Language environment.

Flags

-1	Suppresses the display of the first column (lines in *File1*).
-2	Suppresses the display of the second column (lines in *File2*).
-3	Suppresses the display of the third column (lines common to *File1* and *File2*).

Exit Status

This command returns the following exit values:

0 All input files were output successfully.
>0 An error occurred.

Examples

1. To display the lines unique to each file and common to both, enter:

   ```
   comm things.to.do things.done
   ```

   If the files `things.to.do` and `things.done` contain the following lists:

   ```
   things.to.do

   buy soap
   groceries
   luncheon
   meeting at 3
   system update
   tech. review

   things.done

   2nd revision
   interview
   luncheon
   system update
   tech. review
   weekly report
   ```

   then the **comm** command displays:

   ```
          2nd revision
   buy soap
   groceries
          interview
                   luncheon
   meeting at 3
                  system update
                  tech. review
          weekly report
   ```

   The first column contains the lines found only in `things.to.do`. The second column, indented with a tab character, lists the lines found only in `things.done`. The third column, indented with two tabs, lists the lines common to both.

2. To display the lines that appear in only one file, enter:

   ```
   comm -23 things.to.do things.done
   ```

   This suppresses the second and third columns of the **comm** command listing. If the files are the same as in Example 1, then the following is displayed:

   ```
   buy soap
   groceries
   meeting at 3
   ```

File

/usr/bin/comm Contains the **comm** command.

Related Information

The **cmp** command, **diff** command, **sdiff** command, **sort** command, **uniq** command.

The **environment** file.

Understanding Locale in *AIX 5L Version 5.3 National Language Support Guide and Reference*.

National Language Support Overview for Programming in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*.

---

command Command

Purpose

Executes a simple command.

Syntax

command [**-p**] *CommandName* [*Argument* ...]

command [**-v** | **-V**] *CommandName*

Description

The **command** command causes the shell to treat the specified command and arguments as a simple command, suppressing shell function lookup.

Normally, when a / (slash) does not precede a command (indicating a specific path), the shell locates a command by searching the following categories:

1. special shell built-ins
2. shell functions
3. regular shell built-ins
4. **PATH** environment variable

For example, if there is a function with the same name as a regular built-in, the system uses the function. The **command** command allows you to call a command that has the same name as a function and get the simple command.

The **command -v** and **command -V** commands write to standard output what path name will be used by the shell and how the shell interprets the command type (built-in, function, alias, and so forth). Since the **-v** and **-V** flags produce output in relation to the current shell environment, the **command** command is provided as a Korn shell or POSIX shell regular built-in command. The **/usr/bin/command** command might not produce correct results, because it is called in a subshell or separate command execution environment,. In the following example the shell is unable to identify aliases, subroutines, or special shell commands:

```
(PATH=foo command -v)
nohup command -v
```

Flags

-p Performs the command search using a default value for the **PATH** environment variable that is finds all of the standard commands.

-v Writes to standard output the path name used by the current shell to invoke the specified command, according to the following conventions:

- Commands, regular built-in commands, commands including a / (slash), and any implementation-provided functions found by the **PATH** environment variable are written as absolute path names.
- Shell functions, special built-in commands, regular built-in commands not associated with a **PATH** environment variable search, and shell reserved words are written as just their names.
- Aliases are identified as such, and their definitions are included in the string.

If the specified command name cannot be found, no output is written and the exit status returns a **>0** value.

-V Writes to standard output the command name that will be interpreted by the current shell environment. Although the format of this output is unspecified, The output indicates in which of the following categories the command falls:

- Commands, regular shell commands, and any implementation-provided subroutines found using the **PATH** environment variable are identified as such and written as absolute path names.
- Other shell functions are identified as functions.
- Aliases are identified as such, and their definitions are included in the string.
- Special built-in commands are identified as such.
- Regular built-in commands not associated with the **PATH** environment variable search are identified as such.
- Shell reserved words are identified as such.

Exit Status

When the **-v** or **-V** flag is specified, the following exit values are returned:

0 Successful completion.
>0 The command specified with the *CommandName* parameter could not be found or an error occurred.

When the **-v** or **-V** flag is not specified, the following exit values are returned:

126 The command specified by the *CommandName* parameter was found but could not be invoked.
127 An error occurred in the **command** command, or the command specified by the *CommandName* parameter could not be found.

Otherwise, the **command** command returns the exit status associated with the command specified by the *CommandName* parameter.

Examples

1. To make a version of the **cd** command that prints out the new working directory whenever you change directories, enter:

```
cd () {
        command cd "$@" >/dev/null
        pwd
}
```

2. To start off a secure shell script, one in which the script avoids being spoofed by its parent, enter:

```
IFS='
'
#       The preceding value should be <space><tab><newline>.
#       Set IFS to its default value

\unalias -a
#       Unset all possible aliases.
#       Note that unalias is escaped to prevent an alias
#       being used for unalias.

unset -f command
```

```
#       Ensure command is not a user function.

PATH="$(command -p getconf _CS_PATH):$PATH"
#       Put on a reliable PATH prefix.

# ...
```

At this point, given correct permissions on the directories called by the **PATH** environment variable, the script has the ability to ensure that any command it calls is the intended one.

Files

/usr/bin/ksh	Contains the Korn shell **command** built-in command.
/usr/bin/command	Contains the **command** command.

Related Information

The **ksh** command, **type** command.

---

comp Command

Purpose

Composes a message.

Syntax

comp [**+***Folder*] [**-draftfolder** **+***Folder* | **-nodraftfolder** *Folder*] [*Message* | **-draftmessage** *Message*] [**-file** *File*] [**-editor** *Editor* | **-noedit**] [**-form** *FormFile*] [**-use** | **-nouse**] [**-nowhatnowproc** | **-whatnowproc** *Program*]

Description

The **comp** command starts an editor that assists you in creating and modifying messages. The **comp** command provides a header template, the **/etc/mh/components** file. By default, the specified editor creates a *UserMhDirectory***/draft** file. If a **draft** file already exists, the **comp** command prompts you for permission to replace or use the existing file. To edit an existing **draft** file without being prompted for permission, specify the **-use** flag.

Once started, the editor prompts you to enter values for each of the message header fields. The **comp** command uses the definitions in the *UserMhDirectory***/components** file for the header fields. If the file does not exist, the **/etc/mh/components** file is used. You can use the **-form** or **+***Folder* flag to specify an alternative header format.

To exit the editor, use the Ctrl-D sequence. When you exit the editor, the **comp** command responds with a `What now?` prompt. From this prompt, you can specify any of the **whatnow** subcommands. To see a list of the available subcommands, press Enter. You can use the subcommands to continue composing the message, direct the disposition of the message, or end the processing of the **comp** command.

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

The **-file**, **-draftfolder**, and **-draftmessage** flags are used to specify existing draft messages. If the **-draftfolder** **+***Folder* flag is followed by a *Message* parameter, it is the same as specifying the **-draftmessage** flag. If you wish, you can define a default `Draft-Folder:` entry in your Message Handler (MH) **$HOME/.mh_profile** file.

Flags

-draftfolder +*Folder*
Identifies the folder containing the draft message. If a message is not specified with this flag, the default message is **new**.

-draftmessage *Message*
Identifies the draft message. Specifying a *Message* variable after the **-draftfolder +***Folder* flag is the same as specifying the **-draftmessage** flag.

-editor *Editor*
Specifies the initial editor for composing the message. If you do not specify the **-editor** flag, the **comp** command selects the default editor specified by the `Editor:` entry in your **$HOME/.mh_profile** file.

-file *File*
Places the draft message in the specified file. If you do not specify the absolute path name for the *File* variable, the **comp** command places the file in the user's MH directory. If a file is specified, the **comp** command prompts you for the disposition of the draft.

+*Folder Message*
Uses the header information from a file in the specified folder. If you specify a folder but no message, the **comp** command uses the current message as the default.

-form *FormFile*
Uses the header fields specified by the *FormFile* variable. The **comp** command treats each line in *FormFile* as a format string.

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

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

Message
Specifies a message. Use the following references to specify a message:

Number
Number of the message.

cur or . (period)
Current message. This is the default.

first First message in a folder.

last Last message in a folder.

next Message following the current message.

prev Message preceding the current message.

-nodraftfolder
Places the draft in the *UserMhDirectory***/draft** file. This is the default.

-noedit
Suppresses the initial edit. When you specify this flag, you receive the `What now?` prompt.

-nouse
Creates a new message.

-nowhatnowproc
Prevents interaction with the editor and the `What now?` prompt.

-use
Continues composing an existing draft of a message.

-whatnowproc *Program*
Starts the specified program to guide you through the composing tasks. If you specify the **whatnow** command as the value for the *Program* variable, the **comp** command starts an internal **whatnow** procedure, instead of a program with the file name **whatnow**.

Security

RBAC Disclaimer

Profile Entries

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

`Draft-Folder:` Sets the default folder for drafts.
`Editor:` Sets the default initial editor.
`fileproc:` Specifies the program used to refile messages.
`Msg-Protect:` Sets the protection level for the new message files.
`Path:` Specifies a user's MH directory.

`whatnowproc:` Specifies the program used to prompt `What now?` questions.

Examples

1. To compose a new message, enter:

   ```
   comp
   ```

   The system prompts you to enter the information for the message fields. To bypass a field, press the Enter key. When the header information is complete, enter the text for the body of the message.

   To finish composing a message and exit the editor, press the Ctrl-D sequence. The following prompt is displayed on your screen:

   ```
   What now?
   ```

   Pressing the Enter key displays a list of the **whatnow** subcommands. If you want to send the message, enter the **send** subcommand after the `What now?` prompt.
2. To compose a new message using the `vi` editor, enter:

   ```
   comp  -editor vi
   ```
3. To compose a message using message `8` in the `schedules` folder, enter:

   ```
   comp  +schedules 8  -use
   ```
4. To compose a message using a message draft in the `/home/mike/parts` file, enter:

   ```
   comp  -file /home/mike/parts
   ```

   The system prompts you for the disposition of the file. Press the Enter key for a list of options. Select the appropriate option.

Files

UserMhDirectory/**components**	Specifies the user's default message format. (If it exists, it overrides the system default message format.)
UserMhDirectory/**draft**	Contains the current draft message.
$HOME/.mh_profile	Specifies the user's MH profile.
/etc/mh/components	Identifies the system default message format.
/usr/bin/comp	Contains the **comp** command.

Related Information

The **ali** command, **dist** command, **forw** command, **refile** command, **repl** command, **send** command, **whatnow** command, **whom** command.

The **mh_alias** file format, **mh_profile** file format.

Mail applications in *Networks and communication management*.

---

compare_report Command

Purpose

Compares fileset levels to those available and generates a report of filesets needed.

Syntax

To compare filesets installed on a system to filesets contained in a fix repository:

compare_report -s -i *FixDir* { [**-l**] [**-h**] [**-m**] [**-n**] } [[[**-t** *ReportDir*] [**-Z**]] | **-v**]

To compare filesets installed on a system to filesets available from the support Web site:

compare_report -s -r *ServiceReport* { [**-l**] [**-h**] } [[[**-t** *ReportDir*] [**-Z**]] | **-v**]

To compare filesets contained in a fix repository to filesets available from the support Web site:

compare_report -i *FixDir* **-r** *ServiceReport* [[[**-t** *ReportDir*] [**-Z**]] | **-v**]

To compare a list of installed software on a base system to another system:

compare_report -b *BaseList* **-o** *OtherList* { [**-l**] [**-h**] [**-m**] [**-n**] } [[[**-t** *ReportDir*] [**-Z**]] | **-v**]

To compare a list of installed software to filesets contained in a fix repository:

compare_report -b *BaseList* **-i** *FixDir* { [**-l**] [**-h**] [**-m**] [**-n**] } [[[**-t** *ReportDir*] [**-Z**]] | **-v**]

To compare a list of installed software to filesets available from the support Web site:

compare_report -b *BaseList* **-r** *ServiceReport* { [**-l**] [**-h**] } [[[**-t** *ReportDir*] [**-Z**]] | **-v**]

Description

The **compare_report** command compares the filesets installed on a system with the contents of a fix directory or a service report that contains a list of the latest available fixes. The comparison reports produced provide assistance in assuring a system is running a certain level of software. The fix directory can be an image repository, such as an **lpp_source**. The service report is a list of both the latest level fixes and the fixes contained in the latest technology level and can be downloaded from the IBM System p Support for AIX 5L and Linux servers site (http://www-03.ibm.com/servers/eserver/support/unixservers/index.html). Some of the generated reports can be used as input to request fixes from the IBM System p Support for AIX 5L and Linux servers site.

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 must be run by root, and 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.

Flags

-b *BaseList*
: The name of the file containing the software installed on the base system (generated with **lslpp -Lc**)

-h
: Indicates that the higher level fileset reports should be generated. This will generate one or all of the reports **higherlevel.rpt**, **higherthanmaint.rpt**, or **basehigher.rpt**, depending on which comparisons are performed. This flag is only valid when used either with the **-s** or with both the **-b** and the **-o** flags.

-i *FixDir*
: Specifies the name of the fix repository directory. The fileset levels of the images contained in this directory will be used in the comparison.

-l Indicates that the lower level fileset reports should be generated. This will generate one or all of the reports **lowerlevel.rpt**, **lowerthanlatest1.rpt**, **lowerthanmaint.rpt**, **lowerthanlatest2.rpt**, or **baselower.rpt**, depending on which of the comparisons are performed. This flag is only valid when used either with the **-s** or with both the **-b** and the **-o** flags.

-m Indicates that a fileset report should be generated that lists either the filesets installed on the system that are not in the image repository, or the filesets installed on the base system that are not installed on the other system. This will generate either the **no_update_found.rpt** or the **baseonly.rpt** report file. This flag is only valid when both the **-s** and **-i** flags are specified or when both the **-b** and **-o** flags are specified.

-n Indicates that a fileset report should be generated that lists either the filesets in the image repository that are not installed on the system or the filesets installed on the other system that are not installed on the base system. This will generate either the **notinstalled.rpt** or the **otheronly.rpt** report file. This flag is only valid when both the **-s** and **-i** flags are specified or when both the **-b** and **-o** flags are specified.

-o *OtherList* The name of the file containing the software installed on another system that will be compared to a base system (generated with the command **lslpp -Lc**).

-r *ServiceReport* Specifies a file that contains the list of available updates. This file can be obtained from the support Web site.

-s Specifies that the comparison should involve a list of the fileset levels that are installed on this system.

-t *ReportDir* Specifies the target directory where the comparison reports will be stored. If the **-t** flag is not specified, the reports will be stored in the **/tmp** directory. If report files already exist in the specified directory, they will be removed and recreated. This flag is not valid with the **-v** flag.

-v Specifies that no report files should be saved to disk. This flag is not valid with the **-t** or **-Z** flags.

-Z Suppresses displaying the report output to stdout. This flag is not valid with the **-v** flag.

Exit Status

0 The command completed successfully.

>0 An error occurred.

Examples

1. To compare filesets installed on the system to filesets contained in an image repository, type:

```
compare_report -s -i /tmp/imagedir -l -n
```

This command will create reports listing filesets on the system that are at a lower level and filesets in the image repository that are not installed on the system. If all reports (**-l**, **-h**, **-m**, **-n**) are requested for this type of comparison, the following reports will be generated:

- **lowerlevel.rpt** (generated with **-l**)
- **higherlevel.rpt** (generated with **-h**)
- **no_update_found.rpt** (generated with **-m**)

- **notinstalled.rpt** (generated with **-n**)

2. To compare filesets installed on the system to filesets available from the support Web site, type:

   ```
   compare_report -s -r /tmp/LatestFixData -l -Z
   ```

   This command will create reports listing filesets on the system that are at a lower level from the latest levels, and those that are at a lower level than the last technology level. The reports will be saved to disk but not displayed to stdout. If all reports (**-l**, **-h**) are requested for this type of comparison, the following reports will be generated:
   - **lowerthanlatest1.rpt** (generated with **-l**)
   - **lowerthanmaint.rpt** (generated with **-l**)
   - **higherthanmaint.rpt** (generated with **-h**)
3. To compare filesets contained in an image repository to filesets available from the support Web site, type:

   ```
   compare_report -i /tmp/imagedir -r /tmp/LatestFixData
   ```

   This command creates a report listing filesets in the image repository that are at lower levels than the latest levels available from the support Web site. The **lowerthanlatest2.rpt** report is the only report generated from this type of comparison.
4. To compare a list of installed software on a base system to a list of installed software on another system, type:

   ```
   compare_report -b /tmp/base.lslpp.out -o /tmp/other.lslpp.out -l -h -m -n
   ```

   This command will create reports listing the following:
   - filesets on the base system that are at a lower level than the other system
   - filesets on the base system that are at a higher level than the other system
   - filesets installed on the base system that are not installed on the other system
   - filesets installed on the other system that are not installed on the base system

   If all reports (**-l**, **-h**, **-m**, and **-n**) are requested for this type of comparison, the following reports will be generated respectively:
   - **baselower.rpt** (generated with **-l**)
   - **basehigher.rpt** (generated with **-h**)
   - **baseonly.rpt** (generated with **-m**)
   - **otheronly.rpt** (generated with **-n**)

Files

/usr/sbin/compare_report	Contains the **compare_report** command.

compress Command

Purpose

Compresses data.

Syntax

compress [**-c**] [**-C**] [**-d**] [**-F**] [**-f**] [**-n**] [**-q**] [**-v**] [**-V**] [**-b** *Bits*] [*File* ...]

Description

The **compress** command compresses data, using adaptive Lempel-Zev coding to reduce the size of files. Each original file specified by the *File* parameter is replaced when possible by a compressed file with a **.Z** appended to its name. If the invoking process has appropriate privileges, the compressed file retains the same ownership, modes, and modification time of the original file. If the path of the file specified is more than 1023 bytes the command does not work. If no files are specified, the standard input is compressed to the standard output. If compression does not reduce the size of a file, a message is written to standard error and the original file is not replaced.

Note: Files must have correct permissions to be replaced.

The amount of compression depends on the size of the input, the number of bits per code specified by the *Bits* variable, and the distribution of common substrings. Typically, source code or English text is reduced by 50 to 60%. The compression of the **compress** command is usually more compact and takes less time to compute than the compression achieved by Huffman coding (as used in the **pack** command) or adaptive Huffman coding.

Flags

-b *Bits* Specifies the maximum number of bits to use to replace common substrings in the file. The value of the *Bits* variable must be in the range from 9 bits through 16 bits, with the default being 16 bits. When compressing data, the algorithm first uses all of the 9-bit codes (257 through 512) to replace as many substrings as possible. Then it uses all 10-bit codes, and so on, continuing until the limit specified by the **-b** flag is reached.

-c Writes to standard output. No files are changed.

-C Produces output compatible with the Berkeley Software Distribution (BSD) Revision 2.0.

-d Causes the **compress** command to function exactly like the **uncompress** command.

-f or **-F** Forces compression. The **-f** and **-F** flags are interchangeable. Overwrites the *File***.Z** file if it already exists.

After the value of the *Bits* variable is attained, the **compress** command periodically checks the compression ratio. If it is increasing, the **compress** command continues to use the existing code dictionary. However, if the compression ratio decreases, the **compress** command discards the table of substrings and rebuilds it. Rebuilding the table allows the algorithm to adapt to the next block of the file. When the **.Z** file already exist, if the **-f** flag is not given, and the process is not running in the background, it prompts to verify whether to overwrite the existing **.Z** file.

-n Omits the compressed file header from the compressed file.
Note: If this option is used, the **-n** flag should also be used when using the **uncompress** command to uncompress the file.

-q Suppresses the display of compression statistics generated by the **-v** flag. If several **-v** and **-q** flags are on the same command line, the last one specified controls the display of the statistics.

-v Writes the percentage of compression.

-V Writes the current version and compile options to standard error.

Parameters

File Specifies the file to compress.

Return Values

If an error occurs, the exit status is 1. If the **compress** command exits without compressing a file, it exits with a status of 2. Otherwise, the **compress** command exits with a status of 0.

The **compress** command detects an error and exits with a status of 1 if any of the following events occur:

- An input file is not a regular file.

- An input file name is too long to append the **.Z** extension.
- An input file cannot be read or an output file cannot be written.

Exit Status

0 Successful completion.
1 An error occurred.
2 One or more files were not compressed because they would have increased in size (and the **-f** flag was not specified).
>2 An error occurred.

Security

RBAC Disclaimer

Example

To compress the `foo` file and write the percentage of compression to standard error, enter:

```
compress -v foo
```

The `foo` file is compressed and renamed `foo.Z`.

Related Information

The **pack** command, **uncompress** command, **unpack** command, **zcat** command.

Commands in *Operating system and device management*.

---

comsat Daemon

Purpose

Notifies users of incoming mail.

Syntax

/usr/sbin/comsat [**-d** *Directory*]

Description

The **comsat** daemon is the server that receives reports of incoming mail and notifies users if they have enabled this service with the **biff** command. Started by the **inetd** daemon, the **comsat** daemon is not meant to be used at the command line. The **comsat** daemon receives messages on a datagram port associated with the **biff** service specification. The one-line messages are of the form:

```
user@mailbox-offset
```

If the user specified is logged in to the system and has run the **biff y** command, the first 7 lines or 560 characters of the message are displayed on the user's login terminal. Lines that appear to be part of a message header other than the `From:` or `Subject:` lines are not included in the displayed message.

Flags

-d *Directory* Specifies the name of the directory to use as the system mail directory. If the **-d** flag is not specified, the **comsat** daemon uses the **/var/spool/mail** directory as the default system mail directory.

Files

/etc/utmp	Contains a list of users who are logged in, including their terminals.
/etc/services	Contains a list of Internet network services and the well-known ports where the servers accept connections.

Related Information

The **biff** command.

The **inetd** daemon.

The **inetd.conf** file format.

Mail management in *Networks and communication management*.

---

configassist Command

Purpose

Displays the Configuration Assistant wizard.

Syntax

/usr/websm/bin/configassist

Description

The Configuration Assistant wizard displays automatically after the operating system is installed and is used to assist with configuration tasks. It can also be run at any time to complete additional configuration. Use the Configuration Assistant to configure a system that has an HTTP Server installed to run Web-based System Manager in a browser. See Applet Mode on page 4 for more informatoin.

Note: The full pathname of this command, **/usr/websm/bin/configassist**, must be specified.

Flags

None

Examples

N/A

Related Information

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

---

conflict Command

Purpose

Searches for alias and password conflicts.

Syntax

conflict [**-mail** *User*] [**-search** *Directory ...*] [*File ...*]

Description

The **conflict** command finds invalid mail drops and alias conflicts. The **conflict** command is not started by the user. The **conflict** command is called by the **cron** daemon and other programs used for system accounting. However, root user authority and the full path name of the command, **/usr/lib/mh/conflict**, are required to invoke the program.

The **conflict** command searches specified mail drop directories for mailbox files with names that do not correspond to valid users in the **/etc/passwd** file. In addition, the program searches alias files specified by the *File* parameter for duplicate names that do not resolve to the same address. By default, the **conflict** command searches the **/etc/mh/MailAliases** file.

The **conflict** command also searches entries in the group file (**/etc/group**) for invalid user names and users who do not have a valid group number.

Command output is to the monitor unless you specify the **-mail** flag. The **-mail** flag sends the command output to the specified user.

Flags

-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.

-mail *User* Sends the results of the **conflict** command to the user specified by the *User* variable.

-search *Directory* Searches the directory indicated by the *Directory* variable for mailboxes that are not valid. You can specify any number of **-search** flags. The default mailbox directory is **/var/spool/mail**.

Files

/etc/mh/MailAliases	Contains the default mail alias file.
/etc/passwd	Contains a list of users.
/etc/group	Contains a list of groups.
/var/spool/$USER	The mail drop for the user **$USER.**
/$HOME/.mh_profile	Contains the MH user profile.
/etc/mh/mtstailor	Contains MH command definitions.

Related Information

The **ali** command, **whom** command.

The **mh_alias** file format.

Mail applications in *Networks and communication management*.

---

confsetcntrl Command

Purpose

Manage a set of time-based Workload Manager (WLM) configurations.

Syntax

confsetcntrl -C *ConfigurationSet DefaultConfig*

confsetcntrl { **-D** | **-R** } *ConfigurationSet*

confsetcntrl [**-d** *ConfigurationSet*] { **-a** | **-r** } *Configuration TimeRange*

confsetcntrl [**-d** *ConfigurationSet*] [**-l** | **-c**]

Description

The **confsetcntrl** command supports the following operations:

- Create a new configuration set with its initial default regular configuration.
- Delete an existing configuration set (this includes the configuration set directory and its **.times** and **description** files, but does not affect the regular configurations of the set).
- Add or remove from a configuration set a configuration and its associated time range.
- Remove from a configuration set all configurations and associated time ranges.
- Check the configuration set file.
- List all the configurations contained in a set with their associated time ranges.

Note: Only the root user can create, delete, or change configuration sets, but any user can list or check them.

Time Ranges

Time ranges are used to indicate at which day of the week and which times of the day the associated configuration will be used by the WLM for classifying processes, for accounting, and regulation.

A time range is represented by a range of days, with 0 representing Sunday and 6 representing Saturday, and a range of time, in 24 hour format with hours and minutes specified. These two ranges are separated with a comma. In each range, values are separated with a minus sign, and values may wrap (the first value may be greater than the second one).

The range of days may be omitted, which means every day of the week. Both ends of this range are included. It may then also consist in only one day: 1 is valid and stands for 1-1.

The range of time may be omitted, which means the whole day. Elsewhere, start and end times must be specified. Hours and minutes are separated with a colon or a dot. The end time is not part of the range, so 24:00 is a valid end time but 12:00-12:00 is empty and not valid.

At least one of the day or time ranges must be present. A single minus sign is a valid time range and is a special case: It is called the default time range and means always outside the other defined time ranges if any. This is different from specifying all the time, for example with 0-6,00:00-24:00

For the WLM to be able to find which configuration must be activated, there must exist one and only one configuration applicable at any time of the week. The default time range, which is added when creating a set, is useful to avoid the possibility that no configuration would be applicable for some time. Additional time ranges must not overlap with each other.

Time range examples:

1-4,8:00-17:00
Monday to Thursday, from 8 am to 5 pm

5-0,22:00-6:00
Friday, Saturday and Sunday, from midnight to 6 am and from 10 pm to midnight

3 Wednesday

14:00-16:30
Every day from 2 pm to 4:30 pm

- The default time range

Flags

-a *Configuration TimeRange*
Adds *Configuration* to the configuration set for the given *TimeRange*. *Configuration* must be an existing WLM regular configuration. It may appear several times in a set associated with different time ranges.
Note: Even if time ranges become not coherent due to this operation, the changes are performed, but a warning is reported indicating that further changes are needed.

-c
Checks all the configuration/time range pairs of the set.

-C *ConfigurationSet DefaultConfig*
Creates configuration set *ConfigurationSet* with *DefaultConfig* initial configuration, having default time range. (The default time range means always outside any other explicit time range. Only one is allowed in a set.) *DefaultConfig* must be an existing WLM regular configuration.

-d *ConfigurationSet*
Specifies an alternate configuration set. If not given, current configuration set will be the target of the command.

-D *ConfigurationSet*
Deletes configuration set *ConfigurationSet*.

-l
Checks and lists all the configuration/time range pairs of the set. This is the default operation if no flag is given.

-r *Configuration TimeRange*
Removes the *Configuration* and *TimeRange* pair from the configuration set. This pair is supposed to exist in the set.
Note: Even if time ranges become not coherent due to this operation, the changes are performed, but a warning is reported indicating that further changes are needed.

-R *ConfigurationSet*
Erases configuration set *ConfigurationSet* (removes from *ConfigurationSet* all configuration/time range pairs). This operation is not recommended as the resulting configuration set state is not consistent and requires additional changes.

Security

RBAC Disclaimer

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
   ```

```
fvtregul
fvtdfct
fvtsynt
fvtthreads
```

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, type:

   ```
   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. To use the **lswlmconf** command to show the new configuration set, type:

   ```
   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
fvtthreads
```

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
fvtlimits
fvtregul
fvtdfct
fvtsynt
fvtthreads
confset1/fvtregul
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/fvtregul
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″, using the **wlmcheck** command, type:

```
wlmcheck -d confset1
```

In this example, this produces the following output:

```
WLM is not running.
Checking classes and rules for 'confset1' configuration...
fvtlimits/System
fvtlimits/Default
fvtlimits/Shared
fvtlimits/login
fvtregul/System
fvtregul/Default
fvtregul/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 configuration set files reside in a subdirectory of **/etc/wlm** whose name is the set name.

.times	Contains the list of all the configuration/time range pairs of the set.
description	Contains an optional description text of the set.

Related Information

The **wlmcntrl** command, **lswlmconf** command.

---

confsrc Command

Purpose

The **confsrc** command configures a subsystem, a group of subsystems, or a subserver.

Syntax

confsrc [[**-R**] **-h** Host] [**-p** *SubsystemPID*] [**-q**] [**-Q**] [**-u** *UserID*] [**-U** *Password*] **-s** *Subsystem* **-a** *ConfigString*

Description

The **confsrc** command sends a request to the System Resource Controller (SRC) to configure a subsystem. When a request to configure the subserver is passed to the SRC and the subsystem to which the subserver belongs is not active, the SRC starts the subsystem and transmits the request to the subsystem.

Note: The configure subserver request is processed only when the subsystem supports the request.

Flags

-a *ConfigString*	Specifies a string containing the configuration information. This string is passed from the command line and appended to the command-line arguments from the subsystem object class. If the specified string exceeds 1200 characters, the command is unsuccessful. The command argument is passed by the SRC to the subsystem according to the rules used by the shell. Quoted strings are passed as a single argument, and blanks outside a quoted string delimit an argument. Single and double quotes can be used in the string.
-h *Host*	Specifies the foreign host on which the configure action is requested. The local user must be running as root. The remote system must be configured to accept remote SRC requests by starting the **srcmstr** daemon (**/etc/inittab**) with the **-r** flag and configuring the **/etc/hosts.equiv** or**rhosts** file to allow remote requests.
-p *SubsystemPID*	Specifies an instance of the subsystem to which the configure request is passed.
-q	Specifies the quiet mode of operation with minimum local output.
-Q	Specifies the quiet mode of operation with suppressed output.
-R	Uses TCP for remote connections. **Note:** This flag is active only when the **-h** flag is used.
-s *Subsystem*	Specifies the subsystem to be started. The specified *Subsystem* can be the actual subsystem name or the synonym name for the subsystem. The command is unsuccessful if the specified *Subsystem* is not contained in the subsystem object class.
-u *UserID*	Specifies the user ID used on the remote host. **Note:** This flag is active only when the **-h** flag is used.
-U *Password*	Specifies the password for the user ID. **Note:** This flag is active only when the **-u** flag is used.

Security

RBAC Disclaimer

Files

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

Related Information

The **startsrc** command, the **stopsrc** command, the **refresh** command.

The System resource controller in *Operating system and device management* gives an explanation of subsystems, subservers, and the SRC.

The Trusted AIX in *Security*.

The RBAC in *Security*.

cp Command

Purpose

Copies files.

Syntax

To Copy a File to another File

cp [**-d**] [**-e**] [**-E**{**force**|**ignore**|**warn**}] [**-f**] [**-h**] [**-i**] [**-p**] [**-I**] [**-U**] [**-**] *SourceFile TargetFile*

To Copy a File to a Directory

cp [**-d**] [**-e**] [**-E**{**force**|**ignore**|**warn**}] [**-f**] [**-h**] [**-i**] [**-p**] [[**-r** | **-R**] [**-H** | **-L** | **-P**]] [**-I**] [**-U**] [**-**] *SourceFile ... TargetDirectory*

To Copy a Directory to a Directory

cp [**-d**] [**-e**] [**-E**{**force**|**ignore**|**warn**}] [**-f**] [**-h**] [**-i**] [**-p**] { **-r** | **-R** } [**-H** | **-L** | **-P**] [**-I**] [**-U**] [**-**] *SourceDirectory ... TargetDirectory*

Description

The **cp** command copies the source file specified by the *SourceFile* parameter to the destination file specified by the *TargetFile* parameter. If the target file exists, **cp** overwrites the contents, but the mode, owner, and group associated with it are not changed. The last access time of the *SourceFile* and the last modification time of the *TargetFile* are set to the time the copy was done. If the *TargetFile* does not exist, **cp** creates a new file named *TargetFile* that has the same mode as the source file except that the sticky bit is not set unless it was done by a superuser; the owner and group of the *TargetFile* is that of the user. When the *TargetFile* is a link to another file, **cp** overwrites the destination link with the content of the source file; the links from the *TargetFile* remains. Also, the **cp** command can copy the source files specified by the *SourceFile* parameter (or directories named by the *SourceDirectory* parameter) to the directory specified by the *TargetDirectory* parameter.

Note: If one of the source parameters is a directory, you need to specify one of the **-r** or **-R** flags.

If any directories are created by the **cp** command during the copying process, the newly created directory will have the same mode as the corresponding source directory.

You can also copy special device files. The preferred option for accomplishing this is the **-R** flag. Specifying **-R** causes the special files to be re-created under the new path name. Specifying the **-r** flag causes the **cp** command to attempt to copy the special file to a regular file.

Flags

-d Specifies that the source file is stored in decrypted (clear-text) format on target.

-e Specifies that the source file is stored in encrypted form, if the target file system is an Encrypted File System (EFS).

-E The **-E** option requires one of the following arguments. If you omit the **-E** option, **warn** is the default behavior.

- **force** Fails the **cp** 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 Specifies removal of the target file if it cannot be opened for write operations. The removal precedes any copying performed by the **cp** command.

-h Forces the **cp** command to copy symbolic links. The default is to follow symbolic links, that is, to copy files to which symbolic links point.

-H Take actions based on the type and contents of the file referenced by any symbolic link specified as a *SourceFile* operand.

-i Prompts you with the name of a file to be overwritten. This occurs if the *TargetDirectory* or *TargetFile* parameter contains a file with the same name as a file specified in the *SourceFile* or *SourceDirectory* parameter. If you enter `y` or the locale's equivalent of `y`, the **cp** command continues. Any other answer prevents the **cp** command from overwriting the file.

-I Suppresses the warning message during ACL conversion.

-L Take actions based on the type and contents of the file referenced by any symbolic link specified as a *SourceFile* operand or any symbolic links encountered during traversal of a file hierarchy.

-p Duplicates the following characteristics of each *SourceFile*/*SourceDirectory* in the corresponding *TargetFile* and/or *TargetDirectory*:

- The time of the last data modification and the time of the last access. If this duplication fails for any reason, the **cp** command will write a diagnostic message to standard error. The **nanoseconds** field of the *SourceFile*/*SourceDirectory* is not duplicated for last modification time or last access time.
- The user ID and group ID. If this duplication fails for any reason, the **cp** command may write a diagnostic message to standard error.
- The file permission bits and the S_ISUID and S_ISGID bits. If this duplication fails for any reason, the **cp** command will write a diagnostic message to standard error.

If the user ID or group ID cannot be duplicated, the file permission bits S_ISUID and S_ISGID are cleared.

In order to preserve the owner ID and group ID, permission modes, modification and access times, user must have the appropriate file access permissions (user should be a superuser or have the same owner ID as the destination file)

The target file will not be deleted if these characteristics cannot be preserved.

Access control lists (ACLs) associated with the *SourceFile* are preserved if the target filesystem supports the same. If the source file contains NFS4 ACL and the target filesystem does not support NFS4 ACL, the NFS4 ACL is converted to AIXC. See User accounts, in *Security* to learn more about ACLs.

When ACL conversion succeeds, a warning message is printed out the stderr.

If the source file is encrypted and the **-p** flag is specified, the **cp** command preserves the EFS information. Generally, the **-e** or **-d** flag takes precedence over the **–p** flag. If a user requests to convert a clear-text file to an encrypted format using the **-e** flag, then even if the user specifies the **-p** flag, the copy does not preserve attributes like the time of the last data modification, the time of the last access and so on. As long as the encryption or decryption status remains the same, the **-p** flag preserves the file attributes and EFS information.

-P Take actions on any symbolic link specified as a *SourceFile* operand or any symbolic link encountered during traversal of a file hierarchy.

-r Copies file hierarchies under the file or directory specified by the *SourceFile* or *SourceDirectory* parameter (recursive copy). The **-r** flag processes special files in the same manner as regular files.

-R Copies file hierarchies under the regular files and directories from the directory specified by the *SourceFile* or *SourceDirectory* parameter to the directory specified by the *TargetDirectory* parameter. Special file types, such as first-in, first-out (FIFO) files and block and character device files, are re-created instead of copied. Symbolic links are followed unless the **-h** flag is specified. (The **-R** flag is preferred to the **-r** flag.)

If none of the **-H**, **-L**, or **-P** options were specified, it is unspecified which of those options will be used as the default. Consider the following:

- If the **-H** option was specified, the **cp** command will take action based on the type and contents of the file referenced by any symbolic link specified as a *SourceFile* operand.
- If the **-L** option was specified, the **cp** command will take action based on the type and contents of the file referenced by any symbolic link specified as a *SourceFile* operand or any symbolic links encountered during traversal of a file hierarchy.
- If the **-P** option was specified, the **cp** command will copy any symbolic link specified as a *SourceFile* operand and any symbolic links encountered during traversal of a file hierarchy and will not follow any symbolic links.

-U Copies Extended Attributes (EA), Access Control Lists (ACL) in the *SourceFile* to the *TargetFile*. If the EA is not supported on the target filesystem then it is ignored. If the source ACL type is not supported on the target filesystem then it is converted to the compatible ACL type supported by the target filesystem.

-- Indicates that parameters following the **--** (dash, dash) flag are to be interpreted as file names. This null flag allows the specification of file names that start with a **-** (minus sign).

The following table shows the encryption or decryption status of the target file under different conditions:

Explicit flag for the cp 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.

Exit Status

This command returns the following exit values:

0	All files were copied successfully.
>0	An error occurred.

Examples

1. To make a copy of a file in the current directory, enter:

   ```
   cp prog.c prog.bak
   ```

   This copies `prog.c` to `prog.bak`. If the `prog.bak` file does not already exist, the **cp** command creates it. If it does exist, the **cp** command replaces it with a copy of the `prog.c` file.
2. To copy a file in your current directory into another directory, enter:

   ```
   cp jones /home/nick/clients
   ```

   This copies the `jones` file to `/home/nick/clients/jones`.
3. To copy a file to a new file and preserve the modification date, time, and access control list associated with the source file, enter:

   ```
   cp  -p smith smith.jr
   ```

   This copies the `smith` file to the `smith.jr` file. Instead of creating the file with the current date and time stamp, the system gives the `smith.jr` file the same date and time as the `smith` file. The `smith.jr` file also inherits the `smith` file's access control protection.
4. To copy all the files in a directory to a new directory, enter:

   ```
   cp /home/janet/clients/* /home/nick/customers
   ```

   This copies only the files in the `clients` directory to the `customers` directory.
5. To copy a directory, including all its files and subdirectories, to another directory, enter:

   ```
   cp  -R /home/nick/clients /home/nick/customers
   ```

   Note: A directory cannot be copied into itself.
   This copies the `clients` directory, including all its files, subdirectories, and the files in those subdirectories, to the `customers/clients` directory.
6. To copy a specific set of files to another directory, enter:

   ```
   cp jones lewis smith /home/nick/clients
   ```

   This copies the `jones`, `lewis`, and `smith` files in your current working directory to the `/home/nick/clients` directory.
7. To use pattern-matching characters to copy files, enter:

   ```
   cp programs/*.c .
   ```

   This copies the files in the `programs` directory that end with `.c` to the current directory, signified by the single `.` (dot). You must type a space between the `c` and the final dot.
8. To copy a file to a new file and preserve the ACL and EA associated with the source file, enter:

   ```
   cp -U smith smith.jr
   ```

Files

/usr/bin/cp
Contains the **cp** command.

Related Information

The **cpio** command, **link** command, **ln** command, **mv** command, **unlink** command.

Directories in *Operating system and device management*.

National Language Support Overview in *AIX 5L Version 5.3 National Language Support Guide and Reference*.

---

cpcosi Command

Purpose

Clones a Common Operating System Image (COSI).

Syntax

cpcosi -c *COSI* [**-S** *Server*] [**-l** *Location*] [**-v**] *COSI*

Description

The **cpcosi** command clones a Common Operating System Image (COSI). A COSI is a repository that contains all the software necessary to bring up a system to a functional state. The **mkcosi** command creates the COSI.

The **cpcosi** command takes a common image and attempts to make a duplicate copy of it. The copied version is stored at the location specified with the **-l** flag. If the **-l** flag is not specified, the location of the originating common image is used instead. If the **-S** flag is specified, the clone common image is stored on that particular server. The **-S** flag must point to a machine that is managed by the caller of the **cpcosi** command. The naming convention for the clone is the original common image name suffixed with an _X{*count*}, where *count* is a number that is incremented every time a common image is cloned.

A common image must exist on the system before it can be cloned. Use the **mkcosi** command to create a common image. The **lscosi** command lists any common images that exist in the environment. The **lscosi** command depends on the **bos.sysmgt.nim.master** fileset being present on the system.

Flags

-c	Specifies the COSI to clone.
-l *Location*	Specifies the full path name to a location for storing the COSI.
-S *Server*	Specifies the name of the machine on which the COSI image will reside.
-v	Enables verbose debug output when the **cpcosi** command runs.

Exit Status

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

Security

Access Control: You must have root authority to run the **cpcosi** command.

Examples

1. To clone a COSI named `cosi2` from a COSI named `cosi1`, enter:

   ```
   cpcosi -c cosi1 cosi2
   ```

   Because no location path was specified in the preceding example, if `cosi1` was stored at **/export/cosi1**, the cloned COSI will be placed in **/export/cosi2**.

Location

/usr/sbin/cpcosi

Files

/etc/niminfo	Contains variables used by NIM.

Related Information

The "chcosi Command" on page 368, **lscosi** command, **mkcosi** command, **mkts** command, **nim** command, **nim_clients_setup** command, **nim_master_setup** command, **nimconfig** command, **rmcosi** command.

---

cpio Command

Purpose

Copies files into and out of archive storage and directories. This document describes the AIX **cpio** command and the System V **cpio** command.

Syntax

cpio -o [**a**] [**c**] [**-E**{**force**|**ignore**|**warn**}] [**-U**] [**v**] [**B** | **C** *Value*] [**-Z**] <*FileName* >*Output*

cpio -i [**b**] [**c**] [**d**] [**-E**{**force**|**ignore**|**warn**}] [**f**] [**m**] [**M**] [**r**] [**s**] [**t**] [**-U**] [**u**] [**v**] [**S**] [**6**] [**B** | **C** *Value*] [**-Z**] [*Pattern...*] <*Input*

cpio -p [**a**] [**d**] [**-E**{**force**|**ignore**|**warn**}] [**l**] [**m**] [**M**] [**-U**] [**u**] [**v**] [**-Z**] *Directory* <*FileName*

Description

Attention: If you redirect the output from the **cpio** command to a special file (device), you should redirect it to the raw device and not the block device. Because writing to a block device is done asynchronously, there is no way to know if the end of the device is reached.

Notes:

1. The **cpio** command is not enabled for files greater than 2GB in size due to limitations imposed by XPG/4 and POSIX.2 standards.
2. **cpio** does not preserve the sparse nature of any file that is sparsely allocated. Any file that was originally sparse before the restoration will have all space allocated within the filesystem for the size of the file.
3. You cannot use the System V **cpio** command for Encrypted File Systems.

cpio -o Command

The **cpio -o** command reads file path names from standard input and copies these files to standard output, along with path names and status information. Avoid giving the **cpio** command path names made up of many uniquely linked files, as it may not have enough memory to keep track of them and would lose linking information.

cpio -i Command

The **cpio -i** command reads from standard input an archive file created by the **cpio -o** command and copies from it the files with names that match the *Pattern* parameter. These files are copied into the current directory tree. You can list more than one *Pattern* parameter, using the file name notation described in the **ksh** command. Note that in this application the special characters * (asterisk), ? (question mark), and [...] (brackets and ellipses) match the / (slash) in path names, in addition to their use as described in the **ksh** command**.** The default for the *Pattern* parameter is an * (asterisk), selecting all files in the Input. In an expression such as [a-z], the minus sign means *through* according to the current collating sequence.

A collating sequence can define equivalence classes for use in character ranges.

cpio -p Command

The **cpio -p** command reads file path names from standard input and copies these files into the directory named by the *Directory* parameter. The specified directory must already exist. If these path names include directory names that do not already exist, you must use the **d** flag to cause the specified directory to be created.

Note: You can copy special files only if you have root user authority.

cpio -U command

For AIX 5.3, the **cpio** command will ignore extended attributes by default. The **-U** option informs **cpio** to archive or restore attributes, which includes ACLs.

A new record type is required for extended attribute entries in **cpio** archive files. A new record type is also required for ACL entries in **cpio** archive files.

Each object in the **cpio** archive contains a **cpio** header followed by the data for the specified object.

The following table describes the **cpio** header:

Name of field	Size (number of bytes)	Use
h_magic	2	Magic number for identifying header.
h_dev	2	Device that contains a directory entry for this file.
h_ino	2	Inode number that identifies the input file to the file system.
h_mode	2	Mode of the input file, as defined in the mode.h file. The POSIX standard has 0130000, 0150000 - 0170000 available for file types that are not to be transported to other systems.
h_uid	2	User ID of the owner of the input file.
h_gid	2	Group ID of the owner of the input file.
h_nlink	2	Number of links that are connected to the input file.
h_rdev	2	ID of the remote device from which the input file is taken.
h_mtime	4	Time when data was last modified.
h_namesize	2	Length of the pathname including the NULL.
h_filesize	4	Length of the file in bytes.

Name of field	Size (number of bytes)	Use
h_name	PATH_MAX	Null-terminated pathname.

Each file which has an ACL will have a <header,data> object immediately preceding the object itself which describes the ACL as follows:

Header for ACL
The h_mode field set to 0130000 indicates the header describes an ACL. Additionally, the h_mode bits are set to indicate who can write the ACL. All other fields in the cpio header are set as for the inode of the file owning the ACL.

Data The data will be the ACL itself. The first 64-bits of the data will be the ACL type. It will be immediately followed by the ACL value.

Each extended attribute will have a single <header,data> object in the archive which completely describes the extended attribute as follows:

Header for EA
The h_mode field set to 0150000 indicates an extended attribute header. All fields in the cpio header are set as for the inode of the extended attribute. Except the h_name field is set to <NULL><*EAName*><NULL>

Data: This is formatted to describe the owner of the extended attribute as well as the data for the extended attribute. There is a eaHeader followed by the pathname of the owner of the extended attribute, followed by the extended attribute data.

```
struct eaHeader {
 char pathLen[12];
 char dataLen[12];
};
```

Parameters

Directory	Specifies the directory.
<*FileName*	Specifies a list of file names for the **cpio** command to use as input.
>*Output*	Specifies the output device such as a diskette or file. For more information on using tape devices see the **rmt** special file.
<*Input*	Specifies the input device (where *Input* is the *Output* file created by the **cpio -o** command). For more information on using tape devices, see the **rmt** special file.
Pattern	Specifies the pattern (as described in the **ksh** command) to be used with the command. The default for the *Pattern*parameter is an * (asterisk), selecting all the files in the *Input*.

Flags

All flags must be listed together, without any blanks between them. Not all of the following flags can be used with each of the **-o**, **-i**, and **-p** flags.

a Resets the access times of the source files to their previous times.

b Swaps both bytes and halfwords.

> **Note:** If there is an odd number of bytes or halfwords in the file being processed, data can be lost.

B Performs block input and output using 512 bytes to a record.

> **Note:** When using the **B** or **C** options to extract or create a tape archive, the blocking factor must be a multiple of the physical block size for that tape device.
>
> When using the **B** or **C** options to extract an archive from tape, the blocking factor should not be larger than the size of the archive as it exists on the tape.
>
> The **B** flag and the **C** flag are mutually exclusive. If you list both, the **cpio** command uses the last one it encounters in the flag list.

c Reads and writes header information in ASCII character form. If a **cpio** archive was created using the **c** flag, it must be extracted with **c** flag.

C *Value* Performs block input and output using the *Value* parameter times 512 bytes to a record. For instance, a **-C2** flag changes the block input and output sizes to 1024 bytes to a record.

d Creates directories as needed.

-E The **-E** option requires one of the following arguments. If you omit the **-E** option, **warn** is the default **cpio** behavior.

- **force** Fails the extract or copy operation on a file if the file's extent attributes 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. This is the default behavior.

f Copies all files except those matching the *Pattern* parameter.

l Links files rather than copying them, whenever possible. This flag can only be used with the **cpio -p** command.

m Retains previous file modification time. This flag does not work when copying directories.

M Retains previous file modification time even when directories are copied.

r Renames files interactively. If you do not want to change the file name, enter a single period or press the <Enter> key. In the latter case, the **cpio** command does not copy the file.

s Swaps bytes. This flag is used only with the **cpio -i** command.

> **Note:** If there is an odd number of bytes in the file being processed, data can be lost.

S Swaps halfwords. This flag is usable only with the **cpio -i** command.

> **Note:** If there is an odd number of halfwords in the file being processed, data can be lost.

t Creates a table of contents. This operation does not copy any files.

-U 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 *Target* filesystem then it is converted to the ACL type supported by the *Target* filesystem. If the EA is not supported on the filesystem then it is not copied.

u Copies unconditionally. An older file now replaces a newer file with the same name.

v Lists file names. If you use this with the **t** flag, the output looks similar to that of the **ls -l** command.

6 Processes an old file (for example, one written in UNIX Sixth Edition format). This flag is usable only with the **cpio -i** command.

-Z Archives the Encrypted File System (EFS) information of encrypted files or directories. The EFS information is extracted. When you specify the **-t** and **-v** flags along with the **-Z** flag, an **e** indicator is displayed after the file mode for encrypted files and directories that were archived with the **-Z** flag, and a hyphen (-) is displayed for other files.
Note: Archives created with the **-Z** flag can be restored only on AIX 6.1 or later releases.

Exit Status

This command returns the following exit values:

0 Successful completion.

>0 An error occurred.

Security

RBAC Disclaimer

Examples

1. To copy files onto diskette, enter:

   ```
   cpio -ov <filenames >/dev/rfd0
   ```

   This copies the files with path names listed in the `filenames` file in a compact form onto the diskette (`>/dev/rfd0`). The **v** flag causes the **cpio** command to display the name of each file as it is copied. This command is useful for making backup copies of files. The diskette must already be formatted, but it must not contain a file system or be mounted.

   > **Note:** Files with uid's and gid's greater than 65535 cannot be archived using the **cpio** command. In such instances, the user should use backup and restore.

2. To copy files in the current directory onto diskette, enter:

   ```
   ls *.c | cpio -ov >/dev/rfd0
   ```

   This copies all the files in the current directory whose names end with `.c`

3. To copy the current directory and all subdirectories onto diskette, enter:

   ```
   find . -print | cpio -ov >/dev/rfd0
   ```

   This saves the directory tree that starts with the current directory (`.`) and includes all of its subdirectories and files. Do this faster by entering:

   ```
   find . -cpio /dev/rfd0 -print
   ```

   The `-print` entry displays the name of each file as it is copied.

4. To list the files that have been saved onto a diskette with the **cpio** command, enter:

   ```
   cpio -itv </dev/rfd0
   ```

   This displays the table of contents of the data previously saved onto the `/dev/rfd0` file in the **cpio** command format. The listing is similar to the long directory listing produced by the **ls -l** command. To list only the file path names, use only the **-it** flags.

5. To copy the files previously saved with the **cpio** command from a diskette, enter:

   ```
   cpio -idmv </dev/rfd0
   ```

   This copies the files previously saved onto the `/dev/rfd0` file by the **cpio** command back into the file system (specify the **-i** flag). The **d** flag allows the **cpio** command to create the appropriate directories if a directory tree is saved. The **m** flag maintains the last modification time in effect when the files are saved. The **v** flag causes the **cpio** command to display the name of each file as it is copied.

6. To copy selected files from diskette, enter:

   ```
   cpio -i "*.c" "*.o" </dev/rfd0
   ```

   This copies the files that end with `.c` or `.o` from diskette. Note that the patterns `"*.c"` and `"*.o"` must be enclosed in quotation marks to prevent the shell from treating the * (asterisk) as a pattern-matching character. This is a special case in which the **cpio** command itself decodes the pattern-matching characters.

7. To rename files as they are copied from diskette, enter:

   ```
   cpio -ir </dev/rfd0
   ```

   The **-r** flag causes the **cpio** command to ask you whether to rename each file before copying it from diskette. For example, the message:

```
Rename <prog.c>
```

asks whether to give the file saved as `prog.c` a new name as it is copied. To rename the file, type the new name and press the Enter key. To keep the same name, you must enter the name again. To avoid copying the file at all, press the Enter key.

8. To copy a directory and all of its subdirectories, enter:

```
mkdir /home/jim/newdir
find . -print | cpio -pdl /home/jim/newdir
```

This duplicates the current directory tree, including the current directory and all of its subdirectories and files. The duplicate is placed in the new `/home/jim/newdir` directory. The **l** flag causes the **cpio** command to link files instead of copying them, when possible.

> **Note:** The performance of **cpio** to the 9348 Magnetic Tape Unit Model 12 can be improved by changing the default block size. To change the block size, enter the following at the command line:
>
> ```
> chdev -l <device_name> -a block_size=32k
> ```

9. To copy files in the current directory onto diskette and preserve the ACL and EA associated with the files, enter:

```
ls *.c | cpio -oUv >/dev/rfd0
```

Files

/usr/bin/cpio	Contains the **cpio** command.

Related Information

The **find** command, **ksh** command, **ln** command, **ls** command.

The **cpio** file format, **rmt** special file.

The Backup methods in *Operating system and device management* provides information on different methods of backing up, restoring process, different types of backup media, and guidelines for backup policies.

The Directories in *Operating system and device management* explains working with directories and path names.

The File systems in *Operating system and device management* explains file system types, management, structure, and maintenance.

The Files in *Operating system and device management* provides information on working with files.

System V cpio Command

Purpose

Copies files into and out of archive storage and directories.

Syntax

cpio -i [**-b**] [**-B**] [**-c**] [**-d**] [**-f**] [**-k**] [**-m**] [**-r**] [**-s**] [**-S**] [**-T**] [**-t**] [**-u**] [**-v**] [**-V**] [**–6**] [**-C** *bufsize*] [**-E** *file*] [**-H** *hdr*] [**-I** *file* [**-M** *message*]] [**-R** *ID*]] [*Patterns* ...]

cpio -o [**-a**] [**-A**] [**-B**] [**-c**] [**-L**] [**-v**] [**-V**] [**-C** *bufsize*] [**-H** *hdr*] [**-K** *mediasize*] [**-O** *file* [**-M** *message*]]

cpio -p [**-a**] [**-d**] [**-I**] [**-L**] [**-m**] [**-u**] [**-v**] [**-V**] [**-R** *ID*] *Directory*

Description

The **cpio** command copies files into and out of an archive. The **-i**, **-o** and **-p** options select the action to be performed. The following list describes each of the actions. The **-o**, **-p** and **-i** options are mutually exclusive.

cpio -i (copy in)

cpio -i (copy in) extracts files from the standard input (only if **-I** is not specified), which is assumed to be the product of a previous **cpio -o**. Only files with names that match *Patterns* are selected. *Patterns* are regular expressions given in the filename generating notation of **ksh**. In *Patterns*, meta-characters ″?″, ″*″, and ″[. . .]″ match the slash (″/″) character, and backslash (″\″) is an escape character. A ″!″ meta-character means not. (For example, the ″[!abc]″ pattern would exclude all files that begin with either a, b or c.) Multiple patterns may be specified and if no patterns are specified, the default for *Patterns* is ″*″ (that is, select all files). Each pattern must be enclosed in double quotes; otherwise, the name of a file in the current directory might be used. Extracted files are conditionally created and copied into the current directory tree based on the options described below.

The **cpio -i** command reads the standard input of an archive file created that was using the **cpio -o** command, and copies the files with names that match the *pattern* parameter. The *pattern* parameter is a regular expression given with general notation of **ksh**. These files are copied into the current directory tree. More than one pattern parameter can be used, using the file name notation described in the **ksh** command. The patterns can be special characters * (asterisk), ? (question mark), and [...] (brackets and ellipses). The default for the pattern parameter is an * (asterisk), selecting all files in the input. In an expression such as [a-z], the minus sign means `through` according to the current collating sequence.

The permissions of the files will be those of the previous **cpio -o**. Owner and group permissions will be the same as the current user unless the current user is the root user. If this is true, owner and group permissions will be the same as those resulting from the previous **cpio -o**. Blocks are reported in 512-byte quantities.

If **cpio -i** tries to create a file that already exists and the existing file is the same age or younger (newer), **cpio** will output a warning message and not replace the file. On the other hand if the file being extracted is older than the one in the cpio archive then the existing file will be replaced without any warning from the command.

cpio -o (copy out)

cpio -o reads the standard input to obtain a list of path names and copies those files onto the standard output together with path name and status information.

cpio -p (copy pass)

cpio -p reads the standard input to obtain a list of path names of files and copies these files into the directory named by the *Directory* parameter. The specified directory must already exist. If these path names include directory names that do not already exist, you must use the d flag to cause the specified directory to be created. By default the Access Control List's (ACL) are transferred [copied] from source file to destination file with this option only.

Flags

-a	Resets the access time of the source files to their previous times.
-A	Appends files to an archive. The **-A** option requires the **-O** option. The append option **-A** is not valid for the **rmt** special file and diskettes.
-b	Reverse the order of the bytes within each word. This option is valid only with the **-i** option.
-B	The default buffer size is 512 bytes when neither this nor the **-C** option is used. But when **-B** flag is used the buffer size is set to 5120 bytes block for the Input/Output operations.

-c Read or write header information in ASCII character form for system interoperability and portability. The **-c** option is mutually exclusive with **-H** and **-6**. Either the **-c** or **-H** option can be used when the target and destination machines are different types.

-C *bufsize* The block size for Input/Output operation is set to *bufsize*, where *bufsize* indicates the buffer size in positive integer. If used with **-K**, *bufsize* must be a multiple of 1K.

-d Creates directories as needed.

-E *file* Specify an input file (*file*) that contains a list of file names to be extracted from the archive with one file name per line.

-f Copy in all files except those in *Pattern* parameter.

-H *hdr* Read or write header information in hdr format. Either the **-h** or **-c** option can be used when the target and the destination machines are different types. This option is mutually exclusive with the **-c** and **-6** options. This format allows system interoperability and portability. The **cpio** utility supports the archival of files larger than 2 GB in size when using the ASCII (**-c**), CRC (**-Hcrc**), tar (**-Htar**), or ustar (**- Hustar**) formats. Valid values for *hdr* are:

crc Same as **CRC**. ASCII header with an additional per-file checksum. The **crc** file format will handle files larger than 2 GB.

ustar Same as **USTAR**. IEEE/P1003 Data Interchange Standard header and format.

tar Same as **TAR**. Tar header and format. The **tar** format is provided for compatibility with the **tar** program.

odc ASCII header with small fundamental types.

-I *file* Read the contents of *file* as an input archive. If *file* is a character special device, and the current medium has been completely read, replace the medium and press the Enter key to continue to the next medium. This option is valid only with the **-i** option.

-k Attempt to skip corrupted file headers and I/O errors that may be encountered. This option lets the user read only those files with good headers if files from a medium that is corrupted. This option is valid only with the **-i** option.

-K *mediasize* Specify the media size as a multiple of 1K. If used with **-C** *bufsize*, then *bufsize* must be a multiple of 1K.

-l Hard links files rather than copying them, whenever possible. If a file cannot be linked, then it will be copied. This option is valid only with **-p** option.

-L This option assists in copying the files rather than linking. The content of the link file is copied with the links name. Without **-L** or **-l** option, the symbolic links will be maintained as is default with **-p**.

-m Retain previous file modification time. The modification time and access time of a restored file is set to the modification time of the file when it was backed up. Modification time of directories is not retained.

-M *message* Define a message to use when switching media. When the **-O** or **-I** options are given cpio on a special device, this option can be used to define the message that is printed when you reach the end of the medium. A **%d** can be placed in message to print the sequence number of the next medium needed to continue.

-O *file* Direct the output of **cpio** to *file*. If *file* is a special device and the current medium is full, replace the medium and type Enter to continue to the next medium. This option is valid only with the **-o** option.

-r Renames files interactively. To skip a file, type Enter. To retain the original path name, type . (period). This option is valid only with the **-i** option.

-R *ID* Reassigns ownership and group information for each file to a valid user *ID*. This option is valid only for the root user.

-s Swap bytes within each half word.
Note: The **-s** and the **-S** flags are basically for byte sequencing.

-S Swap half words within each word.
Note: The **-s** and the **-S** flags are basically for byte sequencing.

-t Creates a table of contents. This operation does not create any files. The **-t** flag and the **-V** flag are mutually exclusive.

-T Truncates long file names to 14 characters. This option is valid only with the **-i** option.

-u Copies unconditionally (normally, an older file will not replace a newer file with the same name).

-v This is the verbose option that causes a list of file names to be printed. When used with the **-t** option, the table of contents looks like the output of an **ls -l** command.

-V This is a special verbose option that allows to print a dot for each file read or written. Useful to assure the user that cpio is working without printing out all file names. Note that the **-V** and **-v** options are mutually exclusive and whichever occurs earlier in the command line will be processed accordingly ignoring the other.

-6 Processes a UNIX System Sixth Edition archive format file. This option is mutually exclusive with the **-c** and **-H** options.

Parameters

Directory Specifies the directory.

Patterns Specifies one or more patterns (as described in the **ksh** command) to be used with the command. The default for the *Patterns* parameter is an * (asterisk), selecting all the files in the input.

Exit Status

0 The command completed successfully.

>0 An error occurred.

Examples

1. To copy all the files in the current directory onto tape device **/dev/rmt0**, enter:
   ```
   find . | /usr/sysv/bin/cpio -oc >/dev/rmt0
   ```
   The **-c** option ensures that the file is made portable to other machines. Instead of **find** you can also use **ls**, **cat**, **echo** and so on to pipe a list of names to **cpio**. The output could also be redirected to a regular **cpio** file instead of a device.
2. To extract an **cpio** archive file named ″arfile″ created by **cpio** command use the following:
   ```
   /usr/sysv/bin/cpio -icdI arfile
   ```
   Here all the files are extracted from the **cpio** archive and the **-d** option ensures that the required directory paths are created as when required.
3. A **cpio** archive file can also be extracted as follows:
   ```
   /usr/sysv/bin/cpio -icd < arfile
   ```
   The **-d** option ensures that all the required directories are created under the current directory. The standard input can be used only if **-I** flag is not specified.
4. To extract unconditionally all the files in ″arfile″ use the following:
   ```
   /usr/sysv/bin/cpio -icduI arfile
   ```
5. To skip any files which corrupted headers, **cpio** can be used as follows:
   ```
   /usr/sysv/bin/cpio -ickudI arfile
   ```
6. If the access time of the files archived needs to be reset when cpio is used to create an archive, use **cpio** in the following way:
   ```
   ls | /usr/sysv/bin/cpio -oca > arfile
   ```
7. To extract only the files matching the pattern ″a*″ from the archive ″ar″, use the following:
   ```
   cat ar | /usr/sysv/bin/cpio -ickud "a*"
   ```
   This command extracts all the files starting with letter ″a″.
8. To display the list of files archived, use **cpio** in the following way:
   ```
   cat ar | /usr/sysv/bin/cpio -itv
   ```

The verbose option (**-v**) ensures that the list given by **-t** option is listed in a very similar way as **ls -l** command.

9. The **cpio -p** command can be used to copy a directory tree to a new path, as follows:
   ```
   find . -print | /usr/sysv/bin/cpio -pd /home/user1/newdir
   ```
   The entire directory tree from current directory is copied to **/home/user1/newdir**. The **-d** option ensures that directories are created as necessary.
10. To retain the modification time and access control list while copying the directory tree, use the **cpio** command as follows:
    ```
    find . -name "*.o" -print | /usr/sysv/bin/cpio -pdlmv /home/user1/newdir
    ```
    In this example only the **.o** files under the directory tree are copied to **/home/user1/newdir**.
11. To append a list of files to a **cpio** archive matching a particular pattern, invoke a command similar to the following:
    ```
    ls d* | /usr/sysv/bin/cpio -oA0 /tmp/ar
    ```
    In this example, all files starting with ″d″ in the current directory will be appended to the **cpio** archive.
12. To extract only a list of files listed inside a regular file from an **cpio** archive, use the following command:
    ```
    cat ar | /usr/sysv/bin/cpio -i -E Efile
    ```
    In this example, **cpio** extracts only those files that are listed in the regular file ″Efile″, provided the specified file name exists in the archive.
13. To hard link all the files instead of copying them, invoke a command similar to the following:
    ```
    ls d* | /usr/sysv/bin/cpio -pdl /home/user2/newdir
    ```
    In this example, the **-l** flag ensures all the file names starting with the character ″d″ are hard linked to the **/home/user2/newdir**, the directory specified. Hard linking across file systems is not allowed, thus the **-l** option cannot be used when the destination directory is in any other filesystem.

Files

/usr/sysv/bin/cpio	Contains the System V **cpio** command.

Related Information

The **/usr/bin/cpio** command, **tar** command, **dd** command, **pax** command.

---

cplv Command

Purpose

Copies the contents of a logical volume to a new logical volume.

Syntax

To Copy to a New Logical Volume

cplv [**-v** *VolumeGroup*] [**-y** *NewLogicalVolume* | -Y *Prefix*] *SourceLogicalVolume*

To Copy to an Existing Logical Volume

cplv **-e** *DestinationLogicalVolume* [**-f**] *SourceLogicalVolume*

Description

Attention: Do not copy from a larger logical volume containing data to a smaller one. Doing so results in a corrupted file system because some data (including the superblock) is not copied. This command will fail if the **cplv** creates a new logical volume and the volume group is varied on in concurrent mode.

The **cplv** command **copies** the contents of *SourceLogicalVolume* to a new or existing *DestinationLogicalVolume*. The *SourceLogicalVolume* parameter can be a logical volume name or a logical volume ID. The **cplv** command creates a new logical volume with a system-generated name by using the default syntax. The system-generated name is displayed.

Notes:

1. If you are copying a striped logical volume and the destination logical volume does not exist, an identical copy, including the striped block size and striping width of the source logical volume is created and then the data is copied.
2. If you are copying a striped logical volume and you have created the destination logical volume, with the **mklv** command using a different stripe block size and striping width, or the destination is not a striped logical volume, the new characteristics are maintained, and the data is copied from the source logical volume.
3. To use this command, you must either have root user authority or be a member of the **system** group.
4. The **cplv** command is not allowed on a snapshot volume group.
5. If the *SourceLogicalVolume* is a jfs or jfs2 type, the file system must be successfully unmounted and **fsck** must be run successfully on the newly created file system before the **cplv** command can be run. If you run the **fsck** command before mounting the new file system, errors are returned because the log device contained in the superblock would still refer to the original file system. Mount the file system before running **fsck** so that a new log device is created.

You can use the Volumes application in Web-based System Manager to change volume characteristics. You could also use the System Management Interface Tool (SMIT) **smit cplv** fast path to run this command.

Flags

-e	Specifies that the *DestinationLogicalVolume* exists and that a new logical volume should not be created. If the *DestinationLogicalVolume* is smaller than the *SourceLogicalVolume*, the extra logical partitions are not copied. When you use this flag, any data already in the *DestinationLogicalVolume* is destroyed. For this reason, user confirmation is required, unless the **-f** flag is added. The *Type* characteristic of the *DestinationLogicalVolume* must be **copy** to prevent inadvertently overwriting data. To change the *Type* characteristic, use the **chlv** command.
-f	Copies to an existing logical volume without requesting user confirmation.
-v *VolumeGroup*	Specifies the volume group where the new logical volume resides. If this is not specified, the new logical volume resides in the same volume group as the *SourceLogicalVolume*.
-y *NewLogicalVolume*	Specifies the name to use, in place of a system-generated name, for the new logical volume. Logical volume names must be unique systemwide names, and can range from 1 to 15 characters.
-Y *Prefix*	Specifies a prefix to use in building a system-generated name for the new logical volume. The prefix must be less than or equal to 13 characters. A name cannot begin with a prefix already defined in the PdDv class in the Device Configuration Database for other devices, or a name already used by another device.

Security

RBAC Disclaimer

Examples

1. To copy the contents of logical volume `fslv03` to a new logical volume, type:

   ```
   cplv fslv03
   ```

   The new logical volume is created, placed in the same volume group as `fslv03`, and named by the system.
2. To copy the contents of logical volume `fslv03` to a new logical volume in volume group `vg02`, type:

   ```
   cplv  -v vg02
   fslv03
   ```

   The new logical volume is created, named, and added to volume group `vg02`.
3. To copy the contents of logical volume `lv02` to a smaller, existing logical volume, `lvtest`, without requiring user confirmation, type:

   ```
   cplv  -e lvtest  -f lv02
   ```

Files

/usr/sbin Directory where the **cplv** command resides.

Related Information

The **chlv** command, **migratepv** command, **mklv** command.

The Logical volume storage in *Operating system and device management* explains the Logical Volume Manager, physical volumes, logical volumes, volume groups, organization, ensuring data integrity, and understanding the allocation characteristics.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

The System Management Interface Tool (SMIT) in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* explains the structure, main menus, and tasks that are done with SMIT.

---

cpp Command

Purpose

Performs file inclusion and macro substitution on C language source files.

Syntax

/usr/ccs/lib/cpp [**-C**] [**-P**] [**-q***DBCS*] [**-I***Directory*] [**-U***Name*] [**-D***Name* [*=Defin ition*]] [**-qlanglvl=***Language*] [*InFile*] [*OutFile*]

Description

The **cpp** command performs file inclusion and macro substitution on C language source files. It reads *InFile* and writes to *OutFile* (standard input and standard output by default).

The **cpp** command is designed to conform to the preprocessing directives and instructions for the C language as defined by the document ″Draft American National Standard for Information Systems - Programming Language C″ (X3J11/88-159).

The **cpp** program recognizes the following special names:

__LINE__	The current line number.
__DATE__	The date of translation of the source file.
__TIME__	The time of translation of the source file.
__STDC__	Indicates a conforming implementation.
__FILE__	The current file name.
__STR__	Indicates the compiler will generate inline code for certain string functions (as defined in **/usr/include/string.h**).
__MATH__	Indicates the compiler will generate inline code for certain math functions (as defined in **/usr/include/math.h**).
__ANSI__	Indicates **langlvl** is set equal to ANSI.
__SAA__	Indicates **langlvl** is set equal to SAA.
__SAA_L2__	Indicates **langlvl** is set equal to SAAL2.
__EXTENDED__	Indicates **langlvl** is set equal to extended.
__TIMESTAMP__	Indicates the date and time when the source file was last modified.

All **cpp** directive lines must begin with a **#** (pound sign). These directives are:

#define *Name TokenString*	Replaces subsequent instances of *Name* with *TokenString*.
#define *Name*(*Argument,...,Argument*) *TokenString*	Replaces subsequent instances of the sequence *Name* (*Argument*, . . . ,*Argument*) with *TokenString*, where each occurrence of an *Argument* in *TokenString* is replaced by the corresponding token in the comma-separated list. Note that there must not be any space between *Name* and the left parenthesis.
#undef *Name*	Ignores the definition of *Name* from this point on.
#include *"File"* or **#include <***File***>**	Includes at this point the contents of *File*, which **cpp** then processes. If you enclose *File* in *" "* (double quotation marks) the **cpp** command searches first in the directory of *InFile*, second in directories named with the **-I** flag, and last in directories on a standard list. If you use the <*File*> notation, the **cpp** command searches for *File* only in the standard directories. It does not search the directory in which *InFile* resides.
#line *Number* [*"File"*]	Causes the implementation to behave as if the following sequence of source lines begins with a source line that has a line number as specified by *Number*. If *File* is supplied, the presumed name of the file is changed to be *File*.
#error *TokenString*	Produces a diagnostic message that includes *TokenString*.
#pragma *TokenString*	An implementation-defined instruction to the compiler.
#endif	Ends a section of lines begun by a test directive (**#if**, **#ifdef**, or **#ifndef**). Each test directive must have a matching **#endif**.

#ifdef *Name*

Places the subsequent lines in the output only if:

Name has been defined by a previous **#define**

OR

Name has been defined by the **-D** flag,

OR

Name is a special name recognized by the **cpp** command,

AND

Name has not been undefined by an intervening **#undef**,

OR

Name has not been undefined with the **-U** flag.

#ifndef *Name*

Places the subsequent lines in the output only if:

Name has never been defined by a previous **#define**,

AND

Name is not a special name recognized by the **cpp** command,

OR

Name has been defined by a previous **#define** but it has been undefined by an intervening **#undef**,

OR

Name is a special name recognized by the **cpp** command, but it has been undefined with the **-U** flag.

#if *Expression*

Places subsequent lines in the output only if *Expression* evaluates to nonzero. All the binary nonassignment C operators, the ?: operator, and the unary -, !, and - operators are legal in *Expression*. The precedence of the operators is the same as that defined in the C Language. There is also a unary operator **defined**, which can be used in *Expression* in these two forms:

defined (*Name***) or defined** *Name*
: This allows the utility of **#ifdef** and **#ifndef** in a **#if** directive. Only these operators, integer constants, and names that are known by **cpp** should be used in *Expression*. The **sizeof** operator is not available.

#elif *Expression*

Places subsequent lines in the output only if the expression in the preceding **#if** or **#elif** directive evaluates to false or is undefined, and this *Expression* evaluates to true.

#else

Places subsequent lines in the output only if the expression in the preceding **#if** or **#elif** directive evaluates to false or is undefined (and hence the lines following the **#if** and preceding the **#else** have been ignored).

Each test directive's condition is checked in order. If it evaluates to false (0), the group that it controls is skipped. Directives are processed only through the name that determines the directive in order to keep track of the level of nested conditionals; the rest of the directives' preprocessing tokens are ignored, as are the other preprocessing tokens in the group. Only the first group whose control condition evaluates to true (nonzero) is processed. If none of the conditions evaluates to true, and there is a **#else** directive, the group controlled by the **#else** is processed; lacking a **#else** directive, all the groups until the **#endif** are skipped.

Flags

-C
Copies C language comments from the source file to the output file. If you omit this flag, the **cpp** command removes all C language comments except those found on a **cpp** directive line.

-D*Name*[**=***Definition*]
Defines *Name* as in a **#define** directive. The default *Definition* is **1**.

-I*Directory*
Looks first in *Directory*, then looks in the directories on the standard list for **#include** files with names that do not begin with a / (slash). See the previous discussion of **#include**.

-P
Preprocesses input without producing line control information for the next pass of the C compiler.

-q*DBCS*
Specifies double-byte character set mode.

-U*Name*
Removes any initial definition of *Name*, where *Name* is a symbol predefined by the preprocessor (except for the four preprocessor mode indicators: __**ANSI**__, __**EXTENDED**__, __**SAA**__, and __**SAA_L2**__). This flag is not recognized in ANSI mode.

-qlanglvl=*Language*
Selects a language level for processing. *Language* can be ANSI, SAA, SAAL2, or extended. The default is extended.

> **Note:** When *Language* is extended, _NO_PROTO is not automatically defined. Such definition can be done using the **-D** option in the **/etc/xlc.cfg** file.

Examples

1. To display the text that the preprocessor sends to the C compiler, enter:

   ```
   /usr/ccs/lib/cpp pgm.c
   ```

   This preprocesses `pgm.c` and displays the resulting text at the work station. You may want to see the preprocessor output when looking for errors in your macro definitions.
2. To create a file containing more readable preprocessed text, enter:

   ```
   /usr/ccs/lib/cpp -P -C pgm.c pgm.i
   ```

   This preprocesses `pgm.c` and stores the result in `pgm.i`. It omits line numbering information intended for the C compiler (`-P`), and includes program comments (`-C`).
3. To predefine macro identifiers, enter:

```
/usr/ccs/lib/cpp -DBUFFERSIZE=512 -DDEBUG
pgm.c
pgm.i
```

This defines BUFFERSIZE with the value 512 and DEBUG with the value 1 before preprocessing.

4. To use **#include** files located in nonstandard directories, enter:

```
/usr/ccs/lib/cpp -I/home/jim/include
pgm.c
```

This looks in the current directory for quoted **#include** files, then in /home/jim/include, and then in the standard directories. It looks in /home/jim/include for angle-bracketed **#include** files (< >) and then in the standard directories.

5. To preprocess with the ANSI definition, enter:

```
/usr/ccs/lib/cpp -qlanglvl=ansi pgm.c
```

Files

/usr/include	Standard directory for **#include** files.

Related Information

The **m4** command.

---

cpupstat Command

Purpose

Detects configurations that could cause a CPU DR operation to fail.

Syntax

cpupstat [**-v**] **-i** *identifier*

Description

The purpose of this command is to detect configurations that could cause a CPU DR operation to fail. There are multiple steps to the command.

1. Parse and validate the input.
2. Check all the WLM class control block rsets for rsets with a single active CPU matching the passed in CPU. Class control block rsets are located in ccb[cid]->cl_rset, to iterate through all of them the value of CID must be incremented and class validity checked for each possible value. A count of the number of classes with such an rset will printed. If the verbose option is given, the names of the classes will be printed as well.
3. Check all the kernel registry rsets for rsets with a single active cpu matching the passed in CPU. A count of the number of processes with attachments to such rsets will be printed to the user. If the verbose option is given, the process IDs will be printed as well.
4. A count of **bindprocessor** attachments for the highest numbered bind ID will be printed for the user. If the verbose option is given, the process IDs will be printed as well.

Flags

-i	The index of the logical CPU ID.
-v	Verbose option.

Exit Status

If an error is encountered in the execution a suitable error message is written to stderr, and the command exits with a non-zero exit status.

Examples

1.
```
# cpupstat -i 2

3 WLM classes have single CPU rsets with CPU ID 2.
0 processes have single CPU rset attachments with CPU ID 2.
0 processes are bound to bind ID 2.
```
2.
```
# cpupstat -v -i 2

3 WLM classes have single CPU rsets with CPU ID 2.
        c1
        c1.Default
        c1.Shared
0 processes have single CPU rset attachments with CPU ID 2.
0 processes are bound to bind ID 2.
```
3.
```
# cpupstat -i 2

0 WLM classes have single CPU rsets with CPU ID 2.
2 processes have single CPU rset attachments with CPU ID 2.
0 processes are bound to bind ID 2.
```
4.
```
# cpupstat -v -i 2

0 WLM classes have single CPU rsets with CPU ID 2.
2 processes have single CPU rset attachments with CPU ID 2.
        16600
        26444
0 processes are bound to bind ID 2.
```

   For bound processes, the last list, the output is the same as for rset attachments, where the PID gets printed if the **-v** option is specified.

Location

/usr/bin/cpupstat

Related Information

"bindprocessor Command" on page 256.

Dynamic Logical Partitioning.

---

craps Command

Purpose

Starts the craps game.

Syntax

craps

Description

The **craps** command starts the craps game similar to ones played in Las Vegas. The **craps** command simulates the roller while you place bets. You can bet with the roller by making a positive bet or you can bet with the house by making a negative bet.

You begin the game with a two thousand dollar bankroll. When the program prompts with `bet?`, you can bet all or part of your bankroll. You can not bet more than your current bankroll. The roller throws the dice. The payoff odds are one-to-one.

On the first roll, 7 or 11 wins for the roller; 2, 3, or 12 wins for the house; and any other number becomes the point and you roll again. On subsequent rolls, the point wins for the roller; 7 wins for the house; and any other number rolls again. For example:

```
Your bankroll is $2000
bet? 100
5       3
The point is 8
        6       6
        4       1
        2       1
        2       5
You lose your bet of $100
Your bankroll is $1900
```

In this example, the player has a bankroll of two thousand dollars and bets one hundred dollars. The first roll was 8. This became the point because neither you nor the house wins on a first roll of 8. Subsequent rolls were: 12, 5, 3, and 7. The house wins on a roll of 7 when the roller is trying to match the point. The player lost the bet of one hundred dollars. After displaying the new bankroll, the game will prompt `bet?` and the game will continue.

If you lose your bankroll, the game prompts with `marker?`, offering to lend you an additional two thousand dollars. Accept the loan by responding `Y` (yes). Any other response ends the game.

When you hold markers, the house reminds you before a bet how many markers are outstanding. When you have markers and your bankroll exceeds two thousand dollars, the game asks `Repay marker?`. If you want to repay part or all of your loan, enter `Y` (yes). If you have more than one marker, the **craps** command prompts `How many?` If you respond with a number greater than the number of markers you hold, it repeats the prompt until you enter a valid number. If you accumulate 10 markers (a total loan of twenty thousand dollars), the game tells you so and exits. If you accumulate a bankroll of more than fifty thousand dollars while holding markers, the money owed is repaid automatically.

A bankroll of more than one hundred thousand dollars breaks the bank, and the game prompts `New game?` To quit the game, press the Interrupt (Ctrl-C) or End Of File (Ctrl-D) key sequence; the game indicates whether you won, lost, or broke even, and exits.

Files

/usr/games Location of the system's games.

Related Information

The **arithmetic** command, **back** command, **bj** command, **fish** command, **fortune** command, **hangman** command, **moo** command, **number** command, **quiz** command, **ttt** command, **turnoff** command, **turnon** command, **wump** command.

---

createvsd Command

Purpose

createvsd – Creates a set of virtual shared disks, with their associated logical volumes.

Syntax

createvsd **–n** {*node_list* | **ALL**} **–s** *size_in_MB* **–g** *vg_name*
[{**–c** *vsds_per_node* | **–L**}] [–A]
[{**–m** *mirror_count* | **–p** *lvm_strip_size_in_K*}] [**–v** *vsd_name_prefix*]
[**–l** *lv_name_prefix*] [**–T** *lp_size_in_MB*] [**–k** *vsd_type*] [**–x**]

Description

Use this command to create a volume group with the specified name (if one does not already exist) and to create a logical volume within that volume group. You specify the logical volume size using the **-s** flag.

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

```
smit vsd_data
```

and select the **Create a virtual shared disk** option.

Flags

Note: Some examples shown in this list do not contain enough flags to be executable. They are shown in an incomplete form to illustrate specific flags.

–n *node_list* Specifies the node numbers of the peer domain on which you are creating virtual shared disks. The backup node cannot be the same as the primary node. For nonconcurrent virtual shared disks, the format of the node list is:

```
[P/S] : disk_list1+disk_list2/
```

For concurrent virtual shared disks, the format of the node list is:

```
[S1/S2/...Sn] : disk_list1+disk_list2/
```

″P″ specifies the primary server node for serially accessed shared disks, ″S″ specifies the backup (secondary) server node for serially accessed shared disks, and S1 and S2 specifies the server nodes for concurrently accessed shared disks. *disk_list1* is the list of local physical disks, or vpaths, for the logical volume on the primary. In other words, this list can be made up of *hdiskx, hdisky,...* or *vpathx, vpathy,...*.

Notes:

1. Vpaths are available only if the ″Subsystem Device Driver″ is installed. Vpaths provide ″virtual paths″ to the same physical volume.
2. Hdisks and vpaths cannot both be specified in the same list.

disk_list1+disk_list2 is the list of local physical disks or vpaths in the volume group on the primary, if you want to have more disks in the volume group than are needed for the logical volume. The sequence in which nodes are listed determines the names given to the virtual shared disks. For example:

```
createvsd -n 1,6,4 -v PRE
```

(with the *vsd_prefix* PRE) creates virtual shared disks PRE1n1 on node 1, PRE2n6 on node 6, and PRE3n4 on node 4.

To create a volume group that spans hdisk2, hdisk3, and hdisk4 on node 1, with a backup on node 3, enter:

```
createvsd -n 1/3:hdisk2,hdisk3,hdisk4/ -v DATA
```

This command creates:

- Virtual shared disk DATA1n1 with logical volume lvDATA1n1 on a volume group with the global volume group name DATA1n1b3 on node 1, exported to node 3. The Logical Volume Manager (LVM) volume group name is DATA. The logical volumes span hdisk2, hdisk3, and hdisk4.

To create volume groups just like that one on nodes 1, 2, and 3 of a system with backup on nodes 4, 5, and 6 of the same system, enter:

```
createvsd -n 1/4:hdisk1,hdisk2,hdisk3/,2/5:hdisk5,hdisk6, \
          hdisk7/,3/6:hdisk2,hdisk4,hdisk6/ -v DATA
```

This command is shown on two lines here, but you must enter it without any spaces between the items in *node_list*.

The command creates:

- Virtual shared disk DATA1n1 with logical volume lvDATA1n1 on a volume group with the local volume group name DATA on node 1, exported to node 4. The global volume group name is DATAn1b4.
- Virtual shared disk DATA2n2 with logical volume lvDATA2n2 on a volume group with the local volume group name DATA on node 2, exported to node 5. The global volume group name is DATAn2b5.
- Virtual shared disk DATA3n3 with logical volume lvDATA3n3 on a volume group with the local volume group name DATA on node 3, exported to node 6. The global volume group name is DATAn3b6.

To create a virtual shared disk where the logical volume spans only two of the physical disks in the volume group, enter:

```
createvsd -n 1/3:hdisk1,hdisk2+hdisk3/ -v DATA
```

This command creates the virtual shared disk DATA1n1 with logical volume lvDATA1n1 spanning hdisk1 and hdisk2 in the volume group DATA, which includes hdisk1, hdisk2, and hdisk3. It exports the volume group DATA to node 3.

If a volume group is already created and the combined physical hdisk lists contain disks that are not needed for the logical volume, those hdisks are added to the volume group. If the volume group has not already been created, **createvsd** creates a volume group that spans *hdisk_list1+hdisk_list2*.

Backup nodes cannot use the same physical disk as the primary does to serve virtual shared disks.

ALL specifies that you are creating virtual shared disks on all nodes in the RSCT peer domain. No backup nodes are assigned if you use this operand. The virtual shared disks will be created on all the physical disks attached to the nodes in *node_list* (you cannot specify which physical disks to use.)

–s Specifies the size in megabytes of each virtual shared disk.

–g Specifies the Logical Volume Manager (LVM) volume group name. This name is concatenated with the node number to produce the global volume group name. For example:

```
createvsd -n 6 -g VSDVG
```

creates a volume group with the local volume group name VSDVG and the global volume group name VSDVG1n6 on node 6. The node number is added to the prefix to avoid name conflicts when a backup node takes over a volume group. If a backup node exists, the global volume group name will be concatenated with the backup node number as well as the primary. For example:

```
createvsd -n 6/3/ -g VSDVG
```

creates a volume group with the local volume group name VSDVG and the global volume group name VSDVGn6b3. The primary node is node 6 and the backup node for this volume group is node 3.

–c Specifies the number of virtual shared disks to be created on each node. If *number_of_vsds_per_node* is not specified, one virtual shared disk is created for each node specified on **createvsd**. If more than one virtual shared disk is to be created for each node, the names will be allocated alternately. For example:

```
createvsd -n 1,6 -c 2 -v DATA
```

creates virtual shared disks DATA1n1 on node 1, DATA2n6 on node 6, DATA3n1 on node 1, and DATA4n6 on node 6.

–L Allows you to create one virtual shared disk on each node without using sequential numbers, for locally accessed virtual shared disks.

–A Specifies that virtual shared disk names will be allocated to each node in turn, for example:

```
createvsd -n 1,6 -c 2 -A DATA
```

creates DATA1n1 and DATA2n1 on node 1, and DATA3n6 and DATA4n6 on node 6.

–m Specifies the LVM mirroring count. The mirroring count sets the number of physical partitions allocated to each logical partition. The range is from 1 to 3 and the default value is 1.

–p Specifies the LVM strip size (a strip size multiplied by the number of disks in an array equals the stripe size). If this flag is not specified, the logical volumes are not striped. To use striping, the node on which the virtual shared disks are defined must have more than one physical disk.

–v Specifies a prefix to be given to the names of the created virtual shared disks. This prefix will be concatenated with the virtual shared disk number, node number, and backup node number, if a backup disk is specified. For example, if the prefix PRE is given to a virtual shared disk created on node 1 and there are already two virtual shared disks with this prefix across the partition, the new virtual shared disk name will be PRE3n1. The name given to the underlying logical volume will be lvPRE3n1, unless the **–l** flag is used. The **createvsd** command continues to sequence virtual shared disk names from the last PRE-prefixed virtual shared disk.

If **-v** is not specified, the prefix `vsd` is used.

Note: The last character of the *vsd_name_prefix* cannot be a digit. Otherwise, the 11th virtual shared disk with the prefix PRE would have the same name as the first virtual shared disk with the prefix PRE1. Nor can the *vsd_name_prefix* contain the character '**.**' because '**.**' can be any character in regular expressions.

–l Overrides the prefix *lvx* that is given by default to a logical volume by the **createvsd** command, where *x* is the virtual shared disk name prefix specified by *vsd_name_prefix* or the default (`vsd`). For example:

```
createvsd -n 1 -v DATA
```

creates one virtual shared disk on node 1 named DATA1n1 with an underlying logical volume lvDATA1n1. If the command

```
createvsd -n 1 -v DATA -l new
```

is used, the virtual shared disk on node 1 is still named DATA1n1, but the underlying logical volume is named lvnew1n1.

It is usually more helpful **not** to specify **–l**, so that your lists of virtual shared disk names and logical volume names are easy to associate with each other and you avoid naming conflicts.

–T Specifies the size of the physical partition in the Logical Volume Manager (LVM) logical volume group and also the logical partition size (they will be the same) in megabytes. You must select a power of 2 in the range 2 – 256. The default is 4MB.

The Logical Volume Manager limits the number of physical partitions to 1016 per disk. If a disk is greater than 4 gigabytes in size, the physical partition size must be greater than 4MB to keep the number of partitions under the limit.

–k *vsd_type* Specifies the type of virtual shared disk. The options are:

- **VSD**: specifies a serial access, or nonconcurrent, shared disk, or
- **CVSD**: specifies a concurrent access shared disk.

The default is **VSD**.

–x Specifies that the steps required to synchronize the virtual shared disks on the primary and secondary nodes should **not** be performed; that is, the sequence:

- **varyoffvg** on the primary node
- **exportvg** on the secondary node
- **importvg** on the secondary node
- **chvg** on the secondary node
- **varyoffvg** on the secondary node
- **varyonvg** on the primary nodes

is not done as part of the **createvsd** processing. This speeds the operation of the command and avoids unnecessary processing in the case where several virtual shared disks are being created on the same primary/secondary nodes. In this case, however, you should either **not** specify **–x** on the last **createvsd** in the sequence or issue the volume group commands listed above explicitly.

Parameters

None.

Security

You must have root authority to run this command.

Exit Status

0 Indicates the successful completion of the command.

–1 Indicates that an error occurred.

Restrictions

1. The backup node cannot be the same as the primary node.
2. The last character of *vsd_name_prefix* cannot be numeric.
3. The *vsd_name_prefix* cannot contain the character '**.**'.

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

Standard Output

For the following command:

```
createvsd -n 1/2:hdisk13/ -s 1024 -g testvg -v testvsd -T 16
```

The messages to standard output will be similar to:

```
createvsd: calls Getopts.
createvsd: parsing node_list.
createvsd: creates task tables.
createvsd: calls checkclvm.perl on the nodes c164n11.ppd.pok.ibm.com
createvsd: calls domkvglv.perl.
OK:1:mkvg -f -y testvg -s 16 hdisk13
OK:1:mklv -a c -y lvtestvsd1n1 -e x testvg 64 hdisk13
It took about 8 seconds in mkvglv.
createvsd: calls dovaryoffvg.perl testvg on the primary node c164n11.ppd.pok.ibm.com
OK:1:chvg -a n testvg
OK:1:varyoffvg testvg
createvsd: calls doimportvg.perl testvg on the nodes c164n12.ppd.pok.ibm.com with 000048186b991a6f
importvg : testvg
importvg : OK:2:importvg -y testvg hdisk5
importvg : OK:2:chvg -a n testvg
importvg : timestamp 2 testvg 3e036cb33403c8c8
importvg : OK:2:varyoffvg testvg
importvg : It took about 10 seconds.
It took about 12 seconds in importvg.
createvsd: calls vsdvg.
OK:1:vsdvg -g testvgn1b2 testvg 1 2
It took about 12 seconds in vsdvg.
createvsd: calls dovaryonvg.perl testvg on pri nodes c164n11.ppd.pok.ibm.com
OK:1:varyonvg  testvg
createvsd: calls defvsd.
OK:1:defvsd lvtestvsd1n1 testvgn1b2 testvsd1n1
It took about 5 seconds in defvsd.
```

Examples

To create two 4MB virtual shared disks on each of three primary nodes, one of which has a backup, enter:

```
createvsd -n 3,4,7/8/ -c 2 -s 4 -g vsdvg -v TEMP
```

This command creates the following virtual shared disks:

- TEMP1n3, with logical volume lvTEMP1n3 on a volume group with the global volume group name vsdvgn3 on node 3
- TEMP2n4, with logical volume lvTEMP2n4 on a volume group with the global volume group name vsdvgn4 on node 4
- TEMP3n7, with logical volume lvTEMP3n7 on a volume group with the global volume group name vsdvgn7b8 on node 7, also imported to node 8
- TEMP4n3, with logical volume lvTEMP4n3 on a volume group with the global volume group name vsdvgn3 on node 3
- TEMP5n4, with logical volume lvTEMP5n4 on a volume group with the global volume group name vsdvgn4 on node 4
- TEMP6n7, with logical volume lvTEMP6n7 on a volume group with the global volume group name vsdvgn7b8 on node 7, also imported to node 8

To create three virtual shared disks, where the logical volume created on node 3 spans fewer disks than the volume group does, enter:

```
createvsd -n 3,4/:hdisk1,hdisk2+hdisk3/,7/8/ -s 4 -g datavg -v USER
```

This command creates:

- USER1n3, with logical volume lvUSER1n3 defined on a volume group with the global volume group name datavgn3 on node 3.
- USER2n4, with logical volume lvUSER2n4 defined on a volume group with the global volume group name datavgn4 on node 4. datavgn4 spans hdisk1, hdisk2, and hdisk3. lvUSER2n4 spans hdisk1 and hdisk2.
- USER3n7, with logical volume lvUSER3n7 defined on a volume group with the global volume group name datavgn7b8 on node 7, also imported to node 8.
- If no volume group was defined on nodes 3 and 7 before this **createvsd** command was issued, the volume groups datavgn3 and datavgn7b8 are created with one 4MB partition from a single physical disk.

Location

/opt/rsct/vsd/bin/createvsd

Related Information

Commands: **defvsd**, **vsdvg**

---

crfs Command

Purpose

Adds a file system.

Syntax

crfs -v *VfsType* { **-g** *VolumeGroup* | **-d** *Device* } [**-l** *LogPartitions*] **-m** *MountPoint* [**-n** *NodeName*] [**-u** *MountGroup*] [**-A** { **yes** | **no** }] [**-p** {**ro** | **rw** }] [**-a** *Attribute*=*Value* ...] [**-t** { **yes** | **no** }]

Description

The **crfs** command creates a file system on a logical volume within a previously created volume group. A new logical volume is created for the file system unless the name of an existing logical volume is specified using the **-d**. An entry for the file system is put into the **/etc/filesystems** file.

The **crfs** command ignores any *Attribute=Value* pair that the command does not understand but add them to an appropriate stanza in the **/etc/filesystems**.

Example:

```
crfs -a abcd=1G /
```

This will set the new **abcd** attribute to the value of **+1G** in the root stanza in **/etc/filesystems**.

Notes:

1. The file system is created with the **setgid** (set group ID) bit enabled. This determines the default group permissions. All directories created under the new file system will have the same default group permissions.
2. For information about creating a filesystem on a striped logical volume, refer to **File Systems on Striped Logical Volumes** in the **mklv** documentation.

You can use the File Systems application in Web-based System Manager (wsm) to change file system characteristics. You could also use the System Management Interface Tool (SMIT) **smit crfs** fast path to run this command.

Flags

-a *Attribute*=*Value* Specifies a virtual file system-dependent attribute/value pair. To specify more than one attribute/value pair, provide multiple **-a** *Attribute=Value* parameters (see example).

The following attribute/value pairs are specific to the Journaled File System (JFS):

-a 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.

-a bf={ true | false }
Specifies a large file enabled file system. See ″Understanding Large File Enabled File Systems″ 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**.

-a compress={ no | LZ }
Specifies data compression. If you do not want data to be compressed, set this option to **no**. The default compress value is **no**. Selecting compression requires a fragment size of 2048 or less.

-a 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.

-a logname=*LVName*
Specifies the log logical volume name. The specified logical volume will be the logging device for the new JFS. The *LVName* logical volume must already exist. The default action is to use an existing logging device in the target volume group.

-a nbpi={ 512 | 1024 | 2048 | 4096 | 8192 | 16384 | 32768 | 65536 | 131072 }
Specifies the number of bytes per i-node (nbpi). The nbpi affects the total number of i-nodes on the file system. The **nbpi** value is inversely proportional to the number of i-nodes on the file system. The default **nbpi** value is 4096 bytes.

-a size=*Value*
Specifies the size of the Journaled File System. Size can be specified in units of 512-byte blocks, Megabytes or Gigabytes. If Value has the M suffix, it is interpreted to be in Megabytes. If Value has a G suffix, it is interpreted to be in Gigabytes. If the specified size is not evenly divisible by the physical partition size, it is rounded up to the closest number that is evenly divisible. This attribute is required when creating a JFS file system. See ″Understanding JFS Size Limitations″ for more information.

The maximum size of a JFS file system is a function of its fragment size and the NBPI value. These values yield the following size restrictions:

NBPI	Minimum AG Size	Fragment Size	Maximum Size (GB)
512	8	512, 1024, 2048, 4096	8
1024	8	512, 1024, 2048, 4096	16
2048	8	512, 1024, 2048, 4096	32
4096	8	512, 1024, 2048, 4096	64
8192	8	512, 1024, 2048, 4096	128
16384	8	1024, 2048, 4096	256
32768	16	2048, 4096	512
65536	32	4096	1024
131072	64	4096	1024

You can have NBPI values from 512 to 128K, with corresponding maximum file system sizes.

The volume group in which the file system resides defines a maximum logical volume size and also limits the file system size.

Notes:

1. 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 once the file system is created.
2. The root filesystem (**/**) cannot be compressed.
3. Some **nbpi** values and allocation group sizes are mutually exclusive. See ″Understanding JFS Size Limitations″ for information.

The following attribute/value pairs are specific to the Enhanced Journaled File System (JFS2):

-a *Attribute*=*Value*

-a agblksize={ 512 | 1024 | 2048 | 4096 }
Specifies the 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.

-a logname=*LVName*
Specifies the log logical volume name. The specified logical volume is the logging device for the new JFS2. The *LVName* logical volume must already exist. The default action is to use an existing logging device in the target volume group. Keyword **INLINE** can be used to place the log in the logical volume with the JFS2 file system. The **INLINE** log defaults to .4% of the logical volume size if **logsize** is not specified.

-a logsize=*Value*
Specifies the size for an INLINE log in MBytes. The input size must be a positive value. If the inline log size is greater than or equal to 1, the input size must be an integer. If the input is floating point value of less than 1 and greater than or equal to 0, the input size is ignored and the default inline log size is taken.

The input is ignored if the INLINE log not being used. It cannot be greater than 10% of the size of the file system and it cannot be greater than 2047 MBytes.

-a options=*mountOptions*
Specifies which **mount** option is passed into **crfs** for the file system being created. For a list of the valid options, refer to the **mount** command.

-a size=*Value*
Specifies the size of an Enhanced Journaled File System (JFS2). Size can be specified in units of 512-byte blocks, Megabytes or Gigabytes. If *Value* has the M suffix, it is interpreted to be in Megabytes. If *Value* has a G suffix, it is interpreted to be in Gigabytes. If the specified size is not evenly divisible by the physical partition size, it is rounded up to the closest number that is evenly divisible. This attribute is required when creating a JFS2 file system unless the **-d** flag has been specified. If the **-d** flag is specified, the file system is the size of the logical volume. The volume group in which the file system resides defines a maximum logical volume size and limits the file system size. The minimum size for a JFS2 file system is 16 MB. The maximum size is determined by the file system block size:

```
fs block size (byte)        MAX fssize (TB)
========================================
512                         4
1024                        8
2048                        16
4096                        32
```

-a 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 versions of AIX. The default format is **v1**.

-a efs={yes | no}
Specifies whether the file system is an Encrypted File System (EFS).

yes The **crfs** command creates a file system that is EFS-enabled. When the file system is EFS-enabled, you do not need to specify the **ea** attribute because the **crfs** command automatically stores scalable extended attributes of the **v2** format.

no The **crfs** command creates a file system that is not EFS-enabled.

Note: The **crfs** commands prevents EFS from enabling the following file systems (mount points) because the security infrastructures (kernel extensions, libraries and so on) are not available during boot:

- **/**
- **/usr**
- **/var**
- **/opt**

-a isnapshot={yes|no}
Specifies whether the file system supports internal snapshots. A file system created to support internal snapshots also uses extended attributes of the **v2** format.

-a vix={yes|no}
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.

yes The file system can allocate variable-length i-node extents. The **yes** value is the default value beginning with AIX 6.1.

no The file system must use the default size of 16 KB for i-node extents. The **no** value has no effect if the file system contains variable-length i-node extents.

The **crfs** command ignores any Attribute=Value pairs that the command does not understand but adds them to an appropriate stanza in the **/etc/filesystems**

Example:

```
chfs -a abcd=1G/
```

This will set the new ″abcd″ attribute to the value ″+1G″ the root stanza in **/etc/filesystems** will have this line.

-A Specifies whether the file system is mounted at each system restart:

yes File system is automatically mounted at system restart.

no File system is not mounted at system restart.

-d *Device* Specifies the device name of a device or logical volume on which to make the file system. This is used to create a file system on an already existing logical volume.

-g *VolumeGroup* Specifies an existing volume group on which to make the file system. A volume group is a collection of one or more physical volumes.

-l *LogPartitions* Specifies the size of the log logical volume, expressed as a number of logical partitions. This flag applies only to JFS and JFS2 file systems that do not already have a log device.

-m *MountPoint* Specifies the mount point, which is the directory where the file system will be made available. **Note:** If you specify a relative path name, it is converted to an absolute path name before being inserted into the **/etc/filesystems** file.

-n *NodeName* Specifies the remote host name where the file system resides. This flag is only valid with remote virtual file systems such as the Network File System (NFS).

-p Sets the permissions for the file system.

ro Read-only permissions

rw Read-write permissions

-t Specifies whether the file system is to be processed by the accounting subsystem:

yes Accounting is enabled on the file system.

no Accounting is not enabled on the file system (default value).

-u *MountGroup* Specifies the mount group.

-v *VfsType* Specifies the virtual file system type.

Note: The **agblksize** attribute is 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 once the file system is created.

The **ea** attributes 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.

Security

Access Control

Only the root user or a member of the **system** group can run this command.

RBAC Disclaimer

Examples

1. To make a JFS on the `rootvg` volume group with nondefault fragment size and nondefault nbpi, enter:

   ```
   crfs  -v jfs  -g  rootvg  -m /test -a \ size=32768 -a frag=512 -a nbpi=1024
   ```

   This command creates the **/test** file system on the `rootvg` volume group with a fragment size of 512 bytes, a number of bytes per i-node (nbpi) ratio of 1024, and an initial size of 16MB (512 * 32768).
2. To make a JFS on the `rootvg` volume group with nondefault fragment size and nondefault nbpi, enter:

   ```
   crfs -v jfs -g rootvg -m /test -a size=16M -a frag=512 -a nbpi=1024
   ```

   This command creates the **/test** file system on the `rootvg` volume group with a fragment size of 512 bytes, a number of bytes per i-node (nbpi) ratio of 1024, and an initial size of 16MB.
3. To create a JFS2 file system which can support NFS4 ACLs, type:

   ```
   crfs -v jfs2 -g rootvg -m /test -a size=1G -a ea=v2
   ```

   This command creates the **/test** JFS2 file system on the rootvg volume group with an initial size of 1 gigabyte. The file system will store extended attributes using the **v2** format.

Files

/etc/filesystems Lists the known file systems and defines their characteristics.

Related Information

The **chfs** command, **mkfs** command, **mklv** command.

The File systems in *Operating system and device management* explains file system types, management, structure, and maintenance.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

The System Management Interface Tool (SMIT) in *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs* explains the structure, main menus, and tasks that are done with SMIT.

JFS and JFS2 size limitations in *Operating system and device management*.

---

cron Daemon

Purpose

Runs commands automatically.

Syntax

cron [**-f** *configurationfile*] [**-Q**]

Description

The **cron** daemon runs shell commands at specified dates and times. The following event types are scheduled by the **cron** daemon:

- **crontab** command events
- **at** command events
- **batch** command events
- **sync** subroutine events
- **ksh** command events
- **csh** command events

The way these events are handled is specified by the **/var/adm/cron/queuedefs** file.

Regularly scheduled commands can be specified according to instructions contained in the **crontab** files. You can submit your **crontab** file with the **crontab** command. Use the **at** command to submit commands that are to be run only once. Because the **cron** daemon never exits, it should be run only once.

The **cron** daemon examines **crontab** files and **at** command files only when the **cron** daemon is initialized. When you make changes to the **crontab** files using the **crontab** command, a message indicating the change is sent to the **cron** daemon. This eliminates the overhead of checking for new or changed files at regularly scheduled intervals.

Note: When a user is no longer available, the **cron** jobs for that user are no longer run. Even if the user eventually becomes available, **cron** events for that user are no longer queued. The **cron** daemon does not log the information about user availability to the **cronlog** file.

When the **TZ** environment variable is changed, either with the **chtz** command, a Web-based System Manager application, or through SMIT, the **cron** daemon must be restarted. This enables the **cron** daemon to use the correct time zone and summer time change information for the new **TZ** environment variable.

Notes:

1. If you have a job that is scheduled to run between 1:00 a.m. and 2:00 a.m. on the day your time zone changes from daylight saving time to standard time, your job will run twice.
2. If you have a job that is scheduled to run between 2:01 a.m. and 2:59 a.m. on the day your time zone changes from standard time to daylight saving time, your job will not run. You can change the time these jobs run, run them manually, or with until the following day to run them. The **cron** daemon does

not need to be stopped. However, if changes are made to the **TZ** environment variable, kill the current **cron** daemon so that it automatically respawns and recognizes the new **TZ** setting.

3. If you have a job that is scheduled to run at 2:00 a.m. on the day your time zone changes from standard time to daylight saving time, your job will run one second early.

The **cron** daemon reads the **/etc/cronlog.conf** configuration file provided by the user to log the information. If a configuration file has not been created, then the **cron** daemon creates a log of its activities in the **/var/adm/cron/log** file. The **cron** daemon reads the configuration file when it is activated and when it receives the hangup signal.

If the **cron** daemon is not able to create or open the user-specified logfile, then it creates a log of its activities in the **/var/adm/cron/log** file.

Flags

-f *ConfigurationFile*	Specifies an alternate configuration file.
-Q	Quiet mode. If specified, **-Q** disables the **cron** logging. This parameter is valid for a user-configured log file as well as the default **/var/adm/cron/log** file. This option must follow the **-f** option (if **-f** is specified).

Security

Auditing Events

If the auditing subsystem is properly configured and is enabled, the **cron** daemon generates the following audit record (event) every time the command is run:

Event	Information
CRON_Start	Lists the name of each job, whether the job was initiated by an **at** or **cron** command, and the time the job started.
CRON_Finish	Lists the user's name, process ID of the job, and the time the processing was completed.

For more details about how to properly select and group audit events, and how to configure audit event data collection, see Setting up Auditing in *Security*.

RBAC Disclaimer

Files

/var/adm/cron/FIFO	A named pipe that sends messages to the **cron** daemon when new jobs are submitted with the **crontab** or **at** commands.
/var/adm/cron	Specifies the main **cron** daemon directory.
/var/adm/cron/log	Default log file which specifies the accounting information for all the executed **cron**. Contains information like the owner, pid, start time, command, and the exit status of the **cron** job. Rotation is not performed on this file.
/etc/cronlog.conf	Specifies the default **cron** configuration file for logging information.
/var/adm/cron/queuedefs	Specifies the **cron** daemon events file.
/var/spool/cron	Specifies the spool area.
/usr	Indicates directory kept open by the **cron** daemon.
/usr/bin	Indicates directory kept open by the **cron** daemon.
/usr/lib	Indicates directory kept open by the **cron** daemon.
/etc	Indicates directory kept open by the **cron** daemon.
/tmp	Indicates directory kept open by the **cron** daemon.

Configuration File
The configuration file informs the **cron** daemon where and how to log the information. Using the configuration file you can specify logfile names, size limits, rotation policies, compress and archive attributes.

If you do not use the **-f** flag, the **cron** daemon reads the default **/etc/cronlog.conf** configuration file.

If **cron** fails to open the configuration file, it continues with **/var/adm/cron/log**.

The **cron** daemon ignores blank lines and lines beginning with a **#** (pound sign).

Related Information
The **at** command, **auditpr** command, **batch** command, **crontab** command, **csh** command, **ksh** command, **rc** command.

The **sync** subroutine.

The Auditing Overview in *Security* explains more about audits and audit events.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.3 Web-based System Manager Administration Guide*.

---

cronadm Command

Purpose
Lists or removes **crontab** or **at** jobs.

Syntax
To List or Remove crontab Jobs
cronadm cron { { **-l** | **-v** } [*UserName*] ... | **-r** *UserName* }

To List or Remove at Jobs
cronadm at { **-l** [*UserName*] | **-r** { *UserName* | *JobName* } }

Description
The **cronadm** command is used by a root user to list or remove all users **crontab** or **at** jobs.

The **cron** jobs are listed and removed by the *UserName* parameter. One or more *UserNames* can be specified. To list all **cron** jobs, do not specify a user. The **at** jobs are listed by *UserName* and can be removed either by the *UserName* parameter or by the *JobName* parameter.

The name of a **crontab** job file is the name of the user who submitted the **crontab** job and the name of the file in the **/var/spool/cron/crontabs** directory. The name of an **at** job is the name of the user who submitted the **at** job concatenated with a code for the time the **at** job was submitted.

Flags
cronadm cron
-l Lists all **crontab** files. If the *UserName* parameter is specified, only the designated **crontab** files are listed.
-r Removes **crontab** files. The *UserName* parameter should be specified, to remove the designated **crontab** file.

-v Lists the status of all **crontab** jobs. If the *UserName* parameter is specified, only the designated **crontab** files are listed verbosely.

cronadm at

-l Lists the **at** jobs for the user specified by the *UserName* parameter.
-r Removes the **at** job specified by either the *UserName* or *JobName* parameter.

Security

Access Control
Used only by a user with root authority.

Auditing Events
If the auditing subsystem is properly configured and is enabled, the **cronadm** command generates the following audit record (event) every time the command is run:

Event	Information
AT_JobRemove	Lists whether a **crontab** or **at** job was removed and when.

For more details about how to properly select and group audit events, and how to configure audit event data collection, see Setting up Auditing in *Security*.

RBAC Disclaimer

Examples
1. To list all **crontab** jobs, enter:

   ```
   cronadm cron  -l
   ```
2. To list all **at** jobs currently queued for user `bob`, enter:

   ```
   cronadm at  -l bob
   ```

Files

/usr/bin/cronadm	Contains the **cronadm** command.

Related Information
The **at** command, **auditpr** command, **crontab** command.

The **cron** daemon.

The Auditing Overview in *Security* explains more about audits and audit events.

---

crontab Command

Purpose
Submits, edits, lists, or removes cron jobs.

Syntax

crontab [**-e** [*UserName*] | **-l** [*UserName*] | **-r** [*UserName*] | **-v** [*UserName*] | *File*]

Description

The **crontab** command submits, edits, lists, or removes cron jobs. A cron job is a command run by the **cron** daemon at regularly scheduled intervals. To submit a cron job, specify the **crontab** command with the **-e** flag. The **crontab** command invokes an editing session that allows you to create a **crontab** file. You create entries for each cron job in this file. Each entry must be in a form acceptable to the **cron** daemon. For information on creating entries, see The crontab File Entry Format.

When you finish creating entries and exit the file, the **crontab** command copies it into the **/var/spool/cron/crontabs** directory and places it in a file named for your current user name. If a file with your name already exists in the **crontabs** directory, the **crontab** command overwrites it.

Alternatively, you can create a **crontab** file by specifying the *File* parameter. If the file exists, it must be in the format the **cron** daemon expects. If the file does not exist, the **crontab** command invokes the editor. If the **EDITOR** environment variable exists, the command invokes the editor it specifies. Otherwise, the **crontab** command uses the **vi** editor.

To list the contents of your **crontab** file, specify the **crontab** command with the **-l** flag. To remove an existing file, use the **-r** flag.

The optional *UserName* parameter can be used by the owner of the **crontab** file or by the root user to edit, list, remove, or verify the status of the cron jobs for the specified user. If the *UserName* is invalid, an error message is generated and the program exits.

If the optional *UserName* parameter is not specified, the **crontab** flags are available for the root user and the current user.

Security

Only the root user or the owner of the **crontab** file can use *UserName* following the **-e**, **-l**, **-r**, and **-v** flags to edit, list, remove, or verify the **crontab** file of the specified user.

The cron Daemon

The **cron** daemon runs commands according to the **crontab** file entries. Unless you redirect the output of a cron job to standard output or error, the **cron** daemon mails you any command output or errors. If you specify a cron job incorrectly in your **crontab** file, the **cron** daemon does not run the job.

The **cron** daemon examines **crontab** files only when the **cron** daemon is initialized. When you make changes to your **crontab** file using the **crontab** command, a message indicating the change is sent to the **cron** daemon. This eliminates the overhead of checking for new or changed files at regularly scheduled intervals.

Controls on Using the crontab Command

The **/var/adm/cron/cron.allow** and **/var/adm/cron/cron.deny** files control which users can use the **crontab** command. A root user can create, edit, or delete these files. Entries in these files are user login names with one name to a line. If your login ID is associated with more than one login name, the **crontab** command uses the first login name that is in the **/etc/passwd** file, regardless of which login name you might actually be using. Also, to allow users to start **cron** jobs, the daemon attribute in the **/etc/security/user** file should be set to TRUE , using the **chuser** command.

The following is an example of an **cron.allow** file:

```
root
nick
dee
sarah
```

If the **cron.allow** file exists, only users whose login names appear in it can use the **crontab** command. The root user's log name must appear in the **cron.allow** file if the file exists. A system administrator can explicitly stop a user from using the **crontab** command by listing the user's login name in the **cron.deny** file. If only the **cron.deny** file exists, any user whose name does not appear in the file can use the **crontab** command.

A user cannot use the **crontab** command if one of the following is true:

- The **cron.allow** file and the **cron.deny** file do not exist (allows root user only).
- The **cron.allow** file exists but the user's login name is not listed in it.
- The **cron.deny** file exists and the user's login name is listed in it.

If neither the **cron.allow** nor the **cron.deny** file exists, only someone with root user authority can submit a job with the **crontab** command.

The crontab File Entry Format

A **crontab** file contains entries for each cron job. Entries are separated by newline characters. Each **crontab** file entry contains six fields separated by spaces or tabs in the following form:

```
minute  hour  day_of_month  month  weekday  command
```

These fields accept the following values:

minute	0 through 59
hour	0 through 23
day_of_month	1 through 31
month	1 through 12
weekday	0 through 6 for Sunday through Saturday
command	a shell command

You must specify a value for each field. Except for the *command* field, these fields can contain the following:

- A number in the specified range. To run a command in May, specify 5 in the **month** field.
- Two numbers separated by a dash to indicate an inclusive range. To run a **cron** job on Tuesday through Friday, place 2-5 in the **weekday** field.
- A list of numbers separated by commas. To run a command on the first and last day of January, you would specify 1,31 in the **day_of_month** field.
- A combination of two numbers separated by a dash to indicate an inclusive range and a list of numbers separated by commas can be used in conjunction. To run a command on the first, tenth to sixteenth and last day of January, you would specify 1,10-16,31 in the day_of_month field. The above two points can also be used in combination.
- An ***** (asterisk), meaning all allowed values. To run a job every hour, specify an asterisk in the hour field.

  Note: Any character preceded by a backslash (including the %) causes that character to be treated literally. The specification of days may be made by two fields (day of the month and day of the week). If you specify both as a list of elements, both are adhered to. For example, the following entry:

  ```
  0 0 1,15 * 1 command
  ```

would run command on the first and fifteenth days of each month, as well as every Monday. To specify days by only one field, the other field should contain an ***** .

Specifying Commands

The **cron** daemon runs the command named in the sixth field at the selected date and time. If you include a % (percent sign) in the sixth field, the **cron** daemon treats everything that precedes it as the command invocation and makes all that follows it available to standard input, unless you escape the percent sign (\%). Blank lines and lines whose first non-blank character is the number sign (#) will be ignored. If the arguments to the command have a backslash ('\'), the backslash should be preceded by another backslash.

Note: The shell runs only the first line of the command field. All other lines are made available to the command as standard input.

The **cron** daemon starts a subshell from your **HOME** directory. If you schedule a command to run when you are not logged in and you want commands in your **.profile** file to run, the command must explicitly read your **.profile** file.

The **cron** daemon supplies a default environment for every shell, defining **HOME, LOGNAME, SHELL** (=**/usr/bin/sh**), and **PATH** (=**/usr/bin**).

Flags

-e *UserName*	Edits a copy of the user's **crontab** file or creates an empty file to edit if the **crontab** file does not exist for a valid *UserName*. When editing is complete, the file is copied into the **crontab** directory as the user's **crontab** file.
-l *UserName*	Lists the user's **crontab** file.
-r *UserName*	Removes the user's **crontab** file from the **crontab** directory.
-v *UserName*	Lists the status of the user's cron jobs.

Security

Auditing Events

If the auditing subsystem is properly configured and is enabled, the **crontab** command generates the following audit record (event) every time the command is run:

Event	Information
CRON_JobRemove	Lists which users removed a **cron** job and when.
CRON_JobAdd	Lists which users added a **cron** job and when.

For more details about how to properly select and group audit events, and how to configure audit event data collection, see Setting up Auditing in *Security*.

RBAC Disclaimer

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Examples

1. To copy a file called `mycronjobs` into the **/var/spool/cron/crontabs** directory, enter the following:

   ```
   crontab mycronjobs
   ```

   The file will be copied as:

   ```
   /var/spool/cron/crontabs/<username>
   ```

   where <*username*> is your current user name.
2. To write the time to the console every hour on the hour, enter:

   ```
   0 * * * * echo The hour is `date` .
   >/dev/console
   ```
3. To run the **calendar** command at 6:30 a.m. every Monday, Wednesday, and Friday, enter:

   ```
   30 6 * * 1,3,5 /usr/bin/calendar
   ```
4. To run the **calendar** command every day of the year at 6:30, enter the following:

   ```
   30 6 * * * /usr/bin/calendar
   ```
5. To run a script called `maintenance` every day at midnight in August, enter the following:

   ```
   0 0 * 8 * /u/harry/bin/maintenance
   ```
6. To define text for the standard input to a command, enter:

   ```
   0 16 * 12 5 /usr/sbin/wall%HAPPY HOLIDAY!%Remember to
   turn in your time card.
   ```

   The text following the **%** (percent sign) defines the standard input to the **wall** command as:

   ```
   HAPPY HOLIDAY!

   Remember to turn in your time card.
   ```

Files

/var/adm/cron/FIFO	A named pipe that sends messages to the **cron** daemon when new jobs are submitted with the **crontab** or **at** command.
/var/spool/cron/crontabs	Specifies the crontab spool area.
/**var/adm/cron/cron.allow**	Specifies a list of users allowed access to the **crontab** command.
/var/adm/cron/cron.deny	Specifies a list of users denied access to the **crontab** command.

Related Information

The **auditpr** command, **sh** command, **wall** command.

The **cron** daemon.

The Auditing Overview in *Security* explains more about audits and audit events.

---

crvfs Command

Purpose

Creates entries in the **/etc/vfs** file.

Syntax

crvfs *VFSEntry*

Description

The **crvfs** command adds **/etc/vfs** file entries by specifying fields within the *VFSEntry* parameter. The *VFSEntry* parameter is composed of the following fields:
VFSName:VFSNumber:MountHelper:FileSystemHelper.

All fields in the *VFSEntry* parameter are required, but the reserved word ″none″ can be specified for the *MountHelper* and *FileSystemHelper* fields if there is no corresponding helper. If all the arguments are satisfactory, and neither the *VFSName* nor the *VFSNumber* given on the command line already exist, a new entry is created in the **/etc/vfs** file.

Parameters

VFSEntry Specifies a string in the following format: *VFSName*:*VFSNumber*:*MountHelper*:*FileSystemHelper*

VFSName
Specifies the name of a virtual file system type.

VFSNumber
Specifies the virtual file system type's internal number as known by the kernel.

MountHelper
Specifies the name of the backend used to mount a file system of this type.

FileSystemHelper
Specifies the name of the backend used by certain file system specific commands to perform operations on a file system of this type.

Security

RBAC Disclaimer

Examples

To create a new vfs entry called `newvfs`, enter:

```
crvfs "newvfs:4:none:/etc/helpers/newvfshelper"
```

This creates the `newvfs` entry.

Files

/etc/vfs Contains descriptions of virtual file system types.

Related Information

The **chvfs** command, **lsvfs** command, **mount** command, **rmvfs** command.

The File systems in *Operating system and device management* explains file system types, management, structure, and maintenance.

The Mounting in *Operating system and device management* explains mounting files and directories, mount points, and automatic mounts.

---

csh Command

Purpose

Invokes the C shell.

Syntax

csh [**-v** | **-V**] [**-x** | **-X**] [**-e**] [**-f**] [**-i**] [**-n**] [**-c** *String* | **-s** | **-t**] [**-b**] [*File* [*Parameter*]]

Description

The C shell is an interactive command interpreter and a command programming language that uses syntax similar to the C programming language. The shell carries out commands either interactively from a terminal keyboard or from a file. The **csh** command invokes the C shell.

When you invoke the **csh** command, it begins by looking in your home directory and executing commands from the **.cshrc** file (used to store customized user information) if it exists. If the **csh** command runs as a login shell, it executes commands from your **.cshrc** and **.login** files.

After the shell processes flag arguments, if neither the **-i**, **-c**, **-s**, nor **-t** flag is specified and the *File* [*Parameter*] is specified, then the shell executes the script file identified by the *File* [*Parameter*], including any parameters specified. The script file specified must have read permission; the shell ignores any **setuid** and **setgid** settings.

Note: You should not specify a script file if you use the **csh** command with either the **-c** or **-s** flag.

If you specify a script file, the command opens the file and saves the script file name for possible resubstitution by `$0` (dollar sign, zero). The script will then be carried out by **csh**. Remaining parameters initialize the **argv** variable.

Notes:

1. If C shell is already running, the **.cshrc** file can be read again by typing `source Pathname`, where the *Pathname* parameter is the path to the **.cshrc** file.
2. To avoid problems with remote operations, the **.cshrc** file should not contain any functions that echo output unless they test for the **$prompt** variable, which signifies that the shell is interactive. Otherwise, whenever a remote system uses the **exec** command on a command sent by the local system, both the command and the shell are carried out. For example, `exec csh rcp -t Filename` executes the **.cshrc** file and treats the echoed output as the expected response. An **if** clause can be used to check for the **$prompt** variable.

Flags

If the first argument to a shell is a **-** (minus sign), that shell is a login shell. The C shell flags are interpreted as follows:

-b Forces a break from option processing, causing any further shell arguments to be treated as non-option arguments. This flag can be used to pass options to a shell script without confusion or possible subterfuge. The shell cannot run a script whose real and effective user and group IDs differ without this flag.

-c Reads commands from the following single argument, which must be present. Any remaining arguments are placed in the **argv** variable.

-e Exits if any invoked command ends abnormally or yields a nonzero exit status.

-f Starts the C shell without searching for or running commands from the **.cshrc** file in your home directory.

-i Prompts for its top-level input (an interactive shell), even if input does not appear to be coming from a workstation. Shells are interactive without this flag if their input and output are attached to workstations.

-n Parses commands but does not run them. This flag aids you in syntactic checking of shell procedures.

-s Takes command input from standard input.

-t Reads and processes a single line of input. You can use a \ (backslash) to escape the new-line character at the end of the current line and continue onto another line.

-V Sets the **verbose** shell variable before the **.cshrc** file runs.

-v Sets the **verbose** shell variable, so that command input is echoed after history substitution.

-X Sets the **echo** shell variable even before the **.cshrc** file runs.

-x Sets the **echo** shell variable, so that commands are echoed after all substitutions and immediately before they run.

Files

$HOME/.cshrc	Read at the beginning of execution by each shell. The **.cshrc** file is user-defined.
$HOME/.login	Read by the login shell after the **.cshrc** file at login.
$HOME/.logout	Read by the login shell at logoff.
/usr/bin/sh	Contains the path to the default shell.
/tmp/sh*	Contains the temporary file for **<<**.
/etc/passwd	Contains the source of home directories for the ~*File* parameter.

Related Information

The **bsh** command, **chuser** command, **ksh** command, **sh** command.

The **environment** file.

C shell in *Operating system and device management*.

---

csmstat Command

Purpose

csmstat – Provides a snapshot of cluster node reachability, power status, and network interface status.

Syntax

csmstat [**–h**]

csmstat [**–l**] [**–a**] [**–S**] [**–s** *select_string*] [**–d** *delimiter*] [**–n** *node_list*] [**–N** *nodegroups*]

Description

The **csmstat** command gathers node reachability, power status and network interface status for one or more nodes and displays the output. The default ordering for output is by host name. If there are multiple hardware control points for a node, multiple HMCs for example, then the first hardware control point in the list is shown.

Note: This command does not currently support nodes on IntelliStation® workstations.

Flags

–a	Displays attribute information for all nodes. This is the default.
–d	Specifies delimiter-formatted output using the specified delimiter – colons, for example. Use this flag to specify a delimiter of one or more characters. This flag cannot be used with the **–a** flag.
–h	Displays command usage.
–l	Returns LCD values for SP Nodes, p660 servers, and HMC-attached IBM System p servers. This flag cannot be used with the **–d** flag.
–n *node_list*	Specifies a comma or space-separated list of node names to display attribute information. Space-separated node names must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.

–N *nodegroups*
Specifies a comma or space-separated list of node groups to display attribute information. Space-separated node groups must be inside double quotes.

–s Specifies, by column headers, which columns to display. **Hostname** is displayed by default. Other values include **HWControlPoint**, **LCDS**, **Network-Interfaces**, **Status**, **PowerStatus** and **all**. This flag cannot be used with the **–l** flag.

–S Sorts output first by hardware control point and then by host name.

Parameters

None.

Security

The command requires root access to the cluster management server and a user ID with access to the IBM.NodeHwCtrl resource class in the RMC **ctrmc.acls** ACL file.

This command could require a **systemid** file. For more information, see the **systemid** man page.

Exit Status

Hostname
Host name for management of the node. This value will be truncated to seventeen characters. The seventeenth character is a ~ to indicate that truncation was used.

HWControlPoint
Host name of the network adapter for the hardware control point. This value will be truncated to seventeen characters. The seventeenth character is a ~ to indicate that truncation was used.

Status
Indicates if the node is reachable on the network and if the RMC subsystem on the node can communicate with the RMC subsystem on the management server. The valid states are **0** (off), **1** (on) and **127** (unknown). The English representation will be used except when using a delimiter.

PowerStatus
Indicates the current power status of the node. The valid states are **0** (off), **1** (on), **127** (unknown), and **128** (hardware control not configured). The English representation will be used except when using a delimiter.

NetworkInterface
Contains the *Name* of the device and the *OpState*.

Name The name of the network interface. For example, **eth0** on Linux and **en0** on AIX. Switch Network interfaces are also shown.

OpState
Represents the current state of the network interface. The valid states are:

1 Online

2 Offline

Examples

1. The following command returns information in the default format:

```
csmstat

-------------------------------------------------------------------------
Hostname         HWControlPoint  Status  PowerStatus Network-Interfaces
-------------------------------------------------------------------------
clsn10.pok.ibm.c~  /dev/tty2     off     off         unknown
```

```
clsn11.pok.ibm.c~  /dev/tty3     off     off         unknown
clsn12.pok.ibm.c~  /dev/tty4     unknown on          unknown
clsn13.pok.ibm.c~  /dev/tty4     unknown on          unknown
clsn14.pok.ibm.c~  /dev/tty4     unknown off         unknown
clsn15.pok.ibm.c~  /dev/tty4     unknown on          unknown
clsn16.pok.ibm.c~  /dev/tty4     unknown on          unknown
clsn17.pok.ibm.c~  /dev/tty4     unknown on          unknown
clsn18.pok.ibm.c~  /dev/tty4     on      off         en0-Online
```

2. The following command returns information with the specified delimiter:

```
csmstat -d ::

clsn10.pok.ibm.com::/dev/tty2::0::0::unknown
clsn11.pok.ibm.com::/dev/tty3::0::0::unknown
clsn12.pok.ibm.com::/dev/tty4::127::1::unknown
clsn13.pok.ibm.com::/dev/tty4::127::1::unknown
clsn14.pok.ibm.com::/dev/tty4::127::0::unknown
clsn15.pok.ibm.com::/dev/tty4::127::1::unknown
clsn16.pok.ibm.com::/dev/tty4::127::1::unknown
clsn17.pok.ibm.com::/dev/tty4::127::1::unknown
clsn18.pok.ibm.com::/dev/tty4::1::0::en0-1::
```

3. The following command returns information for the specified column headers:

```
 csmstat -s Status,Network-Interfaces

-----------------------------------------------
Hostname          Status  Network-Interfaces
-----------------------------------------------
clsn10.pok.ibm.c~ on      en0-Online ml0-Offline
clsn11.pok.ibm.c~ on      sn1-Online sn0-Online at2-Online at1-Online at0-Online
                          en1-Offline en0-Online ml0-Offline
clsn12.pok.ibm.c~ on      en0-Online en1-Offline ml0-Offline sn1-Online sn0-Online
clsn13.pok.ibm.c~ off     unknown
clsn14.pok.ibm.c~ on      en0-Online en1-Offline at0-Online at1-Online at2-Online
                          at3-Online sn1-Online sn0-Online ml0-Offline
clsn15.pok.ibm.c~ on      en0-Online en1-Offline at0-Online at1-Online at2-Online
                          at3-Online ml0-Offline sn1-Online sn0-Online
clsn16.pok.ibm.c~ unknown unknown
```

Location

/opt/csm/bin/csmstat

Related Information

- The **lshwstat** command.

---

csplit Command

Purpose

Splits a file into individual files.

Syntax

csplit [**-f** *Prefix*] [**-k**] [**-n** *Number*] [**-s**] *File Argument* ...

Description

The **csplit** command copies the specified file and separates the copy into segments. The original input file, which remains unaltered, must be a text file.

The **csplit** command writes the segments to files **xx00** . . . **xx99**, depending on how many times the *Argument* parameter is specified (99 is the maximum). By default, the *Argument* parameter expects a line number. The following rules apply when you specify multiple line numbers:

- File **xx00** contains the lines from the beginning of the original file up to, but not including, the line number specified in the first *Argument* parameter.
- File **xx01** contains lines beginning with the number specified by the first *Argument* parameter up to, but not including, the line referenced by the second *Argument* parameter. Each line number specified as an argument marks the beginning of a new file.
- File **xx***nn* (the last file created) contains lines beginning with the number specified by the last *Argument* parameter through the end of the file.

For example, if the original file had 108 lines and you entered:

```
csplit orginal.txt 11 72 98
```

the **csplit** command would create four files: the **xx00** file would contain lines 1-10, the **xx01** file would contain lines 11-71, the **xx02** file would contain lines 72-97, the **xx03** file would contain lines 98-108.

The *Argument* parameter can also contain the following symbols and pattern strings:

/*Pattern***/**	Creates a file that contains the segment from the current line up to, but not including, the line containing the specified pattern. The line containing the pattern becomes the current line.
%*Pattern***%**	Makes the line containing the specified pattern the current line, but does not create a file for the segment.
+*Number*	Moves forward the specified number of lines from the line matched by the preceding pattern. For example, `/Page/+5` searches for `Page`, then advances 5 lines.
-*Number*	Moves backward the specified number of lines from the line matched by the preceding pattern. For example, `/Page/-5` searches for `Page`, then backs up 5 lines.
{*Number***}**	Repeats the preceding option the specified number of times. This number can follow any pattern or line number. If it follows a pattern, the **csplit** command reuses that pattern the specified number of times. If it follows a line number, the **csplit** command splits the file from that point for the number of lines specified by the line number.

Put quotation marks around all patterns that contain spaces or other characters special to the shell. Patterns may not contain embedded new-line characters. In an expression such as `[a-z]`, the - (minus sign) means *through*, according to the current collating sequence. A collating sequence may define *equivalence classes* for use in character ranges.

Flags

-f *Prefix*	Specifies the prefix to be used for the created file segments. The default value for this variable is **xx**.
-k	Leaves created file segments intact in the event of an error.
-n *Number*	Changes the number of decimal places used in the created file names. The default is two decimal places, or **xx00** . . . **xx99**. If you specify the **-n 4** flag, for example, new files are named **xx0000** . . . **xx0099**.
-s	Suppresses the display of character counts.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Examples

1. To split the text of `book` into a separate file for each chapter, enter:

   ```
   csplit book "/^ Chapter *[k.0-9]k./" {9}
   ```

   This creates 10 files, **xx00** through **xx09**. The **xx00** file contains the front matter that comes before the first chapter. Files **xx01** through **xx09** contain individual chapters. Each chapter begins with a line that contains only the word `Chapter` and the chapter number.
2. To specify the prefix `chap` for the files created from `book`, enter:

   ```
   csplit  -f chap book "/^ Chapter *[k.0-9]k./" {9}
   ```

   This splits `book` into files named **chap00** through **chap09**.

Files

/usr/bin/csplit	Contains the **csplit** command.

Related Information

The **ed** command, **regcmp** command, **split** command.

Files in *Operating system and device management* introduces you to files and the way you can work with them.

Shells in *Operating system and device management* describes what shells are, the different types of shells, and how shells affect the way commands are interpreted.

---

csum Command

Purpose

The **csum** command calculates a message digest for the specified files using the specified hash algorithm.

Syntax

csum [**-o** *outputfile*] [**-h** *algorithm*] [**-a**] [*File1*, *File2*, ... | **-**]

csum -i *inputfile*[**-h** *algorithm*]

Description

The **csum** command calculates a message digest for the specified files using the specified hash algorithm. This provides a reliable way to verify file integrity.

The **csum** command writes message digests to a specified file which can later be used to verify file integrity. Note that a file can be specified using absolute or relative path names.

Specifying multiple **-i**, **-o** or **-h** flags is not considered an error; the last instance of the flag specified will be used. However, it is an error to use both the **-i** and **-o** flags at the same time.

Flags

-	Specifies input from stdin.
-a	Specifies that one message digest will be generated for all files.

-h *algorithm*
Specifies which hash algorithms the **csum** command will use to generate a message digest or verify the message digest values when using the **-i** option. The following options are available:

- SHA1: Uses the SHA-1 algorithm to generate a 20 byte message digest.
- MD5: Uses the MD5 algorithm to generate a 16 byte message digest.

Note: these options are case sensitive. If this **-h** option is not used, then the **csum** command will default to using the MD5 algorithm for both generating and verifying message digests.

-i *inputfile*
Specifies an input file, generated by the **-o** flag, which contains trusted message-digest values. The **csum** command calculates the message-digest values of the files specified in the input file and verifies that they match the actual message-digest values of the existing file.

The **-h** flag should be used with the **-i** flag to specify which cryptographic hash algorithm is used to generate the input file. If it is not specified, the MD5 algorithm will be used.

If a file specified in the input file generates a message-digest value different than the value stored in the input file or the file does not exist, the test for that file will fail and the **csum** command will continue to process the files specified in the input file.

-o *outputfile*
Specifies an output file that the **csum** command will use to write message-digest values. This flag cannot be used with the **-i** flag. If the file specified already exists, it will be overwritten.

Exit Status

The command returns the following values:

0	Success.
>0	An error occurred.

Examples

1. To calculate the message digest for the files `cars` and `trucks`, type:

   ```
   csum cars trucks
   ```

   Because the **-h** option is not specified, MD5 values are calculated for the files `cars` and `trucks`.

   If `9875DD0B18C15899988F29E9D85346A4` and `E8C3ABB5E1D48FA519135EAB0FE40932` are the MD5 values for `cars` and `trucks`, respectively, the **csum** command outputs the following:

   ```
   9875DD0B18C15899988F29E9D85346A4        cars
   E8C3ABB5E1D48FA519135EAB0FE40932        trucks
   ```

2. To calculate the message digest for all files with file names beginning with *file* and store the output in a file called `mdvalues`, type:

   ```
   csum -o mdvalues file*
   ```

   The output file, `mdvalues`, will contain the following text if the directory where the **csum** command is executed contains the files `file1`, `file2`, and `file3` and the MD5 values for those files are as listed below:

   ```
   B026324C6904B2A9CB4B88D6D61C81D1        file1
   26AB0DB90D72E28AD0BA1E22EE510510        file2
   D7FCE9FEE471194AA8B5B6E47267F03         file3
   ```

3. To verify that the message digests in the file `mdvalues` match the current message-digest values for those same files, type:

   ```
   csum -i mdvalues
   ```

4. To calculate the message digest for the file **user.dat** using the SHA-1 algorithm, type:

```
csum -h SHA1 user.dat
```

   If the SHA-1 value for the **user.dat** file is A77CBB748AC336558AFA1AE7F2B73F3765728E7B, the **csum** command will output the following:

```
A77CBB748AC336558AFA1AE7F2B73F3765728E7B        user.dat
```

Location

/usr/bin/csum

Related Information

The **sum** command.

---

ct Command

Purpose

Dials an attached terminal and issues a login process.

Syntax

ct [**-h**] [**-s***Speed*] [**-v**] [**-w***Number*] [**-x***Number*] *TelephoneNumber* ...

Description

The **ct** command is a Basic Networking Utilities (BNU) command that enables a user on a remote terminal, such as an 3161, to communicate with a workstation over a telephone line attached to a modem at each end of the connection. The user on the remote terminal can then log in and work on the workstation.

A user on the local system issues the **ct** command with the appropriate telephone number to call the modem attached to the remote terminal. When the connection is established, the **ct** command issues a login prompt that is displayed on the remote terminal. The user on the remote terminal enters a login name at the prompt and opens a new shell. The user at the remote terminal then proceeds to work on the workstation just like a local user.

The **ct** command is useful in the following situations:

- A user working off-site needs to communicate with a local system under strictly supervised conditions, and the local user does not want to disclose the workstation's phone number. Because the local system contacts the remote terminal, the remote user does not need to know the telephone number of the local system. Additionally, the local user issuing the **ct** command can monitor the work of the remote user.
- The cost of the connection should be charged either to the local site or to a specific account on the calling workstation. If the remote user has the appropriate access permission and can make outgoing calls on the attached modem, that user can make the equivalent of a collect call. The remote user calls the specified local system, logs in, and issues the **ct** command with the telephone number of the remote terminal, but without the **-h** flag. The local system hangs up the initial link so that the remote terminal is free for an incoming call and then calls back the modem attached to the remote terminal.

If there are no free lines, the **ct** command displays a message to that effect and asks if the local user wants to wait for one. If the reply is no, the **ct** command hangs up. If the local user wants to wait for a free line, the **ct** command prompts for the number of minutes to wait. The **ct** command continues to dial the remote system at one-minute intervals until the connection is established or until the specified amount of time has elapsed.

In order to establish a **ct** connection, the remote user contacts the local user with a regular telephone call and asks the local user to issue the **ct** command. However, if such connections occur regularly at your site, your system administrator may prefer to set up BNU in such a way that a specified local system automatically issues the **ct** command to one or more specified terminals at certain designated times.

Notes:

1. Before issuing the **ct** command, be certain that the remote terminal is attached to a modem that can answer the telephone.
2. If the user issuing the **ct** command does not have root authority, the port used for the connection must be a shared or delayed port. Otherwise, the remote login will fail. For more information on shared and delayed ports, see the **pshare** and **pdelay** commands. In addition, for the **ct** command to succeed on a shared or delayed port, the user invoking the command must be a member of the UNIX-to-UNIX copy program (uucp) user group.

The **ct** command is not as flexible as the BNU **cu** command. For example, the user cannot issue commands on the local system while connected to a remote system through the **ct** command. However, the **ct** command does have two features not available with the **cu** command:

- The user can instruct the **ct** command to continue dialing the specified telephone number until the connection is established or a set amount of time has elapsed.
- The user can specify more than one telephone number at a time to instruct the **ct** command to continue dialing each modem until a connection is established over one of the lines.

If the local user specifies alternate dialing paths by entering more than one number on the command line, the **ct** command tries each line listed in the BNU **Devices** file(s) (by default, the **/etc/uucp/Devices** file) until it finds an available line with appropriate attributes or runs out of entries. If there are no free lines, the **ct** command asks if it should wait for one and, if so, for how many minutes. The **ct** command continues to try to open the dialers at one-minute intervals until the specified time is exceeded. The local user can override this prompt by specifying a time with the **-w***Number* flag when entering the command.

After the user logs off, the **ct** command prompts the user on the remote terminal with a reconnect option; the system can either display a new login prompt or drop the line.

Flags

-h	Prevents the **ct** command from hanging up the current line to answer an incoming call.
-s*Speed*	Specifies the rate at which data is transmitted. The default is 1200 baud.
-v	Allows the **ct** command to send a running narrative to standard error output.
-w*Number*	Specifies the maximum number of minutes that the **ct** command is to wait for a line. The command then dials the remote modem at one-minute intervals until the connection is established or until the specified time has elapsed.
-x*Number*	Starts debugging, which displays detailed information about the command's execution on standard error output on the local system. The *Number* variable specifies the debugging level, and is a single digit from 0 to 9. The recommended debugging level is 9.
TelephoneNumber	Specifies the telephone number of the modem attached to the remote terminal. The *TelephoneNumber* variable can include the digits 0 through 9, - (minus signs) representing delays, = (equal signs) representing secondary dial tones, * (asterisks), and # (pound signs). The telephone number can contain a maximum of 31 characters.

Examples

1. To dial a modem attached to a remote terminal with an internal telephone number, enter:

   ```
   ct 41589
   ```

   The internal telephone number of `4-1589` is dialed. The - (hyphen) is optional. The system responds:

```
Allocated dialer at 1200 baud
Confirm hang_up? (y to hang_up)
```

2. To dial a modem attached to a remote terminal with a local telephone number, enter:

   ```
   ct  -w3 9=5553017
   ```

   The **ct** command dials the local telephone number of `555-3017`, where dialing `9` is required to reach an outside dial tone. A three-minute wait is specified as the maximum number of minutes that the **ct** command is to wait for a line.
3. To dial a modem attached to a remote terminal with a long-distance telephone number, enter:

   ```
   ct  -w5 9=12345557003
   ```

   The command dials the long-distance telephone number of `1-234-555-7003`, where `9` is required to reach an outside dial tone. A five-minute wait is specified as the maximum number of minutes that the **ct** command is to wait for a line.

Files

/usr/bin/ct	Contains the **ct** command.
/etc/uucp/Devices	Lists information about available devices.
/etc/uucp/Dialcodes	Contains dialing code abbreviations.
/etc/uucp/Dialers	Defines modem dialers.
/etc/uucp/Systems	Lists accessible remote systems.
/etc/uucp/Sysfiles	Specifies alternate files to be used as **Systems**, **Devices**, and **Dialers** files.

Related Information

The **cu** command, **pdelay** command, **pshare** command, **tip** command.

---

ctaclfck Command

Purpose

Verifies the contents of a cluster security services ACL file.

Syntax

ctaclfck -f *acl_file_name* [**-s**] [**-c**] [**-u** *user_name*] [**-v**] [**-h**]

Description

The **ctaclfck** command checks the contents of the cluster security services ACL file specified by the **-f** flag. The check is limited to syntactical errors; a semantic check is not performed.

The command opens the ACL file, and reads and compiles one ACL entry at a time. If the command encounters an error, it will report the error to standard output. If the **-c** flag is provided, the command will continue processing after encountering errors until it reaches the end of the file. Otherwise processing will stop after the first error is found and reported.

The **-u** flag directs the command to verify the ACL file contents owned by the specified operating system user identity. The command user must have permission to change to the home directory of the user specified by the **-u** flag, and must also have permission to read files in that directory. If the **-s** flag is specified along with the **-u** flag, the command user must also have permission to set its effective user identity to this identity (see the man page for the operating system command **su** for examples).

When the **-u** flag is specified, the file name provided in the **-f** flag is expected to be the base name of a file that resides in the home directory of the named user. In this case, the file name specified by the **-f** flag must not contain any directory names, including the **./** and **../** directories.

If the **-s** flag is specified, the command creates a file to contain the compiled contents of the ACL file. This permits applications to compile the ACL data buffer in advance to starting the application that uses it, saving the application this processing during its startup procedure or its ACL reading process. The compiled ACL file will have the same name as the ACL file with the extension **.cacl**. The ownership and file system permissions of the new ***.cacl** file will be set to the same ownership and permissions as the ACL file. If the ACL file is not currently owned by the command user, the command user must be capable of changing its effective user identity to the identity of the user that owns the ACL file. If the command is unable to do this, it will not create the ACL buffer file, but will complete verification of the ACL file.

The command checks for the correct ACL entry type, for the proper identity format, and for a valid permission. A valid permission is defined as one containing only operations that are defined by the permission template. The permission template set defined by cluster security services and used by this command follows.

Entry Type	
r	**Format** 0x1 **Permission** read **Operation** generic read operation
w	**Format** 0x2 **Permission** write **Operation** generic write operation
c	**Format** 0x4 **Permission** control **Operation** generic control operation / RMC refresh configuration operation
x	**Format** 0x8 **Permission** run **Operation** generic execute operation
C	**Format** 0x10 **Permission** cancel **Operation** generic cancel operation
q	**Format** 0x20 **Permission** query **Operation** RMC query resource operation
l	**Format** 0x40 **Permission** list **Operation** RMC enumerated resources operation
e	**Format** 0x80 **Permission** event **Operation** RMC event registration / unregistration / querying

Entry Type	
d	**Format** 0x100 **Permission** define **Operation** RMC define / undefine resource operation
v	**Format** 0x200 **Permission** validate **Operation** RMC validate resource handle operation
s	**Format** 0x400 **Permission** set **Operation** RMC set attribute operation

If the **-u** flag is specified, the command searches for the ACL file in the home directory of the specified user. The user must own the file and the permission must be write-only by the user. When the **-u** flag is specified, the ACL file name specified by the **-f** flag must not contain a relative or full path to the file; it must specify the file name only.

Flags

-f *acl_file_name*
Specifies the cluster security services ACL file to be verified. The file name can be a full or relative path name, unless the **-u** flag is specified.

-s Caches the ACL buffer (that resulted from the compilation of the ACL file) into a file. If the ACL file is not owned by the command user, the command user must be able to set its effective user identity to the owner of the ACL file.

–c Instructs the command to continue after encountering errors until the end of file is reached. All errors encountered will be reported regardless of whether or not the **-v** flag is specified. If not specified, command processing will stop after the first error is encountered and reported.

-u *user_name*
Specifies the user name in whose home directory the ACL file resides. When this flag is used, the file name specified by the **-f** flag must be the base name of a file that resides in the named user's home directory; the file cannot contain any directory information, including the **./** and **../** directory names.

-v Writes the command's verbose messages to standard output.

-h Writes the command's usage statement to standard output.

Security

The file system permission of the ACL file is determined by the end user or the application owning the file. If the invoker does not have sufficient authority to read the file or to create the requested compiled ACL file with the same ownership, the command fails.

Restrictions

The **ctaclfck** command works only on ACL files formatted for cluster security services.

Examples

1. To verify the contents of the ACL file **/my_acl_file**:

   ```
   ctaclfck -f /my_acl_file
   ```

2. To verify the contents of the ACL file **../my_acl_file** (relative to the current directory) and provide detailed output:

   ```
   ctaclfck -f ../my_acl_file -v
   ```

3. To completely verify the contents of the ACL file **/u/fluffy/my_acl_file**, which is owned by the operating system user **fluffy**, and store the compiled ACL buffer into a file for later use:

   ```
   ctaclfck -c -u fluffy -f  my_acl_file  -v -s
   ```

Location

/usr/sbin/rsct/bin/ctaclfck Contains the **ctaclfck** command

---

ctags Command

Purpose

Makes a file of tags to help locate objects in source files.

Syntax

ctags [**-u** | **-x**] [**-B** | **-F**] [**-a**] [**-m**] [**-o**] [**-t**] [**-v**] [**-w**] [**-f** *TagsFile*] *File*. ..

Description

The **ctags** command creates a tags file for use with the ex and vi editors from the specified C, Pascal, FORTRAN, yacc, lex, and LISP source files. The tags file consists of locators of programming language specific objects (such as functions and type definitions) within the source files. A locator consists of the object name, the file in which it is defined, and either a basic regular expression or a line number that can be used in searching for the object definition. Specifiers are given in separate fields on the line, separated by spaces or tabs. Using the tags file, ex and vi can quickly find these object definitions.

The following file name suffixes are supported by the **ctags** command:

.c Treated as C-language source code and searched for C routine and macro definitions.
.h Treated as C-language source code and searched for C routine and macro definitions.
.f Treated as FORTRAN-language source code.
.l Treated as LISP-language source code if its first nonspace character is **[** (open bracket), **(** (open parenthesis), or **;** (semicolon). Treated as lex-language source code otherwise.

File names ending with any other suffixes are first examined to see if they contain any Pascal or FORTRAN routine definitions. If not, they are processed again as C-language source code. Files without a **.** (dot) suffix are processed as C-language source code.

The **main** tag is treated specially in C programs. The tag formed is created by prefixing **M** to the file name, removing a trailing **.c** (if any), and removing the leading path name components. This makes use of **ctags** practical in directories with more than one program.

Notes:

1. Recognition of the keywords **function**, an address specification for the **subroutine**, and **procedure** in FORTRAN and Pascal code ignores block structure. The **ctags** command may yield inadequate results if any two Pascal procedures have the same name, even though they are in different blocks.
2. The **ctags** command does not recognize **#if** and **#ifdef** statements.
3. If both the **-B** and **-F** options are specified, the last one specified will take precedence.
4. The **-x** option takes precedence over any options (**-a**, **-u**, or **-f**) that would otherwise create a tags file.
5. When the **-v** option is specified, the **-x** option is implied.
6. The output of the **ctags** command is always sorted by object identifier.

Flags

-a	Appends to the tags file. After appending, **ctags** sorts the tags file.
-B	Causes **ctags** to use backward searching patterns (?. . .?).
-F	Causes **ctags** to use forward searching patterns (/. . ./). This is the default searching pattern.
-f *TagsFile*	Creates a tags file with the name specified by *TagsFile* instead of the default **tags** file.
-m	Causes **ctags** to not create tags for macro definitions.
-o	Causes **ctags** to generate line numbers for typedefs instead of a basic regular expression which is used in searching for the object definition.
-t	Creates tags for typedefs. This flag is on by default due to standards conformance.
-u	Updates the specified files in tags; that is, all references to them are deleted, and the new values are appended to the file. This flag may slow the processing of the command. (It is usually faster to simply rebuild the tags file.)
-v	Produces an index of the form expected by the **vgrind** command on the standard output. This listing contains the function name, file name, and page number (assuming 64-line pages).
-w	Causes **ctags** to suppress diagnostic warning messages.
-x	Causes the **ctags** command to display a list of object names, the line number and file name on which each is defined, as well as the text of that line. This provides a simple, readable, function index. If you specify this flag, the **ctags** command does not build, update, or append a tags file, but writes to standard output.

Examples

1. To write the output of the **ctags** command to standard output for the C-language source files, **x.c**, **y.c**, and **z.c**, enter:

   ```
   ctags -x x.c y.c z.c
   ```

2. To create a tags file named **foo_tags** for all the C-language source files within the current directory, enter:

   ```
   ctags -f foo_tags *
   ```

3. To add additional tags,including type definitions, to the **foo_tags** tags file for the C-language source file **zip.c**, enter:

   ```
   ctags -utf foo_tags zip.c
   ```

Exit Status

The following exit values are returned:

0	Successful completion.
>0	An error occurred.

Files

usr/bin/more/tags Output tags file.

Related Information

The **ex** command, **lex** command, **vgrind** command, **vi** command, **yacc** command.

---

ctcasd Daemon

Purpose

Provides and authenticates RSCT host-based authentication (HBA) security mechanism credentials for the cluster security services.

Syntax

ctcasd [**-b**]

Description

The **ctcasd** daemon is used by the cluster security services library when the RSCT HBA security mechanism is configured and active within the cluster environment. The cluster security services use **ctcasd** when service requesters and service providers try to create a secured execution environment through a network connection. **ctcasd** is not used when service requesters and providers establish a secured execution environment through a local operating system connection such as a UNIX domain socket.

When a service requester and a service provider have agreed to use HBA authentication through the cluster security services, the cluster security services library uses **ctcasd** to obtain and authenticate HBA credentials. Cluster security services does not provide a direct interface to the daemon that can be invoked by user applications.

The **ctcasd** daemon can be started or stopped using system resource controller (SRC) commands.

During startup, the daemon obtains its operational parameters from the **ctcasd.cfg** configuration file. The daemon expects to find this file in the **/var/ct/cfg/** directory. System administrators can modify the operational parameters in this file to suit their needs. If this file is not located, the daemon will use the default configuration stored in **/usr/sbin/rsct/cfg/ctcasd.cfg**.

HBA credentials are derived from the local node's private and public keys. These keys are located in files that are configured in **ctcasd.cfg**. These credentials are encrypted using the public key of the receiving node. Public keys for the nodes within the cluster are stored in a trusted host list file on each node. The location of this file is also defined in the **ctcasd.cfg** configuration file. The system administrator is responsible for creating and maintaining this trusted host list, as well as for synchronizing the lists throughout the cluster.

If the daemon detects that both the node's public and private key files are not present, **ctcasd** assumes that it is being started for the first time and creates these files. The daemon also creates the initial trusted host list file for this node. This file contains an entry for **localhost** and the host names and IP addresses associated with all AF_INET-configured and active adapters that the daemon can detect. Inadvertent authentication failures could occur if the public and private key files were accidentally or intentionally removed from the local system before the daemon was restarted. **ctcasd** creates new keys for the node that do not match the keys stored on the other cluster nodes. If HBA authentication suddenly fails after a system restart, this is a possible source of the failure.

Critical failures detected by the daemon that cause shutdown of the daemon are recorded to persistent storage. In AIX-based clusters, records are created in the AIX error log and the system log.

Flags

–b Starts the daemon in bootstrap mode. The daemon runs as a foreground process and is not controlled by the system resource controller (SRC).

Restrictions

- The **ctcasd** daemon does not encrypt the HBA identity credentials.
- Cluster security services supports its own file formats, private key formats, and public key formats only. Cluster security services does not support secured remote shell formats.

Location

/usr/sbin/rsct/bin/ctcasd	Contains the **ctcasd** daemon

Files

/usr/sbin/rsct/cfg/ctcasd.cfg	Default configuration for the **ctcasd** daemon
/var/ct/cfg/ctcasd.cfg	Configuration for the **ctcasd** daemon, which can be modified by the system administrator
/var/ct/cfg/ct_has.pkf	Default location of the cluster security services public key file for the node
/var/ct/cfg/ct_has.qkf	Default location of the cluster security services private key file for the node
/var/ct/cfg/ct_has.thl	Default location of the cluster security services trusted host list for the node

Related Information

Commands: **ctskeygen**, **startsrc**, **ctsvhbac**, **ctsvhbal**, **ctsvhbar**, **ctsthl**

Files: **ctcasd.cfg**, **ct_has.pkf**, **ct_has.qkf**, **ct_has.thl**

---

ctctrl Command

Purpose

Modifies or displays the trace attributes of system components. You can specify persistent attribute values for components that have not yet been created.

Syntax

To modify the trace attributes of some or all components, use the following command:

ctctrl [**-nru**] *ComponentSelector* ... *subcommand ...*

To dump component buffers into files, use the following command:

ctctrl [**-ru**] {**-D** [**-d** *dirName*] } *ComponentSelector* ...

To specify persistent attribute values for components that have not been created yet, use the following command:

ctctrl -p [**-ru**] *ComponentSelector* ... *subcommand* ...

To specify persistent attribute values that will take effect after the next restart, use the following command:

ctctrl -P [**-ru**] *ComponentSelector* ... *subcommand* ...

To delete persistent attribute customizations, use the following command:

ctctrl -x {**-P**|**-p**} [**-ru**] *ComponentSelector* ...

To query trace attributes of existing components or to query existing persistent attribute customization, use the following command:

ctctrl -q [**-rupP**] {*ComponentSelector* ...}

To display a usage message, use the following command:

ctctrl {**-h** | **-?**}

To enable or disable memory tracing for all components persistently, use the following command:

ctctrl -P {**memtraceon** | **memtraceoff**}

The values of the *ComponentSelector* parameter are as follows:

-c componentPatternList

-l aliasPatternList

-t typePatternList

Each list consists of one or more patterns that are separated by blank spaces or commas. Patterns can contain special characters as described by the **fnmatch** subroutine. You can use the following pattern characters:

- **?**
- *
- **[ ]**

You cannot use character classes and collation sequences inside brackets (**[ ]**). Specifying **-c all** selects all components, if no other *ComponentSelector* parameter is specified.

Description

The **ctctrl** command modifies or displays the trace settings of some or all components. Components are selected by name, by alias, or by type or subtype. The **ctctrl** command can also be used with the **-p** or **-P** flag to specify persistent attribute customization. See the Persistent Customizations section.

To enable or disable component-level tracing for all components immediately and persistently, specify the **memtraceon** or **memtraceoff** subcommand with the **-P** flag. You cannot specify other flags or subcommands with the **-P** flag. You must use the **bosboot** command to make settings persistent across boots.

The modified attribute depends on the subcommand that is passed to the **ctctrl** command. Multiple subcommands can be used in a single **ctctrl** invocation. You can specify the following subcommands:

memtraceon	Turns on memory trace mode.
memtraceoff	Turns off memory trace mode.
memtraceresume	Resumes memory trace mode.
memtracesuspend	Suspends the memory trace mode.

memtracebufsize=sz	Changes the size of the private buffer allocated in memory trace mode.
memtraceminimal	Changes memory trace mode level to 1.
memtracenormal	Changes memory trace mode level to 3.
memtracedetail	Changes memory trace mode level to 7.
memtracemax	Changes the level of the memory trace mode to the maximum detail level 9.
memtracelevel=d	Changes the level of trace of the memory trace mode. Sets it to the specified level.
memtracefilltime	Displays the data retention time (that is, the estimated time to fill the private memory buffer). This is available only if the memory trace mode is on.
systraceon	Turns on the tracing through the system trace.
systraceoff	Turns off the tracing through the system trace.
systraceminimal	Changes system trace mode level to 1.
systracenormal	Changes system trace mode level to 3.
systracedetail	Changes system trace mode level to 7.
systracemax	Changes the level of system trace mode to the maximum detail level 9.
systracelevel=d	Changes the level of trace used to trace through the system trace. Sets it to the specified value.

Note: The **memtracesuspend**, **memtraceresume**, and **memtracefilltime** subcommands cannot be used with the **-p** or the **-P** flag, because these subcommands cannot be used in persistent customizations.

Other subcommands that are not in the previous list can be recognized by individual components. A subcommand that is not recognized by a component is ignored.

Current attribute values can be displayed by using the **-q** flag. If you do not specify the *ComponentSelector* parameter, attribute values are displayed for all components that use component-level tracing.

Persistent Customizations

The **-p** and **-P** flags allow attribute values to be specified for system components that have not been created yet. Thus, attributes for newly created components can be customized before the components become active. The **-p** flag is used to specify customizations for components that will be created in the future, but before you restart the AIX operating system. The **-P** flag is used to specify customizations that will take effect after the next restart. These customizations are added to the **/var/adm/ras/raspertune** file. You must run the **bosboot** command to save these customizations in the boot image and restart the AIX operating system for the customizations to take effect.

The component specified by the *ComponentSelectors* parameter with the **-p** and **-P** flags can contain pattern-matching characters. Thus, a persistent customization can apply to more than one component. In addition, multiple customizations can apply to the same component, if different components are used. If conflicting attribute values are specified in multiple customizations, the last customization takes precedence. If a customization already exists for a specified component, the new customization replaces the old one.

You can specify multiple components with the *ComponentSelectors* parameter when persistent customizations are specified. In all cases, using multiple selectors is equivalent to specifying multiple commands, each with a single component selector. For example, the customization `ctctrl -p -l hdisk0 -l hdisk1 memtracenormal` is equivalent to the following two customizations:

```
ctctrl -p -l hdisk0 memtracenormal
ctctrl -p -l hdisk1 memtracenormal
```

When you use the **-D** flag, a snapshot of trace buffers for selected components is dumped into files. The default directory is **/var/adm/ras/trc_ct**, but you can specify an optional destination directory. One trace file per component is used; all files are named with the full components names. The files are generated and managed in the same way the **trace** command does for multiple processor files.

Customizations specified with the **-p** or **-P** flag are not deleted even after they are used. Therefore, a single customization can affect multiple new components. You can specify the **-x** flag to delete persistent customizations. You must specify the *ComponentSelector* parameter identically to the way you specify it when the customization is created. For example, if a customization is created with the component specified by `-l hdisk0`, the customization cannot be deleted with the component specified by `-l hdisk[0]`, even though both components match the same component alias. When a persistent customization is deleted, no change is made to the attributes of components that are created when the customization is active.

Persistent customizations that are deleted with the **-x** and **-P** flags remain in effect unless you run the **bosboot** command and restart the AIX operating system. You can delete a persistent customization that is created with the **-P** flag after the restart by using the **-x** and **-p** flags. In this case, the customization are active again if you restart the AIX operating system.

If you do not know the customizations that have been made but want to restore the default system setting, you can use one of the following ways:

- In the **/var/adm/ras/raspertune** file, delete the lines relevant to the customizations. Then run the **bosboot** command and restart the AIX operating system.
- Read the **/var/adm/ras/raspertune** file to figure out the appropriate flags and parameters that have been specified. Then use the **-x** flag to delete the customizations as shown in Example 11 on page 641. Run the **bosboot** command and restart the AIX operating system.

The **-r** and **-u** flags can be used when specifying persistent customizations. Using one flag specifies a different name space for the specified component selectors. Using both flags at the same time is equivalent to two separate command invocations, each with one of the flags. For example, the persistent customization `ctctrl -p -l hdisk0 -u -r memtracedetail` is equivalent to the following two separate customizations:

```
ctctrl -p -l hdisk0 -u memtracedetail
ctctrl -p -l hdisk0 -r memtracedetail
```

The following persistent customizations are all distinct, and can be modified or deleted independently.

```
ctctrl -p -l hdisk0 memtracedetail
ctctrl -p -l hdisk0 -r memtracedetail
ctctrl -p -l hdisk0 -u memtracedetail
```

Recursive-down customizations (specified by the **-r** flag) take precedence over all other customizations, regardless of the order in which they are specified relative to other non-recursive-down customizations.

You can query persistent customizations by using the **-q** flag with either the **-P** or **-p** flag. Specifying the **-q** flag with the **-P** flag displays lines from the **/var/adm/ras/raspertune** file. Specifying the **-q** flag with the **-p** and **-r** flags displays the persistent customizations that you originally specified with the **-r** flag. Without the **-r** flag, the **-q** and **-p** flags display the persistent customizations that you specify with or without the **-u** flag.

You can specify multiple subcommands for a persistent customization. If you specify conflicting subcommands, the last subcommand is used. For example, the **memtracenormal** and **memtracedetail** subcommands specify different values for the same error-checking attribute, so the last specified subcommand is used.

Flags

-n	Applies subcommands immediately. This flag is the default if neither the **-p** nor the **-P** flag is used.
-c *componentList*	Specifies a list of component names. Separate the names in the list using a comma or blank space. The **-c all** flag selects all components if it is the only *ComponentSelector*.
-D	Takes a snapshot of the component's private memory buffer and dumps it into files (one file per component). The default output directory can be changed with the **-d** flag.
-d *dirName*	Specifies the directory used for the dump. The default directory is **/var/adm/ras/trc_ct**. If some files already exist, they are overwritten by the new dump request. The **-p** and **-P** flags are mutually exclusive with the **-d** flag.
-h or **-?**	Displays a usage message.
-l *aliasList*	Specifies a list of component aliases. Separate the aliases using a comma or blank space.
-P	Specifies subcommands that will persist across restarts. You must run the **bosboot** command and restart AIX for these commands to be used.
-x	Deletes the persistent customization for the specified components. The *ComponentSelector(s)* must be entered exactly as they were entered when the customization was originally specified.
-p	Specifies persistent subcommands. The specified subcommands are applied to newly created components.
-q	Displays the component trace settings of the components. This flag can also be used with the **-p** or **-P** flag to display persistent customizations.
-r	Applies the subcommands recursively to all subcomponents of the selected components.
-t *type_subtype*	Specifies a list of *type* or *type_subtype* names. Separate the names using a comma or blank space. Valid *type* names include `device`, `filesystem`, `network`, `services`, `storage`, and `ui`. A complete list of *type* and *type_subtype* names is in the **/usr/include/sys/ras_base.h** header file.
-u	Applies the subcommands recursively to the ancestors of the specified components.

Note: You can use the **-u** and **-r** flags together. You can use multiple **-c**, **-l**, and **-t** flags on the command line.

Exit Status

0	The command completes successfully.
>0	An error occurs.

Examples

1. To dump the contents of all Component Trace buffers, use the following command:
   ```
   ctctrl -D -c all
   ```
2. To dump the contents of the mbuf Component Trace buffer to **/tmp**, use the following command:
   ```
   ctctrl -D -d /tmp -c mbuf
   ```
3. To query the state of all Component Trace aware components, use the following command:
   ```
   ctctrl -q
   ```
4. To query the state of only the **netinet** components, use the following command:
   ```
   ctctrl -c netinet -q -r
   ```
5. To turn on memory tracing mode for the **socket** component, use the following command:
   ```
   ctctrl memtraceon -c socket
   ```

6. To persistently turn off component tracing for all components, use the following command:
   ```
   ctctrl -P memtraceoff
   ```

   Note: A **bosboot** is required to make the command persistent across boots.
7. To specify a persistent customization for the userdata component of new JFS2 file systems, use the following command:
   ```
   ctctrl -p -c 'jfs2.filesystem.*.userdata' memtraceminimal
   ```

   Note: The existing userdata components are not affected.
8. To specify a customization that will persist across restarts, use the following command:
   ```
   ctctrl -P -c 'jfs2.filesystem.*.userdata' memtraceminimal
   ```

   If you run the **bosboot** command and restart AIX, minimal component tracing will be in effect for all JFS2 userdata components.
9. To set minimal component tracing for all JFS2 userdata components, use the following command:
   ```
   ctctrl -npP -c 'jfs2.filesystem.*.userdata' memtraceminimal
   ```
10. To specify multiple persistent attribute values for the ethernet component, use the following command:
    ```
    ctctrl -P -c ethernet memtraceminimal memtracebufsize=1m
    ```
11. To delete the customization specified in example 7, use the following command:
    ```
    ctctrl -p -x -c 'jfs2.filesystem.*.userdata'
    ```
12. To list all persistent, recursive-down attribute customization, use the following command:
    ```
    ctctrl -q -p -r
    ```
13. To enable all component traces for the netmalloc component, use the following command:
    ```
    ctctrl memtracedetail -c netmalloc
    ```

    or
    ```
    ctctrl memtracelevel=7 -c netmalloc
    ```
14. To collect **net_malloc_police** trace events in the component trace buffer, use the following command:
    ```
    ctctrl  memtracedetail -c netmalloc.police
    ```

    Note: On AIX 5.3 TL 5 and above, with the new RAS framework, for a 32 bit kernel, the component trace buffer space is limited to 8MB.

Location

/usr/sbin/ctctrl

Files

/var/adm/ras/raspertune	A file containing persistent attribute customization that will be applied after a restart, if you run the **bosboot** command first.
/var/adm/ras/trc_ct	The default directory where all snapshots of buffers are saved.
trc_ct.master	A master trace file that points to the trace files of all components.

Related Information

The **ras_register and ras_unregister** kernel services, **ras_customize** kernel service, **ras_control** kernel service.

The **trcctl** command and **trcrpt** commands.

The **errctrl** and **dumpctrl** commands.

The /var/adm/ras/raspertune file.

cthactrl Command

Purpose

Controls subsystems within a cluster.

Syntax

cthactrl -i <*init_opt*> | **-s** | **-k** | **-b** | **-r** | **-d** | **-z** | **-h**

Description

The **cthactrl** command establishes and controls cluster subsystem information and manages topology services and group services.

Flags

-i <*init_opt***>**
Initializes the group services and topology services subsystems, where <*init_opt*> can be specified as:

-c <*cluster_name***>**
Specifies the cluster name.

-n <*nodenum***>**
Specifies the node number.

-e <*environ***>**
Specifies the subdirectory that contains the cluster access modules.

[-p <*portspec***>]**
Specifies the UDP port numbers for group services and topology services.

For example:

```
cthactrl -i -c filesys -n 1 -e filesys -p "cthats=12347,cthags=12348"
```

-s Starts the group services and topology services subsystems.

-k Stops the group services and topology services subsystems.

-b Rebuilds the group services and topology services subsystems configurations (**machines.lst**, for example).

-r Refreshes the group services and topology services subsystems.

-d Deletes the group services and topology services subsystems.

-z Deinstalls the group services and topology services subsystems.

-h Writes the command's usage statement to standard output.

Security

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

Exit Status

0 Successful completion.

non-zero
A failure has occurred.

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.

Examples

1. To initialize the local node as a part of the cluster of **filesys1** and designate **12347** as the UDP port number for **cthags** and **12348** as the UDP port number for **cthags**, enter:

   ```
   cthactrl -i -c filesys1 -n 1 -p "cthats=12347,cthags=12348" -e filesys1
   ```

2. To start the group services and topology services subsystems (**cthags** and **cthats**), enter:

   ```
   cthactrl -s
   ```

3. To stop the group services and topology services subsystems (**cthags** and **cthats**), enter:

   ```
   cthactrl -k
   ```

Location

/usr/sbin/rsct/bin/cthactrl

Related Information

Commands: **cthagsctrl**, **cthagstune**, **cthatsctrl**, **cthatstune**, **lssrc**, **nlssrc**

---

cthagsctrl Command

Purpose

Controls the group services subsystem.

Syntax

cthagsctrl { **-a** [**-p** *port-number*] **-s** | **-k** | **-d** | **-r** | **-z** | **-h** | **-t** | **-o** }

Description

The **cthagsctrl** control command controls the operation of the group services subsystem (**cthags**) under the control of the system resource controller (SRC).

An instance of the group services subsystem runs on every node of a cluster.

From an operational point of view, the group services subsystem group is organized as follows:

Subsystem	group services
Subsystem group	**cthags**
SRC subsystems	**cthags**
	The **cthags** subsystem is associated with the **hagsd** daemon.
	The subsystem name on the nodes is **cthags**. There is one subsystem per node and each of these subsystems is associated with the cluster to which the node belongs.
Daemon	**hagsd**
	Provides the group services functions.

In general, the **cthagsctrl** command is not issued from the command line. It is normally called by the **cthactrl** command during the creation of the cluster.

The **cthagsctrl** command provides a variety of controls for operating the group services subsystems:

- Adding, starting, stopping, and deleting the subsystems
- Cleaning up the subsystems (deleting them from the cluster)
- Unconfiguring the subsystems from the cluster
- Turning tracing on and off

Adding the subsystem

When the **-a** flag is specified, the control command adds the group services subsystems to the SRC. The control command:

1. Makes sure the **cthags** subsystem is stopped.
2. Gets the port number for the **cthags** subsystem from the cluster data.
3. Removes the **cthags** subsystem from the SRC (in case it is still there).
4. Adds the **cthags** subsystem to the SRC.
5. Does not currently add an entry for the **cthags** group to the **/etc/inittab** file. As a result, **cthags** is required to be started by another subsystem when it is needed.

Starting the subsystem

When the **-s** flag is specified, the control command uses the **startsrc** command to start the group services subsystem, **cthags**.

Stopping the subsystem

When the **-k** flag is specified, the control command uses the **stopsrc** command to stop the group services subsystem, **cthags**.

Deleting or cleaning the subsystem

When the **-d** flag is specified, the control command uses the **rmssys** command to remove the group services subsystems from the SRC. The control command:

1. Makes sure the **cthags** subsystem is stopped.
2. Removes the **cthags** subsystem from the SRC using the **rmssys** command.
3. Removes the port number from the **/etc/services** file.

Turning tracing on

When the **-t** flag is specified, the control command turns tracing on for the **hagsd** daemon using the **traceson** command.

Turning tracing off

When the **-o** flag is specified, the control command turns tracing off (returns it to its default level) for the **hagsd** daemon using the **tracesoff** command.

Refreshing the subsystem

The **-r** flag refreshes the **cthags** subsystem.

Logging

While they are running, the group services daemons provide information about their operation and errors by writing entries in three log files in the **/var/ct/***cluster_name***/log/cthags** directory. The log files are:

- **/var/ct/***cluster_name***/log/cthags**_*nodenum_instnum***.***cluster_name*
- **/var/ct/***cluster_name***/log/cthags**_*nodenum_instnum.cluster_name***.long**
- **/var/ct/***cluster_name***/log/cthags.default.***nodenum_instnum*

The log files contain the log of the **hagsd** daemons on the nodes.

The log file names include these variables:

- *nodenum* is the node number on which the daemon is running.
- *instnum* is the instance number of the daemon.
- *cluster_name* is the name of the cluster in which the daemon is running.

Each daemon limits the log size to a pre-established number of lines. The default is 5000 lines. When the limit is reached, the daemon appends the string **.bak** to the name of the current log file and begins a new log. If a **.bak** version already exists, it is removed before the current log is renamed.

Flags

-a [-p *port number***]**
: Adds the subsystem.

-s
: Starts the subsystem.

-k
: Stops the subsystem.

-d
: Deletes the subsystem.

-t
: Turns tracing on for the subsystem.

-o
: Turns tracing off for the subsystem.

-r
: Refreshes the subsystem.

-z
: Uninstalls the **cthags** subsystem.

-h
: Writes the command's usage statement to standard output.

Security

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

Exit Status

0
: Indicates that the command completed successfully.

a non-zero value
: Indicates that an error occurred.

Restrictions

This command is valid in a peer domain only.

Use this command *only* under the direction of the IBM Support Center.

Standard Output

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

Standard Error

This command writes error messages, as necessary, to standard error.

Examples

1. To add the group services subsystems to the SRC in the current cluster, enter:
   ```
   cthagsctrl -a
   ```
2. To add the group services subsystems with a port number of 12347, enter:
   ```
   cthagsctrl -a -p 12347
   ```
3. To start the group services subsystems in the current cluster, enter:
   ```
   cthagsctrl -s
   ```
4. To stop the group servicess subsystems in the current cluster, enter:
   ```
   cthagsctrl -k
   ```
5. To delete the group services subsystems from the SRC in the current cluster, enter:
   ```
   cthagsctrl -d
   ```
6. To turn tracing on for the group services daemon in the current cluster, enter:
   ```
   cthagsctrl -t
   ```
7. To turn tracing off for the group services daemon in the current cluster, enter:
   ```
   cthagsctrl -o
   ```

Location

/usr/sbin/rsct/bin/cthagsctrl Contains the **cthagsctrl** command

Related Information

Commands: **cthactrl**, **cthagstune**, **lssrc**, **nlssrc**

---

cthagstune Command

Purpose

Changes the group services subsystem tunable parameters at run time.

Syntax

cthagstune [**-l** *log_length*] [**-d** *log_dirsize*]

cthagstune [**-h**]

Description

The **cthagstune** command changes the group services subsystem tunable parameters at run time.

Flags

-l Specifies the maximum log file length. If the value is **0** or a negative number, a default log file length is used.

-d Specifies the maximum log directory size in kilobytes. If the value is **0** or a negative number, a default log directory size is used.

-h Writes the command's usage statement to standard output.

Security

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

Exit Status

0 Indicates that the command completed successfully.

a non-zero value
Indicates that an error occurred.

Restrictions

This command is valid in a peer domain only.

Standard Output

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

Standard Error

This command writes error messages, as necessary, to standard error.

Examples

To change the log file length to 6000 lines and to set the log directory size to approximately 7 megabytes, enter:

```
cthagstune -l 6000 -d 7000
```

Location

/usr/sbin/rsct/bin/cthagstune Contains the **cthagstune** command

Related Information

Commands: **cthactrl**, **cthagsctrl**, **lssrc**, **nlssrc**

---

cthatsctrl Command

Purpose

Controls the topology services subsystem.

Syntax

cthatsctrl { **-a** [**-p** *port-number*] | **-s** | **-k** | **-d** | **-b** | **-t** | **-o** | **-r** | **-h** }

Description

The **cthatsctrl** control command controls the operation of the topology services subsystem. The subsystem is under the control of the system resource controller (SRC) and belongs to a subsystem group called **cthats**. Associated with each subsystem is a daemon and a command that configures and starts the daemon.

An instance of the topology services subsystem runs on every node of a cluster.

Adding the subsystem

When the **-a** flag is specified, the control command uses the **mkssys** command to add the topology services subsystem to the SRC. The control command:

1. Makes sure the **cthats** subsystem is stopped.
2. Gets the port number from the cluster data makes sure the port number is set in the **/etc/services** file. The service name that is entered in the **/etc/services** file is **cthats**.
3. Removes the **cthats** subsystem from the SRC (in case it is still there).
4. Adds the **cthats** subsystem to the SRC.

Starting the subsystem

When the **-s** flag is specified, the control command uses the **startsrc** command to start to start the topology services subsystem, **cthats**.

Stopping the subsystem

When the **-k** flag is specified, the control command uses the **stoptsrc** command to stop the topology services subsystem, **cthats**.

Deleting the subsystem

When the **-d** flag is specified, the control command uses the **rmssys** command to remove the topology services subsystem from the SRC. The control command:

1. Makes sure the **cthats** subsystem is stopped
2. Removes the **cthats** subsystem from the SRC using the **rmssys** command
3. Removes the **cthats** port number from the **/etc/services** file

Rebuilding the configuration

When the **-b** flag is specified, the control command reads the configuration information from the cluster data and builds a configuration file, **machines.lst**, for the topology services daemon.

Turning tracing on

When the **-t** flag is specified, the control command turns tracing on for the topology services daemon using the **traceson** command.

Turning tracing off

When the **-o** flag is specified, the control command turns tracing off (returns it to its default level) for the topology services daemon using the **tracesoff** command.

Refreshing the subsystem

When the **-r** flag is specified, the control command refreshes the subsystem using the **refresh** command. The **-r** flag signals the daemon to read the rebuilt information.

Flags

-a [-p *port-number***]**
: Adds the subsystem.

-s
: Starts the subsystem.

-k
: Stops the subsystem.

-d
: Deletes the subsystem.

-t
: Turns tracing on for the subsystem.

-o Turns tracing off for the subsystem.

-b Rebuilds the topology services configuration file from the configuration information in the cluster data.

-r Refreshes the subsystem.

-h Writes the command's usage statement to standard output.

Security

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

Exit Status

0 Indicates that the command completed successfully.

a non-zero value
Indicates that an error occurred.

Restrictions

This command is valid in a peer domain only.

Use this command *only* under the direction of the IBM Support Center.

Standard Output

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

Standard Error

This command writes any error messages to standard error.

Examples

1. To add the topology services subsystem to the SRC, enter:
   ```
   cthatsctrl -a
   ```
2. To start the topology services subsystem, enter:
   ```
   cthatsctrl -s
   ```
3. To stop the topology services subsystem, enter:
   ```
   cthatsctrl -k
   ```
4. To delete the topology services subsystem from the SRC, enter:
   ```
   cthatsctrl -d
   ```
5. To turn tracing on for the topology services daemon, enter:
   ```
   cthatsctrl -t
   ```
6. To turn tracing off for the topology services daemon, enter:
   ```
   cthatsctrl -o
   ```
7. To rebuild the topology services configuration file from the configuration information in the cluster data, enter:
   ```
   cthatsctrl -b
   ```
8. To signal all the topology services daemons in the cluster to read the new configuration file, enter:
   ```
   cthatsctrl -r
   ```
9. To write usage information to standard output, enter:
   ```
   cthatsctrl -h
   ```

Location

/usr/sbin/rsct/bin/cthatsctrl Contains the **cthatsctrl** command

Related Information

Commands: **cthactrl**, **cthagsctrl**, **cthatstune**, **lssrc**

---

cthatstune Command

Purpose

Views and changes the topology services subsystem's tunable parameters at run time.

Syntax

cthatstune [**-f** [*network1*]**:***frequency1*[**,**[*network2*]**:***frequency2*...]] [**-s** [*network1*]**:***sensitivity1*[**,**[*network2*]**:***sensitivity2*...]] [**-p** *priority*] [**-l** *log_length*] [**-m** *pin_object*] [**-r**] [**-v**] [**-h**]

Description

The **cthatstune** command changes the topology services subsystem's tunable parameters at run time. The topology services subsystem has two types of tunable parameters:

subsystem-wide
: Affects the behavior of the topology services subsystem. This type includes the fixed priority level, the maximum length of the log file, and the object to be pinned in main memory.

per-network
: Affects the behavior of each network. This type includes the heartbeat frequency and sensitivity.

  The **cthatstune** command changes the parameters in the cluster data. The new values will not take effect until the topology services daemon reads in the new values from the cluster data. You can use a refresh operation to instruct the topology services daemon to read the new values from the cluster data. You can start a refresh operation by issuing the **cthatsctrl -r** command or the **cthatstune -r** command on one of the nodes in the cluster.

  In addition to the real values, two special values: **VIEW** and **DEFAULT**, can be used to display the current setting and to use the default value of the tunable parameter, respectively.

  For per-network tunable parameters, in addition to the network name, an empty network name or the special network name **ALL** can be used to specify that the value following the network name applies to all networks.

Flags

-f [*network1***]:***frequency1***[,[***network2***]:***frequency2***...]**
: Specifies the *heartbeat frequency*, which is the interval in seconds between heartbeats, for one or more networks.

  The value of *frequency* can be an integer from **1** to **30**. The default value is **1**.

-s [*network1***]:***sensitivity1***[,[***network2***]:***sensitivity2***...]**
: Specifies the maximum number of missing heartbeats for one or more networks. If this maximum is exceeded, the topology services daemon considers the peer to be inactive.

  The value of *sensitivity* can be any integer from **4** to **40**. The default value is **4**.

–p *priority*
Specifies the fixed priority level. The value of *priority* can be **0**, which means "do not run in fixed priority level," or an integer from **1** to **80**. The default value is **30**.

-l *log_length*
Specifies the maximum log file length (in number of lines). The value of *log_length* can be any integer from **2000** to **1 000 000**. The default value is **5000**.

–m *pin_object* **[,***pin_object***...]**
Specifies the object to be pinned in main memory. Valid values are:

NONE	Does not pin any object in main memory.
TEXT	Specifies the TEXT object to be pinned in main memory.
DATA	Specifies the DATA object to be pinned in main memory.
STACK	Specifies the STACK object to be pinned in main memory.
PROC	Specifies that all pinnable objects should be pinned in main memory. This is the default value.

–r Applies the new tunables and refreshes the topology services subsystem.

–v Provides verbose output.

-h Writes the command's usage statement to standard output.

Security

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

Exit Status

0 Indicates that the command completed successfully.

a non-zero value
Indicates that an error occurred.

Restrictions

This command is valid in a peer domain only.

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

This command writes any error messages to standard error.

Examples

1. To change the fixed priority level to 40, view the current setting of the maximum log file length, and pin default objects in main memory, without making the new setting take effect immediately, enter:
   ```
   cthatstune -p 40 -l VIEW -m DEFAULT
   ```
2. To make the new setting (previously changed by **cthatstune**) take effect, enter:
   ```
   cthatstune -r
   ```

3. To change the fixed priority level to normal, pin program and data segments in main memory, and make the new settings take effect immediately, enter:

   ```
   cthatstune -p 0 -m TEXT,DATA -r
   ```

4. To change the heartbeat frequency of **filesys_net** to 2 and all other networks to 4, change the sensitivity of all other networks to the default value, and make the new settings take effect immediately, enter:

   ```
   cthatstune -f filesys_net:2,:4 -s :DEFAULT -r
   ```

5. To change the heartbeat frequency of **filesys_net** to the default value and **service_net** to 3, change the sensitivity of all networks to 8, pin the entire topology services subsystem in main memory, and make the new settings take effect immediately, enter:

   ```
   cthatstune -f filesys_net:DEFAULT,service_net:3 -s :8 -m PROC -r
   ```

   You can also do this using the following method:

   ```
   cthatstune -f filesys_net:DEFAULT,service_net:3
   cthatstune -s :8
   cthatstune -m PROC
   cthatstune -r
   ```

Location

/usr/sbin/rsct/bin/cthatstune Contains the **cthatstune** command

Related Information

Commands: **cthatsctrl**, **lssrc**

---

ctlvsd Command

Purpose

Sets the operational parameters for the virtual shared disk subsystem on a node.

Syntax

ctlvsd [**–r** *node_number*... | **–R** | **–p** *parallelism* |

–k *node_number*... | **–t** | **–T** | **–v** *vsd_name* ... |

–V | **–C** | **–K** | **–M** *IP_max_message_size*]

Description

The **ctlvsd** command changes some parameters of the virtual shared disk subsystem. When called with no arguments, the command displays the current and maximum cache buffer count, the request block count, the pbuf count, the minimum buddy buffer size, the maximum buddy buffer size, and the overall size of the buddy buffer.

Sequence number information may or may not be displayed. In general, sequence numbers and the options that reset them are managed entirely within the virtual shared disk and recoverable virtual shared disk subsystems.

Flags

–r Resets the outgoing and expected sequence numbers for the nodes specified on the node on which the command is run. Use this flag when another node has either been rebooted, cast out, or all virtual shared disks have been reconfigured on that node. The specified nodes are also cast in.

Note: This option should be used only under direct guidance from IBM Service. It should never be used under normal circumstances.

–R Resets the outgoing and expected sequence number for all nodes on the node on which the command is run. Use this flag after rebooting the node. All nodes in the virtual shared disk network will be cast in.

Note: This option should be used only under direct guidance from IBM Service. It should never be used under normal circumstances.

–p Sets the level of virtual shared disk parallelism to the number specified. The valid range is 1 to 9. The default is 9. A larger value can potentially give better response time to large requests. (See *RSCT for AIX 5L: Managing Shared Disks* for more information regarding tuning virtual shared disk performance.)

This value is the *buf_cnt* parameter on the **uphysio** call that the virtual shared disk IP device driver makes in the kernel. Use **statvsd** to display the current value on the node on which the command is run.

–k Casts out the node numbers specified on the local node. The local node ignores requests from cast out nodes. Use **–r** to cast nodes back in.

Notes:

1. Before using this flag, refer to the “Restrictions” section that follows.
2. This option should be used only under direct guidance from IBM Service. It should never be used under normal circumstances.

–t Lists the current routing table and mbuf headers cached by the virtual shared disk driver.

–T Clears or releases all cached routes.

–v *vsd_name* **...**
Resets the statistics in the number of read and write requests on the specified virtual shared disks.

–V Resets all the configured virtual shared disk’s statistics in the number of read and write requests.

–C Resets the virtual shared disk device driver counters displayed by the **statvsd** command. Exceptions are the outgoing and expected request sequence numbers among the client and server nodes.

–K Casts out all nodes on the local node. Local requests are still honored.

Notes:

1. Before using this flag, refer to the “Restrictions” section that follows.
2. This option should be used only under direct guidance from IBM Service. It should never be used under normal circumstances.

–M Sets the virtual shared disk maximum IP message size. This is the largest sized block of data the virtual shared disk sends over the network for an I/O request. This limit also affects local virtual shared disk I/O block size. The value is in bytes and must not be greater than the maximum transmission unit (MTU) size of the network. All nodes should use the same value. The recommended values are:

- 61440 (60KB) for a switch
- 8192 (8KB) for jumbo frame Ethernet
- 1024 (1KB) for 1500-byte MTU Ethernet

Parameters

vsd_name Specifies a defined virtual shared disk.

Security

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

Restrictions

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

Examples

1. To display the current parameters, enter:

   ```
   ctlvsd
   ```

   The system displays a message similar to the following:

   ```
   The minimum buddy buffer size is 4096.
   The maximum buddy buffer size is 65536.
   The total buddy buffer size is 4 max buffers, 262144 bytes.
   ```

2. To display the current IP routing table, enter:

   ```
   ctlvsd -t
   ```

   The system displays the following information:

   ```
   Route cache information:

    destination  interface  ref  status  direct/gateway   min managed mbuf
        1           ml0      2     Up        Direct             256
   ```

Location

/opt/rsct/vsd/bin/ctlvsd

Related Information

Commands: **lsvsd**, **statvsd**

Books: *RSCT for AIX 5L: Managing Shared Disks* for information on tuning virtual shared disk performance.

---

ctmsskf Command

Purpose

Displays and manages the contents of a message security services (MSS) key file.

Syntax

ctmsskf {**-a** | **-d** | **-l** | **-h**} [**-f** *key_file*] [**-t** *key_type*] [**-v** *key_version*] [**-k** *key_value*]

Description

The **ctmsskf** command displays and manages the contents of a message security services (MSS) typed key file. Use this command to add a key to, delete a key from, or list the contents of a key file.

Adding a key:

When you use this command to add a key entry to a key file, you must specify the following:

- the name of the key file where the key is to be added

- the type of the key to add
- optionally, the version of the key that is to be added to the key file
- the 16-digit value of the key

If the specified key file does not exist, it is created. If the specified key file *does* exist, the **ctmsskf** command verifies that the key type specified for the new key matches the type used by the keys already recorded within the file. Only keys of the same type can be added to an existing key file. When a key is successfully added to the file, that version of the key becomes the *active key version.* If a key version is specified using the **-v** *key_version* flag, *key_version* is used as the new version number and is made the active version. If *key_version* is not specified, the key is added using a key version value that is one greater than the previous active key version number.

Existing versions of a key cannot be replaced. To replace an existing version of a key or to change the value of an existing version of a key, that key version must first be deleted using the **-d** flag, and then added again using the **-a** flag. The command returns an error if you try to add a key that uses a version number already in use by a key within an existing key file. In general, key replacements should only be performed on the value of the key that is currently active, as replacing the value of an older key version makes the older key version active.

Because key versions can be added to the key file in any order, the highest key version number may or may not be the key version that is currently active. Use the **-l** flag to determine which key version is currently active for a file.

Deleting a key:

When you use this command to delete a key entry from a key file, you must specify the following:

- the name of the key file from where the key is to be deleted
- optionally, the type of key to delete
- optionally, the version of the key to delete

If the key specified is empty, does not exist, or does not have a proper header, the command returns an error. If the key type is specified and it does not match the key type in the header of the, the command returns an error. If the key version is specified, the command locates the record corresponding to the version provided and purges it from the file. If there is no such record, the command returns an error. If no key version is provided, the command purges only the records that are marked as inactive.

Listing the contents of a key file:

When you use this command to list the contents of a key file, the following information is displayed:

- the header of the key file.
- the list of keys in the key file.

  The following information is displayed for each key:
  - an indication of whether the record is inactive
  - the version of the key
  - the type of the key
  - the 16-digit value of the key

Flags

–a Adds a key to the key file. The **-f**, **-k**, and **-t** flags must also be specified.

–d Deletes a key from the key file. The **-f** and **-v** flags must also be specified. If the **-t** flag is specified, the command checks to see if the type of the key file is the same as the key type provided.

–l Lists the contents of the key file. The **-f** flag must also be specified. If the **-v** flag is specified, the command lists only the key that matches the version number provided.

–f *key_file*
Specifies the name of the key file. The key file must be a valid key file created by MSS API or by this command.

–t *key_type*
Specifies the type of the key to add. If the specified key file is not empty, the command checks to see if the key type specified matches the key type in the header of the key file. The valid key type values are: **3des_md5**, **aes256_md5**, **des_cbc**, **des_md5**, **rsa512_sha**, and **rsa1024_sha**.

–v *key_version*
Specifies the version of the key.

–k *key_value*
Specifies the 16-digit value of the key.

–h Writes the command's usage statement to standard output.

Security

The file system permission of the key files is determined by the application owning the file. If the invoker doesn't have sufficient authority to open the file, the command fails.

Exit Status

0 The command completed successfully.

4 The caller invoked this command incorrectly, omitting required flags and parameters, or using mutually-exclusive flags. This command terminated without processing the request.

6 A memory allocation request failed during the operation of this command. The command was unable to complete the requested action.

9 If the **-a** flag was specified, the command detected a key within the key file that used the same version number as the one specified by the **-v** flag. If the **-d** flag was specified, the command was unable to locate a key in the key file using the version number specified by the **-v** flag. The key file was not modified.

21 The key file could not be located. Verify that the path name for the key file specified by the **-f** flag is correct.

27 The key type specified by the **-t** flag does not match the type for keys stored in the file specified by the **-f** flag. The requested action was not performed.

30 **ctmsskf** was unable to obtain exclusive use of the key file. Another instance of this command may be running and attempting to modify the same file, or the process that makes use of this key file may be examining the file. Retry the command at a later time.

36 The command user does not have sufficient permission to modify the contents of the key file.

37 The key file appears to be corrupted. Try to list the contents of the file using the **-l** flag to verify if the file is corrupted. Follow the problem resolution advice listed in the error message for further recovery action.

Restrictions

This command works only on MSS-formatted key files.

Standard Output

When the **-h** flag is specified, this command's usage statement is written to standard output. When the **-i** flag is specified, the list of available key generation methods is displayed. When the **-l** flag is specified, one or more keys from the key file are displayed.

Standard Error

Descriptive information for any detected failure condition is written to standard error.

Examples

1. To view the keys contained in the key file **/my_key_file**, enter:
   ```
   ctmsskf -l -f /my_key_file
   ```
2. To view the key with version 9 from the key file **/my_key_file**, enter:
   ```
   ctmsskf -l -v 9 -f /my_key_file
   ```
3. To add a key to the key file **/my_key_file**, enter:
   ```
   ctmsskf -a -t des_cbc -f /my_key_file -k 16_digit_value
   ```
4. To delete a key from the key file **/my_key_file**, enter:
   ```
   ctmsskf -d -f /my_key_file -v 10
   ```
5. To delete all inactive keys in the key file **/my_key_file**, enter:
   ```
   ctmsskf -d -f /my_key_file
   ```

Location

/usr/sbin/rsct/bin/ctmsskf	Contains the **ctmsskf** command

Files

/usr/sbin/rsct/cfg/ctcasd.cfg	Default configuration for the **ctcasd** daemon
/var/ct/cfg/ctcasd.cfg	Configuration for the **ctcasd** daemon, which can be modified by the system administrator
/var/ct/cfg/ct_has.pkf	Default location of the cluster security services public key file for the node
/var/ct/cfg/ct_has.qkf	Default location of the cluster security services private key file for the node
/var/ct/cfg/ct_has.thl	Default location of the cluster security services trusted host list for the node

Related Information

Commands: **ctskeygen**, **startsrc**, **ctsvhbac**, **ctsvhbal**, **ctsvhbar**, **ctsthl**

Files: **ctcasd.cfg**, **ct_has.pkf**, **ct_has.qkf**, **ct_has.thl**

---

ctscachgen Command

Purpose

Creates or replaces an on-disk version of a key cache.

Syntax

ctscachgen -c *file-name* [**-f**] [**-i** | **-n** *enc-key-name* | **-k** *enc-key-value* **-t** *key-type* | **-q**] [**-m** *key-gen-method*] [**-s** *cache-size*] [**-h**]

Description

The **ctscachgen** command generates a key cache and stores the completed cache to an on-disk file named in *file-name*. This file can later be used and updated by applications through the **libct_skc** library interfaces.

Flags allow you to specify the type of key to be generated, using the mnemonics that are used for symmetric key types by the **ctmsskf** command. You can also specify a key value to be used to encrypt the keys available in this cache. The keys are not encrypted by default. In addition, you can specify the number of keys to be stored in the file.

If the file specified in *file-name* exists, it is overwritten, even if the current contents do not match the flags specified on the command line.

Flags

-c *file-name*
: Specifies the name of the key cache file. It can be either the full path or the relative path to the current directory.

-f
: Instructs the command to overwrite an existing key cache file with the same name without asking the invoker to confirm its overwriting.

-i
: Displays information about the key cache file specified with the **-c** flag. The information displayed contains the version of the cache file, the read count, the number of keys in the cache, the type of keys in the cache, and whether they are encrypted with a pre-encryption key. This flag cannot be used in conjunction with the **-n**, **-k**, **-t**, or **-q** flag.

-n *enc-key-name*
: Provides the name of the file that contains the encryption typed key. This flag cannot be used in conjunction with the **-i**, **-k**, **-t**, or **-q** flag.

-k *enc-key-value*
: Specifies the key value, expressed in hexadecimal form (**6fe45d20a**, for example), to be used as the pre-encryption key. By default, no pre-encryption key value is used. This flag must be used with the **-t** flag. It cannot be used in conjunction with the **-i**, **-n**, or **-q** flag.

-t *key-type*
: Provides the type of the encryption key specified by the **-k** option. The valid key types are: **3des_md5**, **aes256_md5**, **des_cbc**, **des_md5**, **rsa512_sha**, and **rsa1024_sha**. This flag must be used with the **-k** flag. It cannot be used in conjunction with the **-i**, **-n**, or **-q** flag.

–q
: Instructs the command to use the host's HBA private key as encryption key used for pre-encrypting the session keys in the on-disk key cache file. This flag cannot be used in conjunction with the **-i**, **-k**, **-t**, or **-n** flag.

-m *key-gen-method*
: Provides the session key generation method. Valid values are: **3des_md5**, **aes256_md5**, and **des_md5**. If you do not specify this flag, the default method for generating the session keys is **des_md5**.

-s *cache-size*
: Provides the size of the on-disk key cache file in terms of number of keys in the cache. If you do not specify this flag, the default cache size is 128 keys.

–h
: Writes the command's usage statement to standard output.

Security

Permissions on the **ctscachgen** command permit only **root** to run the command.

Exit Status

Upon successful completion, the command returns an exit status code of **0** and generates an on-disk key cache file. In the event of a failure, the routine returns the error code and may remove the existing key cache file that the invoker wants to overwrite.

0 The command completed successfully.

4 Flags are mismatched or not valid. *file-name* remains unmodified.

6 A memory allocation request failed during the operation of this command. The command was unable to complete the requested action.

12 The command user cannot remove the existing key cache file (*file-name* remains unmodified) or access or write to the directory where *file-name* resides.

21 There is not enough space to store *file-name* or the *file-name* contents appear corrupt.

27 The key stored in the file specified by the **-c** flag is not valid or is corrupted. *file-name* remains unmodified.

36 The invoker cannot access the file specified by the **-c** flag. *file-name* remains unmodified.

Restrictions

- On-disk key caches are intended to be used solely upon the system on which they were generated. They are not intended to be shared between systems or migrated to another system. If multiple systems access the same key cache file, the protections offered by these keys is lost, because multiple systems and applications have access to information that is supposed to remain secret to a specific application. Therefore, any files created by this command should not be stored in shared file systems or networked file systems.
- Files generated by this command are generated in a host-ordered binary format. This format makes it impossible for a key cache file generated on one architecture (such as a Power platform) to be used on a different architecture (such as an Intel platform).

Standard Output

When the **-h** flag is specified, this command's usage statement is written to standard output. When the **-i** flag is specified, information about the key cache file is written to standard output.

Standard Error

Descriptive information for any detected failure condition is written to standard error.

Examples

1. To view the keys contained in the key file **/my_key_file**, enter:
   ```
   ctmsskf -l -f /my_key_file
   ```
2. To view the key with version 9 from the key file **/my_key_file**, enter:
   ```
   ctmsskf -l -v 9 -f /my_key_file
   ```
3. To add a key to the key file **/my_key_file**, enter:
   ```
   ctmsskf -a -t des_cbc -f /my_key_file -k 16_digit_value
   ```
4. To delete a key from the key file **/my_key_file**, enter:
   ```
   ctmsskf -d -f /my_key_file -v 10
   ```
5. To delete all inactive keys in the key file **/my_key_file**, enter:
   ```
   ctmsskf -d -f /my_key_file
   ```

Location

/usr/sbin/rsct/bin/ctscachgen
Contains the **ctscachgen** command

Files

/usr/sbin/rsct/cfg/ctcasd.cfg	Default configuration for the **ctcasd** daemon
/var/ct/cfg/ctcasd.cfg	Configuration for the **ctcasd** daemon, which can be modified by the system administrator
/var/ct/cfg/ct_has.pkf	Default location of the cluster security services public key file for the node
/var/ct/cfg/ct_has.qkf	Default location of the cluster security services private key file for the node
/var/ct/cfg/ct_has.thl	Default location of the cluster security services trusted host list for the node

Related Information

Commands: **ctmsskf**

---

ctsidmck Command

Purpose

Verifies the cluster security library identity mapping.

Syntax

ctsidmck -h | **-i** | { [**-dl** | **-dm** | **-dh**] **-m** *security_mechanism network_ID* }

Description

A system administrator can use the **ctsidmck** command to verify the mapping that would be obtained by the cluster security library (**libct_sec**) for a specific security network identifier.

The cluster security library establishes a security context through the exchange between a client of a trusted service and the trusted service server. During the creation of the security context, the cluster security library tries to map the client application's security network identity to an identity that may be present on the server node, called the *mapped identity.* The cluster security library uses the mapped identity later on the server in authorization functions such as access control verification. Whether the client application has a mapped identity on the server depends on whether the following identity mapping definition files are present on the server, and whether any of the entries within these files correspond to the security identity being used by the client application:

- **/usr/sbin/rsct/cfg/ctsec_map.global**
- **/var/ct/cfg/ctsec_map.local**
- **/var/ct/cfg/ctsec_map.global**

The location of definitions within these files is important; entries at the head of the file are processed before entries positioned towards the end of the file. The definition rules also allow for wildcarding of entry information and for expansion of certain reserved words. If a definition is incorrectly specified within one of these files, the mapping result may not be as intended. Also, if a definition is positioned after another definition that can successfully map a security network identifier, the mapping result may not be as intended.

ctsidmck allows an administrator to verify that the correct identity mapping definition is used by the cluster security library to map a security network identity. This command is to be executed on the system that

would act as the server. By specifying a security network identifier to this command on the server, the administrator can determine what the mapped identity for that security network identity would be on that system, and what entry was used from the identity mapping definition files to obtain this mapping.

Flags

–h Writes the command's usage statement to standard output.

–i Displays a list of the supported security mechanisms on this system. The command examines the cluster security library configuration on this node, obtains a list of supported security mechanisms, and displays this list. The mechanisms are listed by the mnemonic used by the cluster security library to refer to these mechanisms.

–d Specifies the level of detail in the command output. One of three levels of detail is permitted:

1. low (**l**): the command will only display the mapped identity for *network_ID*. This is the default detail level.
2. medium (**m**): the command will display the mapped identity for *network_ID*, as well as the entry from the identity mapping definition files that yielded the map.
3. high (**h**): the command will display every entry from the identity mapping definition files that is processed until a mapped identity for *network_ID* is found, or until all entries are processed.

-m *security_mechanism*

Specifies the security mechanism that was used to create the security network identifier provided by *network_ID*. *security_mechanism* is a mnemonic that would be used by the cluster security library to refer to this security mechanism. This flag must be specified when the **-h** and the **-i** flags are not provided.

Use the **-i** flag to display a list of the security mechanisms that this system supports.

Parameters

network_ID

Specifies the security network identifier to be mapped. This should be an identity that can be assumed by a client application of a trusted service.

Security

This command is executable only by the root system user and members of the system user group. It is intended for administrator use only, to verify the security configuration of the system. Because the output of the command could be used as a means for determining how to sabotage or circumvent system security, the permissions on this command should not be altered.

Exit Status

0 This command successfully found a mapped identity for *network_ID*.

3 This command detected a failure in the operation of the cluster security library mechanism pluggable module (MPM) corresponding to the security mechanism that was requested. **ctsidmck** was unable to search for a possible mapped identity for *network_ID* in this case. This failure may be accompanied by descriptive output indicating the nature of the MPM failure. Consult this output and perform any recommended actions.

4 The caller invoked this command incorrectly, omitting required flags and parameters, or using mutually-exclusive flags. **ctsidmck** terminated without trying to find a mapped identity for *network_ID*.

6 A memory allocation request failed during the operation of this command. **ctsidmck** was unable to search for a possible mapped identity for *network_ID* in this case.

21 This command was unable to locate any of the identity mapping definition files on the local

system. **ctsidmck** was unable to search for a possible mapped identity for *network_ID* in this case. Verify that at least one identity mapping definition file exists on the system.

22 This command was unable to dynamically load the cluster security library mechanism pluggable module (MPM) corresponding to the security mechanism what was requested. The module may be missing, corrupted, or one of the shared libraries used by this module may be missing or corrupted. **ctsidmck** was unable to search for a possible mapped identity for *network_ID* in this case. This failure may be accompanied by descriptive output indicating the nature of the MPM failure. Consult this output and perform any recommended actions.

37 At least one of the identity mapping definition files on the system appears to be corrupted. The command was unable to search for a possible mapped identity for *network_ID* in this case. Verify that none of the identity mapping files are corrupted, truncated, or contain syntax errors.

38 ctsidmck could not locate a mapped identity for *network_ID*. No entry within any of the identity mapping definition files yielded a mapped identity for the specified security network identifier.

Restrictions

This command works only on MSS-formatted key files.

Standard Output

The **ctsidmck** command writes any mapped identity found for the security network identifier to standard output. If a medium or high level of detail is requested, any definitions displayed by this command are also written to standard output.

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

Standard Error

Descriptive information for any detected failure condition is written to standard error.

Examples

1. To get a list of the security mechanisms that the local system supports, before verifying an identity map, enter:

   ```
   ctsidmck -i
   ```

2. To get only the mapped identity for the RSCT host-based authentication (HBA) mechanism security network identity **zathras@greatmachine.epsilon3.org**, enter:

   ```
   ctsidmck -m unix zathras@greatmachine.epsilon3.org
   ```

3. To see every identity mapping definition that the command checks while searching for a mapped identity for the HBA mechanism's security network identity **glorfindel@rivendell.elvin.net@endor**, enter:

   ```
   ctsidmck -d h -m unix glorfindel@rivendell.elvin.net@endor
   ```

Location

/usr/sbin/rsct/bin/ctsidmck Contains the **ctsidmck** command

Files

/usr/sbin/rsct/cfg/ctsec_map.global
The default identity mapping definition file. This file contains definitions required by the RSCT cluster trusted services in order for these systems to execute properly immediately after software installation. This file is ignored if the cluster-wide identity mapping definition file **/var/ct/cfg/ctsec_map.global** exists on the system. Therefore, any

definitions within this file should also be included in the cluster-wide identity mapping definition file, if that file exists.

/var/ct/cfg/ctsec_map.local Local override to the cluster-wide identity mapping definitions. Definitions within this file are not expected to be shared between nodes within the cluster.

/var/ct/cfg/ctsec_map.global Cluster-wide identity mapping definitions. This file is expected to contain identity mapping definitions that are common throughout the cluster. If this file exists on the system, the default identity mapping definition file is ignored. Therefore, if this file exists, it should also contain any entries that would also be found in the default identity mapping definition file.

Related Information

Commands: **ctskeygen**, **startsrc**, **ctsvhbac**, **ctsvhbal**, **ctsvhbar**, **ctsthl**

Files: **ctcasd.cfg**, **ct_has.pkf**, **ct_has.qkf**, **ct_has.thl**

---

ctskeygen Command

Purpose

Generates cluster security services private and public keys for the local system and stores these keys in locally-mounted files.

Syntax

ctskeygen -n [**-f**] [**-m** *method*] [**-p** *public-file*] [**-q** *private-file*] | **-d** | **-i** | **-h**

Description

The **ctskeygen** command generates host identifier keys — a private key and public key pair — to be used by the cluster security services library (**libct_sec**) in RSCT host-based authentication (HBA). The command creates a new private key for the node, derives a public key from the new private key, and stores these keys to files on the local node.

Whenever the node's private and public keys are modified, the node's new public key must be distributed to all nodes within the cluster and placed in the trusted host list files on these nodes, replacing the previous value stored there for this node. If this is not done, the node that has generated new private and public keys will be unable to authenticate with other nodes in the cluster using HBA authentication.

Flags

–n Generates host identifier keys (private and public keys).

–f Forces **ctskeygen** to record the keys it generates to the private and public key files if these files already exist. By default, the command will not overwrite these files if they exist, because the presence of the files indicates that the cluster security services service may be active. Removing or modifying these files without informing other nodes of the change in the public key value will cause failures in HBA authentications on this node. This flag is not valid with the **-h** or the **-i** flag.

–m *method*
Instructs the command to use the specified key generation method in creating the host identifier keys. Valid parameters for this flag can be displayed using the **-i** flag. This flag is not valid with the **-h** and **-i** flags.

–p *public-file*
Specified the fully-qualified path name of the file to be used to store the local host's public key. If this file exists, the command will not overwrite the contents of this file unless the **-f** flag is also

specified. If the **-p** flag is not specified, the command records this key to the **/var/ct/cfg/ct_has.pkf** file. This flag is not valid with the **-h** and **-i** flags.

–q private-file
Specified the fully qualified path name of the file to be used to store the private key of the local host. If this file exists, the command will not overwrite the contents of this file unless the **-f** flag is also specified. If the **-q** option is not specified, the command records this key to the file **/var/ct/cfg/ct_has.qkf**. This flag is not valid with the **-h** and **-i** flags.

–d Displays the current public key value for the local system.

–i Displays information about the key generation methods supported by this version of the command. **ctskeygen** displays messages to indicate which values are currently supported as arguments to the **-m** flag, and what the command will use as a default setting for the **-m** flag.

–h Writes the command's usage statement to standard output.

Parameters

network_ID
Specifies the security network identifier to be mapped. This should be an identity that can be assumed by a client application of a trusted service.

Security

Permissions on the **ctskeygen** command permit only **root** to run the command.

Exit Status

0 The command completed successfully.

4 The caller invoked this command incorrectly, omitting required flags and parameters, or using mutually-exclusive flags. This command terminated without processing the request.

6 A memory allocation request failed during the operation of this command. The command was unable to complete the requested action.

12 The command user does not have sufficient permission to view or modify the contents of the key file.

21 The key file could not be located or could not be created.

30 **ctskeygen** was unable to obtain exclusive use of the public or private key file. Another instance of this command may be running and attempting to modify the keys, or the **ctcasd** daemon may be examining these files. Retry the command at a later time.

37 The public or private key file appears to be corrupted. Try to view the public key value using the **-d** flag to verify if the file is corrupted. Follow the problem resolution advice listed in the error message for further recovery action.

Restrictions

- Cluster security services supports its own file formats, private key formats, and public key formats only.
- Trusted host lists are modifiable using the **ctsthl** command only.
- Cluster security services does not provide an automated utility for creating, managing, and maintaining trusted host lists throughout the cluster. This is a procedure left to either the system administrator or the cluster management software.

Standard Output

When the **-h** flag is specified, this command's usage statement is written to standard output. When the **-d** flag is specified, the public key value stored in the public key file is written to standard output.

Standard Error

Descriptive information for any detected failure condition is written to standard error.

Examples

1. To obtain the list of supported key generation methods:

   ```
   ctskeygen -i
   ```

2. To create new host identifier keys for the local system using the default settings:

   ```
   ctskeygen -n
   ```

3. To create new host identifier keys for the local system using 512-bit RSA private keys, storing these keys in locations other than the default location:

   ```
   ctskeygen -n -m rsa512 -p /mysec/public -q /mysec/private
   ```

Location

/usr/sbin/rsct/bin/ctskeygen Contains the **ctskeygen** command

Files

/usr/sbin/rsct/cfg/ctsec_map.global
The default identity mapping definition file. This file contains definitions required by the RSCT cluster trusted services in order for these systems to execute properly immediately after software installation. This file is ignored if the cluster-wide identity mapping definition file **/var/ct/cfg/ctsec_map.global** exists on the system. Therefore, any definitions within this file should also be included in the cluster-wide identity mapping definition file, if that file exists.

/var/ct/cfg/ctsec_map.local Local override to the cluster-wide identity mapping definitions. Definitions within this file are not expected to be shared between nodes within the cluster.

/var/ct/cfg/ctsec_map.global Cluster-wide identity mapping definitions. This file is expected to contain identity mapping definitions that are common throughout the cluster. If this file exists on the system, the default identity mapping definition file is ignored. Therefore, if this file exists, it should also contain any entries that would also be found in the default identity mapping definition file.

Related Information

Commands: **ctsthl**, **ctsvhbac**

Daemons: **ctcasd**

Files: **ct_has.pkf**, **ct_has.qkf**, **ct_has.thl**

---

ctsnap Command

Purpose

Gathers configuration, log, and trace information about the Reliable Scalable Cluster Technology (RSCT) components.

Syntax

ctsnap [**-d** *output_directory*] [**-h**]

Description

The **ctsnap** command gathers configuration, log, and trace information about the RSCT components that are installed with the base operating system. The **ctsnap** command collects data only on the local node on which it (**ctsnap**) is running. Depending on the programs that are installed, the following components may be included:

- Audit log resource manager (**IBM.AuditRM**)
- Cluster security services (**ctsec**)
- Common information model resource manager (**IBM.CIMRM**)
- Configuration resource manager (**IBM.ConfigRM**)
- Domain management server resource manager (**IBM.DMSRM**)
- Event management (**ha_em**)
- Event response resource manager (**IBM.ERRM**)
- File system resource manager (**IBM.FSRM**)
- First failure data capture (**ct_ffdc**)
- Group services (**cthags**)
- Host resource manager (**IBM.HostRM**)
- Least-privilege resource manager (**IBM.LPRM**)
- Low-level application programming interface (**lapi**)
- Resource monitoring and control (**ctrmc**)
- Sensor resource manager (**IBM.SensorRM**)
- Topology services (**cthats**)
- Virtual shared disk (**vsd**)
- Recoverable virtual shared disk (**rvsd**)

This command is typically run when a problem is encountered with any of these components in order to provide information to your software service organization.

The output of **ctsnap** is a compressed tar file (**ctsnap.***host_name***.***nnnnnnnn***.tar.Z**) and a log file (**ctsnap.***host_name***.***nnnnnnnn***.log**, where *nnnnnnnn* is the timestamp when the **ctsnap** command was run and *host_name* is the name of the host on which the command is running. Both files should be provided to the software service organization. By default, these files are placed in the **/tmp/ctsupt** directory.

Flags

–d *output_directory*
: Identifies the output directory. The default directory is **/tmp/ctsupt**.

–h
: Writes the command's usage statement to standard output.

Security

Privilege control: only the **root** user can run this command.

Exit Status

0
: Command has run successfully.

1
: Command was not successful.

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

1. To gather RSCT support information, enter:

   ```
   ctsnap
   ```

2. To gather RSCT support information and place it in the **/tmp/mydir** directory, enter:

   ```
   ctsnap -d /tmp/mydir
   ```

Location

/usr/sbin/rsct/bin/ctsnap Contains the **ctsnap** 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.

Related Information

Commands: **lssrc**

---

ctsthl Command

Purpose

Displays and modifies the contents of a cluster security services trusted host list file.

Syntax

ctsthl {**-a** | **-d** | **-h** | **-l** | **-s** } [**-f** *trusted_host_list_file*] [**-n** *host_name*] [**-m** *method*] [**-p** *identifier_value*]

Description

This command displays and modifies the contents of a cluster security services trusted host list file. Unless the **-f** flag is provided, the command performs its operations on the trusted host list file configured in the **ctcasd.cfg** file. **ctsthl** allows the command user to add, modify, or remove entries in the trusted host list for specific hosts. When a host is added or modified, the command user must provide the following information:

- The identity of the host (**zathras.ibm.com** or **129.34.128.54**, for example)
- The host identifier value to be used for this host, in a character string format representing the identifier's hexadecimal value (**b87c55e0**, for example)
- The method that was used to generate the host identifier (see the description of the **ctskeygen -i** command)

The command validates the generation method name, converts the character string representation to binary form, and creates a new entry within the trusted host list file for this host. Generally, the host identifier value is quite large. For instance, the character representation of a RSA 1024-bit generated

identifier is over 256 characters in size. This can cause a problem on systems such as AIX, which limit the command line length to a smaller size. To avoid this problem, use the **ctsthl -a** command from a shell script, or in conjunction with the **xargs** command.

When the contents of the trusted host list file are displayed, **ctsthl** provides the following information for each entry:

- The network identity of the host
- The host identifier value for that host, represented as a character string
- The method used to generate the host identifier

Flags

–a Adds to or replaces a host entry in the trusted host list. The **-n**, **-m**, and **-p** flags also must be provided. If the host specified already exists in the trusted host list file, the entry for that host is modified to match the information provided to this command.

–d Removes a host's entry from the trusted host list file. The **-n** flag also must be provided to indicate the host being removed.

–h Writes the command's usage statement to standard output.

–l Instructs the command to list the contents of the trusted host list file. If this flag is combined with the **-a** or **-d** flags, the contents are displayed after these flags are processed. If this flag is combined with the **-s** flag, any new entries made by the command are displayed, as well as any public key mismatches detected for host names and IP addresses supported by the local system.

–f *trusted_host_list_file*
Specifies the fully-qualified path name of the trusted host list file. If this flag is not provided, the trusted host list file configured in the **ctcasd.cfg** file is used.

–n *host_name*
Specifies the identity of the host to be used in this operation. The identity should be a host name or IP address specification by which the host is known to the cluster's network.

–m *method*
Instructs the command to use the specified key generation method in creating the host identifier keys. You can use the **ctskeygen -i** command to display valid values for *method*.

–p *identifier_value*
Specifies the host identifier value to be stored for the host. This is a character string that represents the hexadecimal value of the host identifier to be stored for this identifier. For example, if the host identifier value is **0xB87C55E0**, this flag would be specified as **-p b87c55e0**. Generally, host identifier keys will be much longer than this example, making it too large for the command line limit on some systems such as AIX. If the resulting command line is too large, use **xargs** to extend it, or issue the command from a shell script.

–s Explores the local system for all known IP addresses and host names associated with AF_INET-configured and active adapters that the daemon can detect. For any host name or IP address on the local system that is not found in the local system's trusted host list file, an entry is added to associate that value with the local system's public key value.

Parameters

network_ID
Specifies the security network identifier to be mapped. This should be an identity that can be assumed by a client application of a trusted service.

Security

Permissions on the **ctsthl** command permit only **root** to run the command.

Exit Status

0 The command completed successfully.

4 The caller invoked this command incorrectly, omitting required flags and parameters, or using mutually exclusive flags. This command terminated without processing the request.

6 A memory allocation request failed during the operation of this command. The command was unable to complete the requested action.

10 The command was unable to locate any configured and active network (AF_INET) interfaces for the local system while processing the **-s** flag. The local system's identities may not be properly recorded to the trusted host list. Verify that at least one AF_INET or AF_INET6 interface is defined and active on the local system and reissue the command.

12 The command user does not have sufficient permission to view or modify the contents of the trusted host list file.

21 The trusted host list file could not be located, or could not be extended to contain a new public key value.

30 **ctsthl** was unable to obtain exclusive use of the trusted host list file. Another instance of this command may be running and attempting to modify the keys, or the **ctcasd** daemon may be examining these files. Retry the command at a later time.

31 The public key value specified by the **-p** flag does not end on a full byte boundary. Make sure the value contains an even number of digits.

37 The key file appears to be corrupted. Try to view the public key value using the **-d** flag to verify if the file is corrupted. Follow the problem resolution advice listed in the error message for further recovery action.

Restrictions

- Cluster security services supports its own host identifier format and trusted host list file format only.
- Trusted host lists are modifiable using this command only.
- Cluster security services does not provide an automated utility for creating, managing, and maintaining trusted host lists throughout the cluster. This is a procedure left to either the system administrator or the cluster management software.

Standard Output

When the **-h** flag is specified, this command's usage statement is written to standard output. When the **-l** flag is specified, the contents of the trusted host list file are written to standard output.

Standard Error

Descriptive information for any detected failure condition is written to standard error.

Examples

1. To view the contents of the trusted host contained in the file **/mythl**, enter:

   ```
   ctsthl -l -f /mythl
   ```

2. To add an entry to the default trusted host list file for the system **zathras.ibm.com**, enter:

   ```
   ctsthl -a -n zathras.ibm.com -m rsa1024 -p 120400a9...
   ```

   Note that this example does not complete the entire identifier value.

3. To add an entry to the default trusted host list file for the system **129.23.128.76**, enter:

   ```
   ctsthl -a -n 129.23.128.76 -m rsa1024 -p 120400a9...
   ```

Note that this example does not complete the entire identifier value.

4. To remove an entry for zathras.ibm.com from the default trusted host list, enter:

```
ctsthl -d -n zathras.ibm.com
```

Location

/usr/sbin/rsct/bin/ctsthl Contains the **ctsthl** command

Files

/usr/sbin/rsct/cfg/ctsec_map.global
The default identity mapping definition file. This file contains definitions required by the RSCT cluster trusted services in order for these systems to execute properly immediately after software installation. This file is ignored if the cluster-wide identity mapping definition file **/var/ct/cfg/ctsec_map.global** exists on the system. Therefore, any definitions within this file should also be included in the cluster-wide identity mapping definition file, if that file exists.

/var/ct/cfg/ctsec_map.local Local override to the cluster-wide identity mapping definitions. Definitions within this file are not expected to be shared between nodes within the cluster.

/var/ct/cfg/ctsec_map.global Cluster-wide identity mapping definitions. This file is expected to contain identity mapping definitions that are common throughout the cluster. If this file exists on the system, the default identity mapping definition file is ignored. Therefore, if this file exists, it should also contain any entries that would also be found in the default identity mapping definition file.

Related Information

Commands: **ctskeygen**

Daemons: **ctcasd**

Files: **ctcasd.cfg**

---

ctsvhbac Command

Purpose

Verifies the configuration for the RSCT host-based authentication (HBA) security mechanism on the local system.

Syntax

ctsvhbac [[**-d** | **-h** | **-m** | **-s**] | [**-e** *msgnum*[,*msgnum*...]] [**-l** { **1** | **2** | **3** | **4** } | **-b**] [**-p** *pubkeyfile*] [**-q** *pvtkeyfile*] [**-t** *thlfile*]]

Description

The **ctsvhbac** command is a verification utility for the RSCT host-based authentication (HBA) security mechanism. Use the **ctsvhbac** command to verify that the local system has configuration and credential files and information, such as private keys and a trusted host list, ready for the HBA security mechanism to use.

This command performs the following series of tests on the configuration of the HBA security mechanism:

- Verifies that the HBA mechanism configuration file is available and can be processed.

- Verifies that the HBA private key file exists and can be processed.
- Verifies that the HBA public key file exists and can be processed.
- Verifies that the private and public keys for the local system are in pair, which means that the public key is known to be derived from the private key.
- Verifies that the HBA trusted host list file exists and can be processed.
- Checks the contents of the HBA trusted host list for all of the host names and network addresses supported by the local node, determining whether entries exist in the trusted host list file for them. If a host name or network address is found, the command verifies that the same public key value that was used in earlier tests is listed for the name or address.

The command user may specify the private key file, public key file, and trusted host list file to use in the command. By default, this information is extracted from the configuration file for the HBA security mechanism.

Flags

-b Produces brief output. When this option is used, the command displays only summary output of the tests and any errors detected. Further details of any errors can be determined by reissuing this command without this option. If the **-l** option is specified, this option is ignored.

-d Displays the list of probes required for successful execution of this command.

-e Specifies a list of error messages that are not to be displayed by this command during its execution. One or more message numbers may be specified. Message numbers must be in the xxxx-yyy format. Multiple messages are to be separated by commas (,) with no white space characters.

-h Displays a help message for this command.

-l Allows the Cluster System Management (CSM) Probe Infrastructure to set the detail level of the output. Accepted levels are:

- **1** Verbose mode. Displays the command purpose summary and status information for all tests.
- **2** Displays the command purpose summary and any attention or error conditions detected in any tests.
- **3** Displays any attention or error conditions detected in any tests.
- **4** Silent mode. Displays errors detected during the tests.

-m Displays a detailed description of the command and its purpose.

-p Specifies the path name of the public key file that is to be used by the command. If this option is not specified, the command will use the public key file currently configured for the HBA security mechanism.

-q Specifies the path name of the private key file that is to be used by the command. If this option is not specified, the command will use the private key file currently configured for the HBA security mechanism.

-s Displays a summary of the purpose for the command.

-t Specifies the path name of the trusted host list file that is to be used by the command. If this option is not specified, the command will use the trusted host list file currently configured for the HBA security mechanism.

Parameters

None.

Security

Permissions on the **ctsvhbac** command permit members of the **bin** user group to execute this command.

Exit Status

Exit status conforms to the CSM Probe Infrastructure conventions.

0 No problems detected. Any messages displayed either are informational or indicate only minor alerts. No administration intervention is required.

10 No problems were detected, but some items found warrant administrator attention. This exit status most commonly occurs if an IP address or host name supported by the local system is not listed in the trusted host list, or is listed with an incorrect public key value. For this exit status, the system administrator should examine the output to determine which conditions were detected, and whether they require corrective action.

To correct the most commonly reported conditions:

- Ensure that any IP addresses or host names that are not in the trusted host list were purposely omitted. If not, update the trusted host list on the local system.
- Repair any entries for local IP addresses and host names that use incorrect public keys.

20 One or more problems were detected. This exit status occurs for the following conditions:

- The HBA security mechanism is configured incorrectly.
- Public and private keys might not be in pair.
- The trusted host list contains none of the IP address or host name values supported by the local system.

Unless these conditions are corrected, authentication requests using the HBA mechanism probably will not be successful on this system. For this exit status, the system administrator must examine the command output to identify and resolve reported problems. To correct reported problems, follow the problem-resolution advice listed in the command output.

127 Unexpected failure in this command. For this exit status, the administrator should verify that at least one network interface is both configured and active on this system.

Restrictions

- Cluster security services supports its own host identifier format and trusted host list file format only.
- Trusted host lists are modifiable using this command only.
- Cluster security services does not provide an automated utility for creating, managing, and maintaining trusted host lists throughout the cluster. This is a procedure left to either the system administrator or the cluster management software.

Standard Output

When the **-h** flag is specified, this command's usage statement is written to standard output. When the **-l** flag is specified, the contents of the trusted host list file are written to standard output.

Standard Error

Descriptive information for any detected failure condition is written to standard error.

Examples

To verify the HBA security mechanism, enter:

```
ctsvhbac
```

Output would be similar to:

```
-------------------------------------------------------------------------
Host Based Authentication Mechanism Verification Check

Private and Public Key Verifications

      Configuration file:  /usr/sbin/rsct/cfg/ctcasd.cfg
                  Status:  Available
                Key Type:  rsa512
                           RSA key generation method, 512-bit key

        Private Key file:  /var/ct/cfg/ct_has.qkf
                  Source:  Configuration file
                  Status:  Available
                Key Type:  rsa512
                           RSA key generation method, 512-bit key

         Public Key file:  /var/ct/cfg/ct_has.pkf
                  Source:  Configuration file
                  Status:  Available
                Key Type:  rsa512
                           RSA key generation method, 512-bit key

               Key Parity:  Public and private keys are in pair

Trusted Host List File Verifications

  Trusted Host List file:  /var/ct/cfg/ct_has.thl
                  Source:  Configuration file
                  Status:  Available

                Identity:  avenger.pok.ibm.com
                  Status:  Trusted host

                Identity:  9.117.10.4
                  Status:  Trusted host

                Identity:  localhost
                  Status:  Trusted host

                Identity:  127.0.0.1
                  Status:  Trusted host

Host Based Authentication Mechanism Verification Check completed
```

Location

/usr/sbin/rsct/bin/ctsvhbac Contains the **ctsvhbac** command

Files

/usr/sbin/rsct/cfg/ctsec_map.global
The default identity mapping definition file. This file contains definitions required by the RSCT cluster trusted services in order for these systems to execute properly immediately after software installation. This file is ignored if the cluster-wide identity mapping definition file **/var/ct/cfg/ctsec_map.global** exists on the system. Therefore, any definitions within this file should also be included in the cluster-wide identity mapping definition file, if that file exists.

/var/ct/cfg/ctsec_map.local Local override to the cluster-wide identity mapping definitions. Definitions within this file are not expected to be shared between nodes within the cluster.

/var/ct/cfg/ctsec_map.global Cluster-wide identity mapping definitions. This file is expected to contain

identity mapping definitions that are common throughout the cluster. If this file exists on the system, the default identity mapping definition file is ignored. Therefore, if this file exists, it should also contain any entries that would also be found in the default identity mapping definition file.

Related Information

Commands: **ctskeygen**, **ctsthl**, **ctsvhbal**, **ctsvhbar**

Files: **ctcasd.cfg**, **ct_has.pkf**, **ct_has.qkf**, **ct_has.thl**

---

ctsvhbal Command

Purpose

Displays the possible identities that the local system may use to identify itself in RSCT host-based authentication (HBA) security mechanism credentials.

Syntax

ctsvhbal [[**-d** | **-h** | **-m** | **-s**] | [**-e** *msgnum*[,*msgnum*...]] [**-l** { **1** | **2** | **3** | **4** } | **-b**]

Description

The **ctsvhbal** command is a verification utility for the RSCT host-based authentication (HBA) security mechanism. It displays the possible identities that the local system may use to identify itself in HBA credentials.

The HBA security mechanism might use either a host name or a network address value as part of the identification information within a credential, depending on the method chosen by the application. If the local system is to service requests from remote systems, at least one network address and host name for that remote system must appear in the trusted host list on the local system. To verify that the remote system can successfully authenticate the local system, system administrators use a combination of RSCT cluster security commands:

1. On both the local and remote system, issue the **ctsvhbac** command to verify that each system has a valid HBA security mechanism configuration.
2. On the local system, issue the **ctsvhbal** command to determine the values that the HBA security mechanism will use to identify this host to a remote system.
3. On the remote system, issue the **ctsvhbar** command, specifying the local system host name or IP address, to determine the value that the remote system will use to verify HBA credentials transmitted from the local system.
4. Compare the **ctsvhbal** and **ctsvhbar** command output to determine whether the two systems are using the same scheme for host-name resolution. If an exact host-name match does not appear in the output, repair the host-name resolution scheme, and repeat the steps above until both commands yield an exact match.

Completing these steps verifies successful authentication in one direction; in other words, the procedure verifies only that the remote system can authenticate requests from the local system. Because RSCT subsystems often use mutual authentication, system administrators also should verify that the local system can successfully authenticate the remote system. To complete the verification, the following additional steps are required:

- On the remote system, issue the **ctsvhbal** command to determine the values that the HBA security mechanism will use to identify that host to the local system.
- On the local system, issue the **ctsvhbar** command, specifying the remote system host name or IP address, to determine the value that the local system will use to verify HBA credentials transmitted from the remote system.

- Compare the **ctsvhbal** and **ctsvhbar** command output to determine whether the two systems are using the same scheme for host-name resolution. If an exact host-name match does not appear in the output, repair the host-name resolution scheme, and repeat the steps above until both commands yield an exact match.

Completing these additional steps verifies successful authentication when traffic flows in the opposite direction, from the remote system to the local system.

For more detailed instructions and examples, see the cluster security topics in *RSCT Administration Guide*.

Flags

-b Produces brief output. When this option is used, the command displays only the host identities found for the local system and any errors detected. If the **-l** option is specified, this option is ignored.

-d Displays the list of probes required for successful execution of this command.

-e Specifies a list of error messages that are not to be displayed by this command during its execution. One or more message numbers may be specified. Message numbers must be in the xxxx-yyy format. Multiple messages are to be separated by commas (,) with no white space characters.

-h Displays a help message for this command.

-l Allows the Cluster System Management (CSM) Probe Infrastructure to set the detail level of the output. Accepted levels are:

- **1** Verbose mode. Displays the command purpose summary and status information for all tests.
- **2** Displays the command purpose summary and any attention or error conditions detected in any tests.
- **3** Displays any attention or error conditions detected in any tests.
- **4** Silent mode. Displays errors detected during the tests.

-m Displays a detailed description of the command and its purpose.

-s Displays a summary of the purpose for the command.

Parameters

None.

Security

Permissions on the **ctsvhbal** command permit members of the **bin** user group to execute this command.

Exit Status

Exit status conforms to the CSM Probe Infrastructure conventions.

0 No problems detected. Any messages displayed are informational. No administration intervention is required.

10 No problems were detected, but the local system is unable to authenticate itself to any remote systems. The local system does not have any active network interfaces, which is a configuration that RSCT permits. For this exit status, however, the system administrator should verify that this configuration is appropriate.

20 One or more problems were detected. Host-name resolution mechanisms that the local system uses are unable to obtain host names of network interfaces that the local system supports. Unless this condition is corrected, authentication requests using the HBA mechanism probably will not be

successful on this system. For this exit status, the system administrator should follow the problem-resolution advice listed in the command output.

127 Unexpected failure in this command.

Restrictions

- Cluster security services supports its own host identifier format and trusted host list file format only.
- Trusted host lists are modifiable using this command only.
- Cluster security services does not provide an automated utility for creating, managing, and maintaining trusted host lists throughout the cluster. This is a procedure left to either the system administrator or the cluster management software.

Standard Output

When the **-h** flag is specified, this command's usage statement is written to standard output. When the **-l** flag is specified, the contents of the trusted host list file are written to standard output.

Standard Error

Descriptive information for any detected failure condition is written to standard error.

Examples

To display the possible identities that the local system may use to identify itself in HBA credentials, enter:

```
ctsvhbal
```

Output would be similar to:

```
ctsvhbal: The Host Based Authentication (HBA) mechanism identities for
the local system are:

                  Identity:  zathras.pok.ibm.com

                  Identity:  9.127.100.101

ctsvhbal: At least one of the above identities must appear in the
trusted host list on the node where a service application resides in order
for client applications on the local system to authenticate successfully.
Ensure that at least one host name and one network address identity from the
above list appears in the trusted host list on the service systems used by
applications on this local system.
```

Location

/usr/sbin/rsct/bin/ctsvhbal Contains the **ctsvhbal** command

Files

/usr/sbin/rsct/cfg/ctcasd.cfg Default configuration for the **ctcasd** daemon

/var/ct/cfg/ctcasd.cfg Configuration for the **ctcasd** daemon, which can be modified by the system administrator

Related Information

Commands: **ctskeygen**, **ctsthl**, **ctsvhbac**, **ctsvhbar**

Books: See the cluster security topics in *RSCT Administration Guide*.

ctsvhbar Command

Purpose

Returns the host name that the RSCT host-based authentication (HBA) security mechanism uses on the local node to verify credentials from a specified host.

Syntax

ctsvhbar [[**-d** | **-h** | **-m** | **-s**] | [**-e** *msgnum*[,*msgnum*...]] [**-l** { **1** | **2** | **3** | **4** } | **-b**] {*hostname* | *address*} [*hostname*... | *address*...]

Description

The **ctsvhbar** command is a verification utility for the RSCT host-based authentication (HBA) security mechanism. Use this command when you need to determine which host name the HBA security mechanism uses to verify credentials from a remote system.

The HBA security mechanism might use either a host name or a network address value as part of the identification information within a credential, depending on the method chosen by the application. If the local system is to service requests from remote systems, at least one network address and host name for that remote system must appear in the trusted host list on the local system. To verify that the remote system can successfully authenticate the local system, system administrators use a combination of RSCT cluster security commands:

1. On both the local and remote system, issue the **ctsvhbac** command to verify that each system has a valid HBA security mechanism configuration.
2. On the local system, issue the **ctsvhbal** command to determine the values that the HBA security mechanism will use to identify this host to a remote system.
3. On the remote system, issue the **ctsvhbar** command, specifying the local system host name or IP address, to determine the value that the remote system will use to verify HBA credentials transmitted from the local system.
4. Compare the **ctsvhbal** and **ctsvhbar** command output to determine whether the two systems are using the same scheme for host-name resolution. If an exact host-name match does not appear in the output, repair the host-name resolution scheme, and repeat the steps above until both commands yield an exact match.

Completing these steps verifies successful authentication in one direction; in other words, the procedure verifies only that the remote system can authenticate requests from the local system. Because RSCT subsystems often use mutual authentication, system administrators also should verify that the local system can successfully authenticate the remote system. To complete the verification, the following additional steps are required:

- On the remote system, issue the **ctsvhbal** command to determine the values that the HBA security mechanism will use to identify that host to the local system.
- On the local system, issue the **ctsvhbar** command, specifying the remote system host name or IP address, to determine the value that the local system will use to verify HBA credentials transmitted from the remote system.
- Compare the **ctsvhbal** and **ctsvhbar** command output to determine whether the two systems are using the same scheme for host-name resolution. If an exact host-name match does not appear in the output, repair the host-name resolution scheme, and repeat the steps above until both commands yield an exact match.

Completing these additional steps verifies successful authentication when traffic flows in the opposite direction, from the remote system to the local system.

For more detailed instructions and examples, see the cluster security topics in *RSCT Administration Guide*.

Flags

-b Produces brief output. When this option is used, the command displays the host identities provided by the command user, the fully qualified host identities obtained for them, and any errors. If the **-l** option is specified, this option is ignored.

-d Displays the list of probes required for successful execution of this command.

-e Specifies a list of error messages that are not to be displayed by this command during its execution. One or more message numbers may be specified. Message numbers must be in the xxxx-yyy format. Multiple messages are to be separated by commas (,) with no white space characters.

-h Displays a help message for this command.

-l Allows the Cluster System Management (CSM) Probe Infrastructure to set the detail level of the output. Accepted levels are:

- **1** Verbose mode. Displays the command purpose summary and status information for all tests.
- **2** Displays the command purpose summary and any attention or error conditions detected in any tests.
- **3** Displays any attention or error conditions detected in any tests.
- **4** Silent mode. Displays errors detected during the tests.

-m Displays a detailed description of the command and its purpose.

-s Displays a summary of the purpose for the command.

Parameters

hostname The host name of a remote system.

address The network address of a remote system.

Security

Permissions on the **ctsvhbar** command permit members of the **bin** user group to execute this command.

Exit Status

Exit status conforms to the CSM Probe Infrastructure conventions.

0 No problems detected. Any messages displayed are informational. No administration intervention is required.

10 No problems were detected. The command was unable to resolve the host name or IP address provided by the command user. The command user should verify that the correct host name or IP address was used. If the correct name or address was used, the system administrator should verify that the host-name resolution scheme used by the local system permits that name or address to be resolved.

127 Unexpected failure in this command.

Restrictions

- Cluster security services supports its own host identifier format and trusted host list file format only.
- Trusted host lists are modifiable using this command only.
- Cluster security services does not provide an automated utility for creating, managing, and maintaining trusted host lists throughout the cluster. This is a procedure left to either the system administrator or the cluster management software.

Standard Output

When the **-h** flag is specified, this command's usage statement is written to standard output. When the **-l** flag is specified, the contents of the trusted host list file are written to standard output.

Standard Error

Descriptive information for any detected failure condition is written to standard error.

Examples

To return the host name that the HBA security mechanism would use on the local node to verify credentials from the host identified by the host name **zathras**, you would enter:

```
ctsvhbar zathras
```

The output would look like this:

```
Host name or network address: zathras
Fully qualified host name
     used for authentication: zathras.ibm.com
```

To return the host name that the HBA security mechanism would use on the local node to verify credentials from the host identified by the network address **9.127.100.101**, you would enter:

```
ctsvhbar 9.127.100.101
```

The output would look like this:

```
Host name or network address: 9.127.100.101
Fully qualified host name
     used for authentication: epsilon3.pok.ibm.com
```

To return the host name that the HBA security mechanism would use on the local node to verify credentials from both the host identified by the host name **zathras**, and the host identified by the network address **9.127.100.101**, you would enter:

```
ctsvhbar zathras 9.127.100.101
```

The output would look like this:

```
Host name or network address: zathras
Fully qualified host name
     used for authentication: zathras.ibm.com
Host name or network address: 9.127.100.101
Fully qualified host name
     used for authentication: epsilon3.ibm.com
```

Location

/usr/sbin/rsct/bin/ctsvhbar	Contains the **ctsvhbar** command

Files

/usr/sbin/rsct/cfg/ctcasd.cfg	Default configuration for the **ctcasd** daemon
/var/ct/cfg/ctcasd.cfg	Configuration for the **ctcasd** daemon, which can be modified by the system administrator

Related Information

Commands: **ctskeygen**, **ctsthl**, **ctsvhbac**, **ctsvhbal**

Books: See the cluster security topics in *RSCT Administration Guide*.

cu Command

Purpose

Connects directly or indirectly to another system.

Syntax

To Establish a Connection Using a Modem

cu [**-d**] [**-h**] [**-m**] [**-T***Seconds*] [**-n**] [**-s***Speed*] [**-t**] [**-e** | **-o**] *TelephoneNumber*

To Specify the Name of a Device for a Connection

cu [**-d**] [**-h**] [**-m**] [**-T***Seconds*] [**-s***Speed*] [**-e** | **-o**] **-l***Line*

To Specify a System Name for a Connection

cu [**-d**] [**-h**] [**-m**] [**-T***Seconds*] [**-e** | **-o**] *SystemName*

Description

The **cu** command is a Basic Networking Utilities (BNU) command that connects one system to a terminal connected to either a UNIX or non-UNIX system. The connection can be established over a hardwired line or over a telephone line using a modem.

Once the connection is established, a user can be logged in on both systems at the same time, executing commands on either one without dropping the BNU communication link. If the remote computer is also running under UNIX, the user can transfer ASCII files between the two systems.

After issuing the **cu** command from the local system, the user must press the Enter key and then log in to the remote system. After making the connection, the **cu** command runs as two concurrent processes: the transmit process reads data from standard input and, except for lines beginning with a ~ (tilde), passes that data to the remote terminal.

The receive process accepts data from the remote system and, except for lines beginning with a ~, passes it to standard output. Internally, the program accomplishes this by initiating an output diversion to a file on the local system when a line from the remote system begins with ~> (tilde, greater than). The trailing ~> marks the end of the diversion. To control input from the remote system so the buffer is not overrun, the **cu** command uses an automatic **DC3/DC1** (Ctrl-Q/Ctrl-S) protocol.

The **cu** command can be used to connect multiple systems, and commands can then be executed on any of the connected systems. For example, the user can issue the **cu** command on system X to connect to system Y, and then issue the **cu** command on system Y to connect to system Z. System X is then the local computer, and systems Y and Z are remote computers.

The user can execute commands on system Z by logging in and issuing the command. Commands can be executed on system X by prefixing the command with a single tilde (**~***Command*) and on system Y by prefixing the command with two tildes (**~~***Command*). In general, one tilde causes the specified command to be executed on the original local computer, and two tildes cause the command to be executed on the next system on which the **cu** command was issued.

For example, once the multiple systems are connected, the user can execute the **uname -n** command (to display the node name) on systems Z, X, and Y as follows:

```
$ uname -n
Z
$ ~!uname -n
X
$ ~~!uname -n
Y
```

Notes:

1. The **cu** command does not do integrity checking on data it transfers.
2. Data fields with special **cu** characters may not be transmitted properly.
3. The exit code is 0 for normal exit, otherwise, -1.

In addition to issuing regular commands on the remote system, the user can issue special **cu** command subcommands, which are preceded by a ~ (tilde). Use these subcommands to issue commands on the local system and to perform tasks such as transferring files between two UNIX systems. As soon as the user enters the **~!**, **~$**, **~%**, **~l**, or **~t** subcommand, the system displays the name of the local computer in a format similar to the following:

```
~[SystemName]/%
```

The user then enters the subcommand to be executed on the local computer.

Flags

-d Prints diagnostic traces.

-e Designates that even parity is to be generated for data sent to the remote system.

-h Emulates local echo, supporting calls to other systems that expect terminals to be set to half-duplex mode.

-l*Line* Specifies the name of a device to be used as the line of communication between the local and the remote system. This can be used to override the search that would otherwise take place for the first available line with the right speed. When the **-l** flag is used without the **-s** flag, the speed of the *Line* is taken from the **Devices** file(s) (by default, the **/etc/uucp/Devices** file).

When the **-l** and **-s** flags are used together, the **cu** command searches the **Devices** file(s) to check whether the requested speed is available for the specified line. If so, the connection is made at the requested speed; otherwise, an error message is printed, and the call is not made.

The specified device is generally a hardwired asynchronous line (for example, **/dev/tty2**), in which case the *TelephoneNumber* parameter is not required. If the specified device is associated with a modem, a telephone number must be provided. Using this flag with the *SystemName* parameter rather than with *TelephoneNumber* parameter does not give the desired result.

Under ordinary circumstances, the user should not have to specify the transmission speed or a line or device. The defaults set when BNU is installed should be sufficient.

-m Instructs the **cu** command to ignore modem control signal data carrier detect (DCD).

-n For added security, prompts the user to provide the telephone number to be dialed, rather than taking it from the command line.

-o Designates that odd parity is to be generated for data sent to the remote system.

-s*Speed* Specifies the rate at which data is transmitted to the remote system (300, 1200, 2400, 4800, 9600, or 19200 baud).The default value is **Any** speed, which instructs the system to use the rate appropriate for the default (or specified) transmission line. The order of the transmission lines is specified in the BNU **Devices** file(s) (by default, the **/etc/uucp/Devices** file). Most modems operate at 300, 1200, or 2400 baud, while most hardwired lines are set to 1200 baud or higher. When transferring data such as a file between a local and a remote system, a speed of 300 baud may occasionally be needed. The lower baud rate results in less interference on the line.

-t Used to dial an ASCII terminal that has been set to autoanswer. Appropriate mapping of carriage-return to carriage-return line feed pairs is set.

-T*Seconds* Specifies the maximum number of seconds to wait before timing out. The default is 45 seconds.

> **Note:** You can also enter `WAIT=n` before any send string in the **Dialers** file. Where `n` is the number of seconds to wait before timing out.

Parameters

SystemName — The name of the remote system, recognized by BNU, with which a connection is established. A system name can be used rather than a telephone number; in that case, the **cu** command obtains an appropriate hardwired line or telephone number from the BNU **Systems** file(s) (by default, the **/etc/uucp/Systems** file). System names must be ASCII characters only.

> **Note:** Do not use the *SystemName* flag with the **-l** flag and the **-s** flag. If you do, the **cu** command connects to the first available line for the requested system name, ignoring the specified line and speed.

TelephoneNumber — The telephone number used to establish a remote connection using a modem. This entry can be either a local or a long-distance telephone number.

Subcommands

The **cu** command transmit process interprets lines beginning with a **~** (tilde) in the following ways:

~! — Returns the user to an interactive shell on the local system. Toggle between the local and remote systems using **~!** (remote to local) and Ctrl-D (local to remote).

~%break — Transmits a break sequence to the remote system. The break can also be specified as **~%b**.

~%cd *DirectoryName* — Changes the directory on the local system from the current directory to the directory specified by the *DirectoryName* variable.

~%debug — Toggles the **-debug** flag on or off; this can also be specified as **~%d**.

~%nostop — Toggles between **DC3/DC1** input control protocol and no input control. This is useful in case the remote system is one that does not respond properly to the **DC3** and **DC1** characters.

~%put *From* [*To*] — Copies the *From* file on the local system to the *To* file on the remote system. If the *To* variable is omitted, the local file is copied to the remote system under the same file name. As each block of the file is transferred, consecutive single digits are displayed on the terminal screen. Only ASCII files can be transferred using this subcommand.

The use of the **~%put** subcommand requires the **stty** command and the **cat** command on the remote system. It also requires that the current erase and kill characters on the remote system be identical to these current control characters on the local system. Backslashes are inserted at appropriate places in the transmitted data. There is an artificial slowing of transmission by the **cu** command during the **~%put** operation so that loss of data is unlikely.

~%take *From* [*To*] — Copies the *From* file on the remote system to the *To* file on the local system. If the *To* variable is omitted, the remote file is copied to the local system under the same file name. As each block of the file is transferred, consecutive single digits are displayed on the terminal screen. Only ASCII files can be transferred using this subcommand. The use of the **~%take** subcommand requires the **echo** command and the **cat** command on the remote system. Also, **stty tabs** mode should be set on the remote system if tabs are to be copied without expansion to spaces.

~. — Logs the user off the remote computer and then terminates the remote connection. Usually the connection terminates when you log off the remote computer. However, with some types of interconnection hardware, it may be necessary to use a **~.** to terminate the conversation after the normal logoff sequence has been used.

~!*Command* — Executes, on the local system, the command denoted by the *Command* variable.

~$*Command* — Runs, on the local system, the command denoted by the *Command* variable, then sends the command's output to the remote system for execution.

~l — Prints the values of the **TERMIO** structure variables for the remote communication line. This is useful for debugging.

~t	Prints the values of the **TERMIO** structure variables for the user's terminal. This is useful for debugging.
~~*String*	Sends the string denoted by the *String* variable to the remote system.

Examples

The following are examples of connecting to a remote system.

1. To connect to a remote system, enter:

   ```
   cu venus
   ```

   In this example, you are connected to the remote system `venus`. System `venus` must be listed in one of the local **Systems** files (by default, the **/etc/uucp/Systems** file or one of the **Systems** files listed for the **cu** command in the **/etc/uucp/Sysfiles** file).
2. To dial a remote system and set the baud rate, enter:

   ```
   cu  -s1200 9=12015558391
   ```

   In this example, you dial a remote system whose telephone number is `1-201-555-8391`, where dialing `9` is required to get an outside dial tone. The baud rate is set to `1200`.
3. To log in to a system connected by a hardwired line asynchronous line, enter:

   ```
   cu  -l /dev/tty2
   ```

   The **cu** command contacts the system connected to the `tty2` device.
4. To dial a remote system with a specified line and a specific speed, enter:

   ```
   cu  -s 1200  -l tty3
   ```

   The command contacts the system connected to the `tty3` device, using a speed of `1200` baud.
5. To dial a remote system using a specific line associated with a modem, enter:

   ```
   cu  -l cul4 9=12015558391
   ```

   In this example, you dial a remote system whose telephone number is `1-201-555-8391`, where dialing `9` is required to get an outside dial tone. The **cu** command uses the modem connected to the `cul4` device.

1. To display the contents of a file after logging in to the remote system, enter:

   ```
   ~!pg /usr/msg/memos/file10
   ```

   The **~!** subcommand executes the **pg** command on the local system, displaying the contents of the `file10` file in the `/usr/msg/memos` directory on the local system.
2. To copy a file from the local system to the remote system without changing the name of the file, enter:

   ```
   ~%put /home/amy/file
   ```

   The `/home/amy/file` file is copied from the local system to the remote system without changing the name of the file.
3. To copy a file from the local system to the remote system and change the file name, enter:

   ```
   ~%put /home/amy/file /home/amy/tmpfile
   ```

The /home/amy/file file is copied from the local system to the remote system and the file name changed to /home/amy/tmpfile.

4. To copy a file from the remote system to the local system without changing the name of the file, enter:

```
~%take /home/jeanne/test1
```

The /home/jeanne/test1 file is copied from the remote system to the local system without changing the name of the file.

5. To copy a file from the remote system to the local system and change the file name, enter:

```
~%take /home/jeanne/test1 /usr/dev/jeanne/tmptest
```

In this example, the /home/jeanne/test1 file is copied from the remote system to the local system and the file name changed to /usr/dev/jeanne/tmptest.

Files

/etc/locks	Prevents multiple use of device.
/usr/bin/cu	Specifies the path name of the **cu** command.
/bin/cu	Specifies a symbolic link to the **/usr/bin/cu** command.
/etc/uucp/Devices	Contains information about available links.
/etc/uucp/Dialcodes	Contains dialing code abbreviations.
/etc/uucp/Dialers	Controls initial handshaking on a link.
/etc/uucp/Permissions	Contains access permission codes.
/etc/uucp/Systems	Lists accessible remote systems.
/etc/uucp/Sysfiles	Specifies alternate files to be used as **Systems**, **Devices**, and **Dialers** files.

Related Information

The **cat** command, **ct** command, **echo** command, **rmail** command, **stty** command, **uname** command, **uucp** command, **uuname** command, **uupick** command, **uustat** command, **uuto** command, **uux** command.

---

curt Command

Purpose

Generates CPU utilization report from a trace.

Syntax

curt -i *inputfile* [-o *outputfile*] [-n *gensymsfile*] [-m *trcnmfile*] [-a *pidnamefile*] [-f *timestamp*] [-l *timestamp*] [**-r** PURR] [-ehpstP] [-@ {ALL | *WparList*}]

Description

The **curt** command takes an AIX trace file as input and produces a number of statistics related to processor (CPU) utilization and process/thread/pthread activity. The command will work with both uniprocessor and multiprocessor AIX traces if the processor clocks are properly synchronized.

The AIX trace file which is gathered using the **trace** command should contain at least the trace events (trace hooks) listed below. These are the events **curt** looks at to calculate its statistics:

```
HKWD_KERN_SVC, HKWD_KERN_SYSCRET, HKWD_KERN_FLIH, HKWD_KERN_SLIH,
HKWD_KERN_SLIHRET, HKWD_KERN_DISPATCH, HKWD_KERN_RESUME, HKWD_KERN_IDLE,
HKWD_SYSC_FORK, HKWD_SYSC_EXECVE, HKWD_KERN_PIDSIG, HKWD_SYSC__EXIT
```

```
HKWD_SYSC_CRTHREAD, HKWD_KERN_INITP, HKWD_NFS_DISPATCH, HKWD_CPU_PREEMPT,
HKWD_DR, HKWD_KERN_PHANTOM_EXTINT, HKWD_RFS4_VOPS, HHKWD_RFS4_VFSOPS, HKWD_RFS4_MISCOPS, HKWD_RFS4,
HKWD_KERN_HCALL, HKWD_WPAR,
HKWD_PTHREAD_VPSLEEP, HKWD_PTHREAD_GENERAL
```

This means that, if you specify the **-j** flag on your **trace** command, you must include these numbers for **curt**:

```
-j 100,101,102,103,104,106,10C,119,134,135,139,200,210,215,38F,419,465,47F,488,489,48A,48D,492,4C9,605,609
```

Or, you can use **-J curt** instead.

To get the PTHREAD hooks into the trace, you must execute your **pthread** application using the instrumented **libpthreads.a**. One way to cause that to happen is to perform the following three steps before starting your application (KornShell syntax):

1. mkdir /temp.lib; cd /temp.lib
2. ln -s /usr/ccs/lib/perf/libpthreads.a
3. export LIBPATH=$PWD:$LIBPATH

Putting the instrumented library directory in LIBPATH is necessary to activate the user **pthread** instrumentation; the **temp.lib** directory can be put anywhere.

Flags

-i *inputfile* Specifies the input AIX trace file to be analyzed.

-o *outputfile* Specifies the output file (default is stdout).

-n *gensymsfile* Specifies a names file produced by **gensyms**.

-m *trcnmfile* Specifies a names file produced by **trcnm**.

-a *pidnamefile* Specifies a PID to process name mapping file.

-f *timestamp* Starts processing trace at *timestamp* seconds.

-l *timestamp* Stops processing trace at *timestamp* seconds.

-r PURR Uses the PURR register to calculate CPU times.

-e Outputs elapsed time information for system calls and pthread calls.

-h Displays usage text (this information).

-p Outputs detailed process information.

-s Outputs information about errors returned by system calls.

-t Outputs detailed thread information.

-P Outputs detailed pthread information.

-@ Controls the addition of workload partition information to a curt report.

You can use the **-@** flag in one of the following forms:

-@ Outputs a summary of workload partitions. The summary includes the processor usage for workload partitions in various execution modes. In addition, WPAR names are shown for listed processes summarizing the processor usage by processes, threads, or pthreads.

-@ All Outputs reports for the system and all of the workload partitions. The reports are delimited by three lines containing WPAR names or SYSTEM for the overall system.

-@ *WparList*
Outputs reports for the workload partitions specified by the *WparList* parameter, which is a comma-separated list of WPAR names. The reports are delimited by three lines containing WPAR names.

If the **trace** process name table is not accurate, or if more descriptive names are desired, use the **-a** flag to specify a PID to process name mapping file. This is a file with lines consisting of a process ID (in decimal) followed by a space followed by an ASCII string to use as the name for that process.

If the input AIX-trace file is created with the **-n** flag specified, curt will use that address/name table to resolve System Call and Slih addresses to names *if* you do not specify a **-m** or a **-n** flag on the curt command line.

If the input AIX-trace file is created in a workload partition, the **curt** command prints a WPAR report. The **-@** flag is not allowed in this case.

Report Contents

The curt report includes the following information:

curt and AIX Trace Information

The first lines in the curt report give the time when the curt program was executed and the command line used to invoke curt. Following that is this information about the AIX trace file being processed by **curt**: name, size, creation date, and the command used to gather the trace file.

The line `PURR was used to calculate CPU times` is printed if the **-r** PURR option was used and the trace file includes the PURR register.

System Summary

The first major section of the report is the System Summary. This section describes the time spent by the system as a whole (all processors) in various execution modes. These modes are as follows:

APPLICATION
The sum of times spent by all processors in User (non-privileged) mode.

SYSCALL
The sum of times spent by all processors doing System Calls. This is the portion of time that a processor spends executing in the kernel code providing services directly requested by a user process.

HCALL
The sum of times spent by all processors doing Hypervisors Calls. This is the portion of time that a processor spends executing in the hypervisor code providing services directly requested by the kernel.

KPROC
The sum of times spent by all processors executing kernel processes other than the IDLE process and NFS processes. This is the portion of time that a processor spends executing specially created dispatchable processes which only execute kernel code.

NFS The sum of times spent by all processors executing NFS operations. NFS operations begin with RFS_DISPATCH_ENTRY and end with RFS_DISPATCH_EXIT subhooks for NFS V2/V3. NFS operations begin with `start` and end with `done` or `done error` for NFS V4.

FLIH The sum of times spent by all processors in FLIHs (first level interrupt handlers).

SLIH The sum of times spent by all processors in SLIHs (second level interrupt handlers).

DISPATCH
The sum of times spent by all processors in the AIX dispatch code. This sum includes the time spent in dispatching all threads (i.e. it includes the dispatches of the IDLE process).

IDLE DISPATCH
The sum of times spent by all processors in the AIX dispatch code where the process being dispatched was the IDLE process. Because the DISPATCH category includes the IDLE DISPATCH category's time, the IDLE DISPATCH category's time is not separately added to calculate either CPU(s) busy time or TOTAL (see below).

CPU(s) busy time
The sum of times spent by all processors executing in application, syscall, kproc, flih, slih, and dispatch modes.

IDLE The sum of times spent by all processors executing the IDLE process.

TOTAL
The sum of CPU(s) busy time and IDLE. This number is referred to as ″total processing time.″

The column labeled `processing total time (msec)` gives the total time (in milliseconds) for the corresponding processing category. The column labeled `percent total time` gives the processing total time as a percentage of the TOTAL processing total time. The column labeled `percent busy time` gives the processing total time as a percentage of the CPU(s) busy time processing total time. The `Avg. Thread Affinity` is the probability that a thread was dispatched to the same processor that it last executed on.

The Total Physical CPU time (msec) is the real time the CPU(s) were running (not preempted). The Physical CPU percentage gives the Physical CPU(s) Time as a percentage of total time.

Note: In a WPAR report, the system summary information is labeled ″WPAR summary″.

System Application Summary

Following the System Summary is the System Application Summary, which describes the time spent in User mode in details. This section describes the time spent by all processes (on all processors) executing various parts of libpthreads.

PTHREAD
The sum of times spent by all pthreads in traced libpthreads operations.

PDISPATCH
The sum of times spent by all pthreads in the libpthreads dispatch code.

PIDLE The sum of times spent by all pthreads in libpthreads vp_sleep code.

OTHER
The sum of time spent by all threads in user mode outside traced libpthreads operations.

APPLICATION time
The sum of times spent by all processors in user mode.

The column labeled `processing total time (msec)` gives the total time in milliseconds for the corresponding processing category. The column labeled `percent total time` gives the processing total time as a percentage of the TOTAL processing total time of System Summary. The column labeled `percent application time` gives the processing total time as a percentage of the APPLICATION processing total time. The `Avg. Pthread Affinity` is the probability that a pthread was dispatched to the same thread that it last executed on.

Note: In a WPAR report, the system application summary information is labeled ″WPAR application summary″.

WPARs Summary

The WPARs Summary of the report is generated when you specify the **-@** flag. The following system and system application information for workload partitions, shown as column headings in the summary, describes the time spent in all of the workload partitions in details:

appli Percent of the total process time that was spent by the WPAR in user mode (non-privileged).

syscall
Percent of the total process time that was spent by the WPAR performing system calls.

hcall Percent of the total process time that was spent by the WPAR performing hypervisor calls.

kproc Percent of the total process time that was spent by the WPAR running kernel processes calls.

nfs Percent of the total process time that was spent by the WPAR running NFS operations.

flih Percent of the total process time that was spent by the WPAR in the first-level interrupt handlers.

slih Percent of the total process time that was spent by the WPAR in the second-level interrupt handlers.

total Percent of the total process time that was spent by the WPAR.

total(msec)
The sum of processor time, in milliseconds, used by the WPAR.

WPAR The WPAR name.

Note: The WPARs Summary is generated only in an overall system report.

Per Processor Summary

Following the System Application Summary is the Per Processor Summary, which is essentially the same information but broken down on a processor by processor basis. In the description given for the System Summary, the phrase ″sum of times spent by all processors″ can be replaced by ″time spent by this processor″. The Total number of process dispatches refers to how many times AIX dispatched any non-IDLE process on this processor, while Total number of idle dispatches gives the count of IDLE process dispatches.

The `Total Physical CPU time (msec)` is the real time the processor was running (not preempted). The `Physical CPU percentage` gives the Physical CPU Time as a percentage of total time.

`Physical processor affinity` is the probability that a logical processor was dispatched on the same physical processor that it last executed on. Total number of preemptions is the number of times the virtual processor was redispatched on a physical CPU.

`Total number of H_CEDE` is the number of H_CEDE hypervisor call done by this processor; `with preeemption` indicates the number of H_CEDE calls resulting in preemption.

`Total number of H_CONFER` is the number of H_CONFER hypervisor call done by this processor; `with preeemption` indicates the number of H_CONFER calls resulting in preemption.

Note: A per processor summary is not generated in a WPAR report.

Per Processor Application Summary

Following each Processor Summary is the Per Processor Application Summary, which is essentially the same information as System Application Summary but broken down on a processor by processor basis.

The `Total number of pthread dispatches` refers to how many times libpthreads dispatched any pthread on this processor, while `Total number of pthread idle dispatches` gives the count of calls to vp_sleep.

Note: A per processor application summary is not generated in a WPAR report.

Application Summary

The second major section of the report is the Application Summary. The first part of this section summarizes the total system processing time on a per-thread basis (by Tid). For each thread, identified by Process ID (and name if available) and Thread ID, the summary gives the total application (same as APPLICATION above) and syscall (same as SYSCALL above) processing time in milliseconds and as the percentage of the total system processing time for all processors in the trace. In addition, the summary gives the sum of those two times, both as raw time, and as a percentage of the total processing time.

The second part of this section gives the same information on a per-process ID (by Pid) basis. The third part of this section gives the same information on a per-process name (by process type) basis.

The fourth part of this section gives similar information for kernel process threads (Kproc Summary). Since most kprocs provide a specific kernel service, the total processing time is split into two categories, operation and kernel, which loosely correspond to syscall and application for a process which always runs

in kernel code. Each kproc thread is identified by name, Process ID, Thread ID and type of kproc if known. The kproc types are listed and described in a table immediately following this summary.

The fifth part of this section is the Pthread Process Summary. This section gives the total application time on multi-threaded Process (by Pid). For each process, identified by Process ID (and name if available), the summary gives the total application, pthread and other processing time in milliseconds and as the percentage of the total application time for all processors in the trace.

All five sections of the Summary are presented in sorted order from most combined processing time to least.

In all five sections of an Application Summary, the WPAR name is added to identify the thread or process if you specify the **-@** flag.

Note: Pids and Tids (Process and Thread IDs) are always given in decimal.

System Calls Summary

The third major section of the report is the System Calls Summary. This section summarizes the processing time spent in system calls. For each system call (SVC), identified by kernel address (and name if available), the summary gives the number of times the SVC was called and the total processor time for all calls in milliseconds and as a percentage of total system processing time for all processors in the trace. In addition, the summary gives the average, minimum and maximum times for one call to the SVC. If the **-e** flag is specified, the summary gives the total elapsed time for all calls to the SVC and the average, minimum and maximum elapsed times for one call. Elapsed time is the wall-clock time from when the process starts executing the SVC in kernel mode until the process resumes executing in application mode. The Summary is presented in sorted order from most total processor time to least. If the **-s** flag is specified, the summary gives the number of times each error code (errno) was returned by each System Call.

The second part of this section is the Pending System Calls Summary. This part lists the System Calls which have started but not completed. The time that is given is included in the SYSCALL time for the system and the various processors and is included in the syscall time for the pthread, thread and process which issued the SVC, but is not included in the processing time for the system call in the first part of this section. The pending call is also not included in the count given in the first part of this section.

Notes:

1. System call addresses are always given in hexadecimal. Pids and Tids are always given in decimal.
2. WPAR names are added in a System Calls Summary to identify threads or processes if you specify the **-@** flag.

System Hypervisor Calls Summary

If there is hypervisor activity in the trace, an additional section is inserted at this point of the report. This major section of the report is called `Hypervisor Calls Summary`. This section summarizes the processing time spent in hypervisor calls. For each Hypervisor call (HCALL), identified by name (and kernel address), the summary gives the number of times the HCALL was called and the total processor time for all calls in milliseconds and as a percentage of total system processing time for all processors in the trace. In addition, the summary gives the average, minimum and maximum times for one call to the HCALL. If the **-e** flag is specified, the summary gives the total elapsed time for all calls to the HCALL and the average, minimum, and maximum elapsed times for one call. Elapsed time is the wall-clock time between the start and end of an hypervisor call. The summary is presented in sorted order from most total processor time to least.

The second part of this section is called `Pending Hypervisor Calls Summary`. This part lists the Hypervisor Calls which have started but not completed. The time that is given is included in the HCALL time for the system and the various processors and is included in the hypervisor time for the pthread, thread, and

process which issued the HCALL, but is not included in the processing time for the hypervisor call in the first part of this section. The pending call is also not included in the count given in the first part of this section.

Notes:

1. Hypervisor call addresses are always given in hexadecimal. Pids and Tids are always given in decimal.
2. WPAR names are added in a System Hypervisor Calls Summary to identify the threads or processes if you specify the **-@** flag.

Pthread Calls Summary

The fourth major section of the report is the Pthread Calls Summary. This section summarizes the processing time spent in called pthread routines. For each pthread routine, identified by name, the summary gives the number of times the pthread routine was called and the total processor time for all calls, in milliseconds and as a percentage of total system processing time, for all processors in the trace. In addition, the summary gives the average, minimum and maximum times for one call to the pthread routine. If the **-e** flag is specified, the summary gives the total elapsed time for all calls to the pthread routine and the average, minimum and maximum elapsed times for one call. Elapsed time is the wall-clock time from when the process starts executing the pthread routine until the process exits the libpthreads code. The Summary is presented in sorted order from most total processor time to least.

The second part of this section is the Pending Pthread Calls Summary. This part lists the Pthread Calls which have started but not completed.

Note: WPAR names are added in a Pthread Calls Summary to identify threads or processes if you specify the **-@** flag.

System NFS Calls Summary

This major section of the report is the System NFS Calls Summary. This section summarizes the processing time spent in NFS operations. For each NFS operation, identified by operation name and NFS version, the summary gives the number of times the operation was called and the total processor time for all calls in milliseconds and as a percentage of total NFS operation time for all operations with the same NFS version. In addition, the summary gives the average, minimum and maximum times for one call to the operation. If the **-e** flag is specified, the summary gives the total elapsed time for all calls to the operation and the average, minimum and maximum times for one call. The total elapsed time is also given as a percentage of total NFS operation elapsed time for all operations with the same NFS version. Elapsed time is the wall-clock time from the operation dispatch entry hook until the operation dispatch exit hook. In all cases, the summary gives the count of operation calls as a percentage of total NFS operation calls for all operations with the same NFS version. The Summary is presented in numerical order of the operation codes. The operations are presented in order of NFS Version. For NFS V4, the server operations are listed before the client operations.

The System NFS Calls Summary is followed by the Pending NFS Calls Summary. This part lists the NFS calls which have started but not completed. The time that is given is included in the NFS time for the system and the various processors and is included in the operation time for the thread and process which issued the NFS call, but is not included in the processing time for the NFS operation in the first part of this section. The pending call is also not included in the count given in the first part of this section.

Note: WPAR names are added in a System NFS Calls Summary to identify threads or processes if you specify the **-@** flag.

Flih Summary

The fifth major section of the report is the Flih Summary. This section summarizes the amount of time spent in first level interrupt handlers (Flih). The first part of the summary gives the total number of entries to each Flih in the trace, as well as the total processor time for all executions of the Flih by all processors in milliseconds. In addition, the summary gives the average, minimum and maximum times for one execution. Each Flih is identified by a system-defined Flih type and a corresponding Flih name, if known.

The second part is the same information broken down on a processor by processor basis. It is possible that not all Flihs which occurred on the system will have occurred on each processor, so the Global Flih list may not be the same as the Flih list for each processor.

The second part of this section may include the Pending Flih Summary. This is a list of the Flihs which have started but not completed. The time that is given is included in the FLIH time for the system and the affected processor, but is not included in the processing time for the Flih in both parts of this section. The pending Flih is also not included in the counts given in both parts of this section.

Slih Summary

The fifth major section of the report is the Slih Summary. This section summarizes the amount of time spent in second level interrupt handlers (Slih). The first part of the summary gives the total number of entries to each Slih in the trace, as well as the total processor time for all executions of the Slih by all processors in milliseconds. In addition, the summary gives the average, minimum and maximum times for one execution. Each Slih is identified by kernel address and Slih function or module name, if known.

The second part is the same information broken down on a processor by processor basis. It is possible that not all Slihs which occurred on the system will have occurred on each processor, so the Global Slih list may not be the same as the Slih list for each processor.

The second part of this section may include the Pending Slih Summary. This is a list of the Slihs which have started but not completed. The time that is given is included in the SLIH time for the system and the affected processor, but is not included in the processing time for the Slih in both parts of this section. The pending Slih is also not included in the counts given in both parts of this section.

Detailed Process Information

This section of the report is produced when the **-p** flag is specified. It gives detailed information about each process found in the trace. This information is as follows:

- The Process ID (Pid) for that process, the process name if known, and the WPAR name if you specify the **-@** flag.
- A count and a list of the Thread IDs (Tids) for that process.
- A count and a list of Pthread IDs (Ptid) for that process, if any.
- The time spent in application (user) mode, system call mode, and hypervisor mode is shown. For kprocs, the time spent in kernel mode and operation mode is shown instead.
- The detail of time spent in application mode, time spent in pthread operations, time spent in libpthreads dispatch, and time spent in vp_sleep. This is printed only if there are any Ptids for the process.
- Information on what Pthread calls were made by pthreads of this process. For NFS kprocs, information on which NFS Calls were made by threads of this process is shown instead. The **-e** flag also affects this output.
- Information on what hypervisor calls were made by threads of this process. The **-e** flag also affects this output.
- Information on what system calls were made by threads of this process. The **-e** flag also affects this output.

The processes are presented in sorted order from most combined application and syscall processing time to least.

Detailed Thread Information

This section of the report is produced when the **-t** flag is specified. It gives detailed information about each thread found in the trace. This information is as follows:

- The Thread ID (Tid) and Process ID (Pid) for that thread, the process name if known, and the WPAR name if you specify the **-@** flag.
- The time spent in application (user) mode, system call mode, and hypervisor call mode is shown. For kprocs, the time spent in kernel mode and operation mode is shown instead.

- Information on which system calls were made by this thread, including information on errors returned by the system calls if the **-s** flag was specified. For NFS kproc threads, information on which NFS Calls were made by this thread is shown instead. The **-e** flag also affects this output.
- Information on which hypervisor calls were made by this thread. The **-e** flag also affects this output.
- The processor affinity is the probability that, for any dispatch of the thread, the thread was dispatched to the same processor that it last executed on.
- The Dispatch Histogram shows the number of times the thread was dispatched to each CPU in the system.
- The total number of times the thread was dispatched (not including redispatches described in 7 below).
- The number of redispatches due to interrupts being disabled indicates that the same thread which just ran was dispatched again because that thread has set the interrupt mask to INTMAX. This is shown only if nonzero.
- The average dispatch wait time is the average elapsed time since the thread was last undispatched (i.e. average elapsed time since the thread last stopped executing).
- How many times each type of Flih occurred while this thread was executing. Some of these types may be caused by the thread (such as DSI or ISI) while other types (such as IO) are can occur when this thread just happens to be running and are not necessarily caused by the thread itself.

The threads are presented in sorted order from most combined application and syscall processing time to least.

Detailed Pthread Information

This section of the report is produced when the **-P** flag is specified. It gives detailed information about each pthread found in the trace. This information is as follows:

- The Pthread ID (Ptid) and Process ID (Pid) for that pthread, the process name if known, and the WPAR name if you specify the **-@** flag.
- The time spent in application (user) mode, kernel mode, and hypervisor mode is shown.
- Application time detail: time spent in pthread calls, pthread dispatch, vp_sleep (pthread idle), and other application time.
- Information on what system calls were made by this pthread, including information on errors returned by the system calls if the **-s** flag was specified. The **-e** flag also affects this output.
- Information on what hypervisor calls were made by this pthread. The **-e** flag also affects this output.
- Information on what Pthread calls were made by this pthread. The **-e** flag also affects this output.
- The processor affinity is the probability that, for any dispatch of the pthread, the pthread was dispatched to the same processor that it last executed on.
- The Dispatch Histogram for thread shows the number of times the pthread was dispatched to each CPU in the system.
- The total number of times the pthread was dispatched (not including redispatches described in 9 below).
- The number of redispatches due to interrupts being disabled indicates that the same pthread which just ran was dispatched again because that pthread has set the interrupt mask to INTMAX. This is shown only if non-zero.
- The average dispatch wait time is the average elapsed time since the pthread was last undispatched by the kernel dispatcher (that is, average elapsed time since the pthread last stopped executing).
- The thread affinity is the probability that, for any dispatch of the pthread, the pthread was dispatched to the same thread that it last executed on.
- The Dispatch Histogram for pthread shows the number of times the pthread was dispatched to each thread in the system.
- The total number of times the pthread was dispatched in libpthreads.
- The average dispatch wait time is the average elapsed time since the thread was last undispatched by the libpthreads dispatcher (that is, the average elapsed time since the thread last stopped executing).

- How many times each type of Flih occurred while this thread was executing. Some of these types may be caused by the thread (such as DSI or ISI) while other types (such as IO) are can occur when this thread just happens to be running and are not necessarily caused by the thread itself.

The pthreads are presented sorted by Pid-Ptid.

Files

/usr/bin/curt Contains the **curt** command. Located in the **bos.perf.tools** fileset.

Related Information

AIX 5L Version 5.3 Performance Tools Guide and Reference

Performance management

---

custom Command

Purpose

Enables users to customize X applications.

Syntax

custom [**-h** | **-e** *Browser* | [**-s** *ResourceFile*] [*Application*]]

Description

The **custom** command starts the customizing tool, which is used to customize various aspects of applications.

The customizing tool can change the look of an application. It provides a user-friendly way to add resource values to your **.Xdefaults** file. *Resources* are customizable items such as colors, fonts, and other attributes that allow you to customize resources of a client application. Each application has its own set of unique resources, which are listed in an **app-custom** file. The customizing tool describes the resources available for modification for an application and the possible resource values you can select.

Flags

-h	Provides command line help.
-e *Browser*	Calls one of the standalone browsers. Valid values for *Browser* are **color**, **font**, **cursor**, and **picture**.
-s *ResourceFile*	Specifies the resource file from which to load and save resource settings. If the **-s** flag is not specified, the default is to load the values from the resource database stored in the **RESOURCE_MANAGER** property on the X server. If this database does not exist, then **$HOME/.Xdefaults** is loaded.

Most standard X Toolkit command-line options are understood by the **custom** command.The following table lists the standard command-line options:

Standard Command-Line Options in **custom** *command*

Option	Information
-bg	**Resource** *background **Value** Next argument **Sets** Background color
-background	**Resource** *background **Value** Next argument **Sets** Background color
-bd[1]	**Resource** *borderColor **Value** Next argument **Sets** Border color
-bordercolor[1]	**Resource** *borderColor **Value** Next argument **Sets** Color of border
-bw	**Resource** .borderWidth **Value** Next argument **Sets** Width of border in pixels
-borderWidth	**Resource** .borderWidth **Value** Next argument **Sets** Width of border in pixels
-display	**Resource** .display **Value** Next argument **Sets** Server to use
-fn[2]	**Resource** *font **Value** Next argument **Sets** Font name
-font[2]	**Resource** *font **Value** Next argument **Sets** Font name

Standard Command-Line Options in **custom** *command*

Option	Information
-fg	**Resource** *foreground **Value** Next argument **Sets** Foreground color
-foreground	**Resource** *foreground **Value** Next argument **Sets** Foreground color
-geometry	**Resource** .geometry **Value** Next argument **Sets** Size and position
-iconic	**Resource** .iconic **Value** On **Sets** Start as an icon
-name	**Resource** .name **Value** Next argument **Sets** Name of application
-reverse	**Resource** *reverseVideo **Value** On **Sets** Reverse video
-rv	**Resource** *reverseVideo **Value** On **Sets** Reverse video
+rv	**Resource** *reverseVideo **Value** Off **Sets** No Reverse video
-selection- Timeout	**Resource** .selection-Timeout **Value** Next argument **Sets** Selection timeout

Standard Command-Line Options in **custom** *command*

Option	Information
-synchronous	**Resource** *synchronous **Value** On **Sets** Synchronous debug mode
+synchronous	**Resource** *synchronous **Value** Off **Sets** Synchronous debug mode
-title	**Resource** .title **Value** Next argument **Sets** Title of application
-xrm	**Resource** value of argument **Value** Next argument **Sets** Depends on argument
-xnllanguage	**Resource** .xnlLanguage **Value** Next argument **Sets** Locale

Notes:

1. These options often have no visible effect on AIXwindows applications if the AIXwindows Window Manager is running.
2. Motif applications do not generally respond to these options.
3. Resources beginning with an* (asterisk) set the resource of every widget in the application to the same value.
4. Resources that begin with a . (period) set the resources of only the application's top-level Shell widget.

Parameters

Application Specifies the name or class of the application to customize.

Examples

1. To start the customizing tool and use prompts to choose the application to customize, type the following:

   ```
   custom
   ```

2. To start the customizing tool to modify the **app-defaults** file of the **xcalc** application, type the following:

   ```
   custom -s
   /usr/lib/X11/app-defaults/XCalc xcalc
   ```

Resources

The customizing tool has the following application resources:

listOfApps

This resource is used to display the application names on the starting dialog. The application name and corresponding **app-custom** file must be listed in pairs with the following syntax:

```
Application:app-custom
[,Application:app-custom]...
```

For example:

```
Custom.listOfApps:
xclock:XClock,custom:Custom
```

You can specify a maximum of 100 applications.

colorEditor*rgbtxtPath

This resource specifies the full path name of the **rgb.txt** file that the X server uses to define named colors. The default value is **/usr/lib/X11/rgb.txt**, which is correct for an X server running on a display that is directly attached to your system.

windowSearchDepth

The customizing tool must determine the top-level shell window of the application. It starts with the root window and conducts a recursive search to a depth of three windows by default. This default can be changed using the **windowSearchDepth** resource.

timeout

The Instant Changes button is grayed out until communication with the application is established. The amount of time to wait for the application to contact the customizing tool is controlled by the **Custom*timeout** resource.

resourceFile

The resource file is where your resource changes are saved. The default is **$HOME/.Xdefaults**. The **-s** flag allows the user to override this value.

appCustomPath

This resource specifies where the customizing tool is to look for the **app-custom** file. The **appCustomPath** string consists of a series of possible file names separated by colons. Within each name, the following values can be substituted:

%N Name of the **app-custom** file (usually the same as the class name of the application).

%T ″**app-custom**″

%L Locale in which **custom** is running.

%l Language part of the locale.

%t Territory part of the locale.

%c Codeset part of the locale.

%: A : (colon).

%% A % (percent sign).

$*envvar*
Value of the named environment variable.

${*envvar***}**
Value of the named environment variable.

$$ A $ (dollar sign).

The default value of **appCustomPath** is as follows:

```
$HOME/%L/%T/%N:\
$HOME/%T/%N:\
/usr/lib/X11/%L/%T/%N:\
/usr/lib/X11/%T/%N
```

topEditHighlight, **bottomEditHighlight**, **foregroundEditHighlight**, **backgroundEditHighlight**

The Browser button is highlighted when a browser is called and unhighlighted when a browser is canceled. These resources set the highlight color for the top shadow, bottom shadow, foreground, and background of the Browser button.

pictureEditor*editor

You can edit the bitmap or pixmap by pressing the Edit Picture button on the Pictures browser window. The editor is a separate application that exists on your system. It is called on your behalf. The **Custom*pictureEditor*editor** resource determines which editor commands to choose from. This resource accepts a list of commands separated by \n's (backslash 'n's). The first command that identifies an existing program that the user has permission to execute is used. The file name in the Chosen Picture text field is passed as a parameter to the editor when it is invoked. The default setting for this resource is:

```
Custom*pictureEditor*editor:
/usr/dt/bin/dticon  -f \n
/usr/lib/X11/bitmap
```

Note: The default editor, **/usr/dt/bin/dticon** only exists if the Common Desktop Environment (CDE) is installed. It edits both bitmaps (monochrome images) and pixmaps (color images). The **dticon** command accepts bitmaps stored in either the X Pixmap Version 2 Enhanced (XPM2) format which was used by the X Desktop (**xdt**) application shipped in AIXwindows Version 1.2.5, or X Pixmap Version 3 (XPM3) - a new XPG3 compliant format used by CDE. However, it requires pixmap images be stored in the XPM3 format. CDE has documented tools that can convert pixmaps from the XPM2 to the XPM3 format.

The **/usr/bin/X11/bitmap** command is an unsupported sample program that accepts bitmaps in either the XPM2 or XPM3 formats. It does not support pixmap editing. Be sure that the Bitmap app-defaults file has been installed in the **/usr/lib/X11/app-defaults** directory before invoking the **bitmap** command. If not, issue the following command in the **/usr/lpp/X11/Xamples/programs/bitmap** directory:

```
xmkmf;
make install
```

The following object names (and their class names) can be used to customize this tool:

```
custom (Custom)
  startupDialog_popup (XmDialogShell)
     startupDialog (XmSelectionBox)
  helpDialog_popup (XmDialogShell)
     helpDialog (XmForm)
  saveDialog_popup (XmDialogShell)
     saveDialog (XmSelectionBox)
  colorEditor_popup (XmDialogShell)
     colorEditor (XibmColorEditor)
  fontEditor_popup (XmDialogShell)
     fontEditor (XibmFontEditor)
  pictureEditor_popup (XmDialogShell)
     pictureEditor (XibmPictureEditor)
  cursorEditor_popup (XmDialogShell)
     cursorEditor (XibmCursorEditor)
  selectmanyEditor_popup (XmDialogShell)
     selectmanyEditor (XibmSelectManyEditor)
  filenameEditor_popup (XmDialogShell)
     filenameEditor (XmFileSelectionBox)
  mainWindow (XmMainWindow)
     menubar (XmRowColumn)
     form (XmForm)
        appClassLabel (XmLabel)
        appClass (XmLabel)
        groupMenuLabel (XmLabel)
        groupMenu (XmRowColumn)
        scrolledGroup (XmScrolledWindow)
           scrolledGroupForm (XmForm)
                (XmLabelGadget)
                TypeField (XmTextField)
                TypeButton (XmPushButton)
```

where *Type* can be one of the color, font, picture, cursor, selectmany, filename, selectone, string, or number data type values.

Exit Status

This command returns the following exit values:

0 Indicates successful completion.
>0 Indicates an error occurred.

Files

/usr/bin/X11	Is the path from which you run the **custom** command once the custom package is installed.
/usr/lib/X11/app-custom	Contains information about resources for individual applications.
/usr/lib/X11/*locale***/app-custom**	Contains information about resources for individual applications that is translated for specific locales.
/usr/lib/X11/app-defaults/Custom	Contains default settings for the Customizing Tool.
/usr/lib/X11/*locale***/app-defaults/Custom**	Contains default settings for the Customizing Tool in locales that require special settings.

Related Information

How to Start the Customizing Tool in *AIX 5L Version 5.3 AIXwindows Programming Guide*.

---

cut Command

Purpose

Helps split the lines of a file.

Syntax

cut { **-b** *List* [**-n**] | **-c** *List* | **-f** *List* [**-s**] [**-d** *Character*] } [*File* ...]

Description

The **cut** command cuts bytes, characters, or fields from each line of a file and writes these bytes, characters, or fields to standard output. If you do not specify the *File* parameter, the **cut** command reads standard input.

You must specify either the **-b**, **-c**, or **-f** flag. The *List* parameter is a comma-separated, blank-separated, or hyphen-separated list of integer numbers (in increasing order). The hyphen separator indicates ranges. The following entries are some example *List* parameters which could refer to bytes, characters, or fields:

```
1,4,7
1-3,8
-5,10
3-
```

where `-5` is a short form for the first through fifth and `3-` is a short form for the third through last.

If using the **cut** command on fields, the length of the fields specified by the *List* parameter can vary from field to field and line to line. The position of the field delimiter character, such as a tab character, determines the length of a field.

You can also use the **grep** command to make horizontal cuts through a file and the **paste** command to put the files back together. To change the order of columns in a file, use the **cut** and **paste** commands.

Flags

-b *List*	Specifies byte positions. These byte positions ignore multibyte character boundaries unless the **-n** flag is also specified.
-c *List*	Specifies character positions. For example, if you specify `-c 1-72`, the **cut** command writes out the first 72 characters in each line of the file.
-d *Character*	Uses the character specified by the *Character* variable as the field delimiter when you specify the **-f** flag. You must put quotation marks around characters with special meaning to the shell, such as the space character.
-f *List*	Specifies a list of fields assumed to be separated in the file by a delimiter character, which is by default the tab character. For example, if you specify `-f 1,7`, the **cut** command writes out only the first and seventh fields of each line. If a line contains no field delimiters, the **cut** command passes them through intact (useful for table subheadings), unless you specify the **-s** flag.
-n	Suppresses splitting of multibyte characters. Use only with the **-b** flag. If the last byte of a character falls within the range denoted by the *List* variable of the **-b** flag, the character is written; otherwise, the character is excluded.
-s	Suppresses lines that do not contain delimiter characters. Use only with the **-f** flag.

Exit Status

This command returns the following exit values:

0	All input files were output successfully.
>0	An error occurred.

Examples

1. To display several fields of each line of a file, enter:

   ```
   cut  -f 1,5 -d : /etc/passwd
   ```

   This displays the login name and full user name fields of the system password file. These are the first and fifth fields (`-f 1,5`) separated by colons (`-d :`).

   For example, if the **/etc/passwd** file looks like this:

   ```
   su:*:0:0:User with special privileges:/:/usr/bin/sh
   daemon:*:1:1::/etc:
   bin:*:2:2::/usr/bin:
   sys:*:3:3::/usr/src:
   adm:*:4:4:System Administrator:/var/adm:/usr/bin/sh
   pierre:*:200:200:Pierre Harper:/home/pierre:/usr/bin/sh
   joan:*:202:200:Joan Brown:/home/joan:/usr/bin/sh
   ```

   The **cut** command produces:

   ```
   su:User with special privileges
   daemon:
   bin:
   sys:
   adm:System Administrator
   pierre:Pierre Harper
   joan:Joan Brown
   ```

2. To display fields using a blank separated list, enter:

   ```
   cut -f "1 2 3" -d : /etc/passwd
   ```

   The **cut** command produces:

```
su:*:0
daemon:*:1
bin:*:2
sys:*:3
adm:*:4
pierre:*:200
joan:*:202
```

Files

/usr/bin/cut Contains the **cut** command.

Related Information

The **grep** command, **paste** command, **sh** command.

Files in *Operating system and device management* describes what files are and how they are stored by the operating system.

Input and output redirection overview in *Operating system and device management* describes how to redirect input and output.

---

cw or checkcw Command

Purpose

Prepares constant-width text for the **troff** command.

Syntax

cw [**+t** | **-t**] [**-d**] [**-f** *Font*] [**-l** *Delimiter*] [**-r** *Delimiter*] [*File* ...]

checkcw [**-l** *Delimiter*] [**-r** *Delimiter*] [*File* ...]

Description

The **cw** command preprocesses any specified **troff** files containing English-language text to be typeset in the constant-width (CW*)* font. The **cw** command reads standard input if you do not specify a file or if you specify a **-** (minus sign) as one of the input file names. The **cw** command writes to standard output.

Because output resulting from this command resembles the output of line printers and workstations, use this command to typeset examples of programs and computer output for user manuals and programming text. The **cw** command produces distinctive output when used with the Times Roman font.

The CW font contains a nonstandard set of characters. Any text typeset with this font requires different character and interword spacing from that used for standard fonts. Therefore, you must use the **cw** command to preprocess documents that use the CW font.

The CW font contains the following 94 ASCII printing characters:

```
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789
!$%&()`'*+@.,/:;=?[]|-_^~"<>{}#\
```

This font also contains 11 non-ASCII characters represented by 4-character **troff** strings (in some cases attaching these strings to nonstandard graphics).

The **cw** command recognizes five request lines as well as user-defined delimiters. The request lines look like **troff** macro requests. The **cw** command copies them in their entirety onto the output. Thus, you can define the requests as **troff** macros; in fact, the **.CW** and **.CN** macros should be so defined. The five requests are:

.CW	Marks the start of text to be set in the CW font. This request causes a break. It can take the same flags (in the same format) as those available on the **cw** command line.
.CN	Marks the end of text to be set in the CW font. This request causes a break. It can take the same flags (in the same format) as those available on the **cw** command line.
.CD	Changes the delimiters and settings of other flags. It can take the same flags (in the same format) as those available on the **cw** command line. The purpose of this request is to allow the changing of flags other than at the beginning of a document.
.CP *Option-list*	Concatenates all the options (delimited like **troff** macro options), with the odd-numbered options set in the CW font and the even-numbered options set in the prevailing font.
.PC *Option-list*	Acts the same as the **.CP** macro, except the even-numbered options are set in CW font and the odd-numbered options are set in the prevailing font.

The **.CW** and **.CN** requests should bracket text that is to be typeset as is, using the CW font. Normally, the **cw** command operates in the transparent mode. In that mode, every character between **.CW** and **.CN** request lines represents itself, except for the **.CD** request and the special 4-character names listed previously. In particular, the **cw** command causes all **.** (periods) and **'** (apostrophes) at the beginning of lines, and all **** (backslashes) and ligatures (such as fi and ff), to be hidden from the **troff** command. The transparent mode can be turned off by using the **-t** flag, in which case normal **troff** rules apply. In either case, the **cw** command hides from the user the effect of the font changes generated by the **.CW** and **.CN** requests.

You can also use the **-l** and **-r** flags to define delimiters with the same function as the **.CW** and **.CN** requests. These requests are meant to enclose words or phrases that are set in CW font in the running text. The **cw** command treats text between delimiters as it does text bracketed by **.CW**/**.CN** pairs, with one exception. Spaces within **.CW**/**.CN** pairs, have the same width as other CW characters, while spaces within delimited text are half as wide, so they have the same width as spaces in the prevailing text. Delimiters have no special meaning inside **.CW**/**.CN** pairs.

The **checkcw** command checks that left and right delimiters as well as the **.CW**/**.CN** pairs are properly balanced. It prints out all lines in the selection with the unmatched delimiters.

Notes:

1. The **.** (period) or **** (backslash)delimiter characters should not be used.
2. Certain CW characters do not combine well with certain Times Roman characters; for example, the spacing between a CW **&** (ampersand) followed by a Times Roman **,** (comma). In such cases, using **troff** half- and quarter-space requests can help.
3. The **troff** code produced by the **cw** command is difficult to read.
4. The **mm** macro package and **mv** macro package contain definitions of **.CW** and **.CN** macros that are adequate for most users. If you define your own macros, make sure that the **.CW** macro starts the **troff** no-fill (**.nf**) mode, and the **.CN** macro restores the fill mode (**.fi**), if appropriate.
5. When set in running text, the CW font is meant to be set in the same point size as the rest of the text. In displayed matter, on the other hand, it can often be profitably set 1 point smaller than the prevailing point size. The CW font is sized so that, when it is set in 9-point, there are 12 characters per column inch.
6. Documents that contain CW text can also contain tables and equations. In this case, the order of preprocessing must be the **cw** command, **tbl** command, and **eqn** command. Usually, the tables do not contain CW text, although it is possible to have elements in the table set in the CW font.

Ensure that the **cw** command does not modify the **tbl** command format information. Attempts to set equations in the CW font are usually unsuccessful.

7. In the CW font, overstriking is most easily accomplished with backspaces. Because spaces (and therefore backspaces) are half as wide between delimiters as inside **.CW**/**.CN** pairs, two backspaces are required for each overstrike between delimiters.
8. Some devices such as the IBM 3816 Pageprinter do not have a CW font. You receive a `troff can't open /usr/lib/font/devNAME/CW.out` message for these devices. The **troff** command uses the font in font position 3 as the CW font.

Parameters

File Specifies **troff** English-language text files to be preprocessed by the **cw** command to produce constant-width characters in the output file.

File Specifies **troff** English-language text files to be preprocessed by the **checkcw** command to check right and left delimiters as well as **.CW** and **.CN** pair balance.

Flags

+t Turns the transparent mode on (this is the default).

-t Turns the transparent mode off.

-d Displays the current flag settings on the standard error output in the form of **troff** comment lines. This flag is meant for debugging.

-f*Font* Replaces the value of the *Font* variable with the **cw** command font (the default equals 3, which replaces the bold font). The **-f5** flag is commonly used for formatters that allow more than four simultaneous fonts.

Note: This flag is useful only on the command line.

-l*Delimiter* Sets the left delimiter as the 1- or 2-character string specified by the *Delimiter* variable. The left delimiter is undefined by default.

-r*Delimiter* Sets the right delimiter to that specified by the *Delimiter* variable. The right delimiter is undefined by default. The left and right delimiters can (but need not) be different.

Related Information

The **eqn** command, **mmt** command, **tbl** command, **troff** command.

The **mm** macro package, **mv** macro package.

---

cxref Command

Purpose

Creates a C and C++ program cross-reference listing.

Syntax

cxref [**-c**] [**-o** F*ile*] [**-q***Option*] [**-s**] [**-t**] [**-w** *Number*] [[**-D** *Name* [*=Definition*]] [**-I** *Directory*] [**-U** *Name*]] ... [**-Nd***Number*] [**-Nl***Number*] [**-Nn***Number*] [**-Nt***Number*] *File* ...

Description

The **cxref** command analyzes C and C++ program *File*s and creates a cross-reference table, using the **cpp** command to include **#define** directives in its symbol table. It writes to standard output a listing of all symbols in each file processed, either separately or in combination (see the **-c** flag). The formal

parameters in a function definition are always listed; but if a function is only prototyped and not defined, the parameters are not listed. When a reference to a symbol is that symbol's declaration, an * (asterisk) precedes it.

Flags

-c	Displays a combined listing of the cross-references in all input files.
-o *File*	Directs the output to the specified *File*.
-s	Does not display the input file names.
-t	Makes the listing 80 columns wide.
-w *Number*	Makes the listing *Number* columns wide, where *Number* is a decimal integer greater than or equal to 51. If *Number* is less than 51, the listing will be 80 columns wide.
-**Nd***Number*	Changes the dimension table size to *Number*. The default is 2000.
-**Nl***Number*	Changes the number of type nodes to *Number*. The default is 8000.
-**Nn***Number*	Changes the symbol table size to *Number*. The default is 1500.
-**Nt***Number*	Changes the number of tree nodes to *Number*. The default is 1000.

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

-D *Name*[=*Definition*]	Defines *Name* as in a **#define** directive. The default definition is 1.
-I *Directory*	Looks first in directory, then looks in the directories on the standard list for **#include** files with names that do not begin with a slash (/) (see the **cpp** command).
-U *Name*	Removes any initial definition of *Name*, where *Name* is a reserved symbol predefined by the preprocessor.
-q*Option*	Pass **-q***Option* to the preprocessor. For example, `-qmbcs` sets multibyte mode specified by the current locale, and `-qidirfirst` modifies the search order for files included with the **#include file_name** directive.

Examples

To provide a combined cross-reference listing of `stdin1.c` and `stdin2.c`, making the output 80 columns wide, enter:

```
cxref  -c  -t stdin1.c stdin2.c > output
```

Files

/usr/ccs/lib/xpass	Special version of C compiler first-pass.
/usr/ccs/bin/cxref	Contains the **cxref** command.

Related Information

The **cpp** command.

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

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 give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: 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 states 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.

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 Corporation
Dept. LRAS/Bldg. 003
11400 Burnet Road
Austin, TX 78758-3498
U.S.A.

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.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

© Copyright IBM Corp. 1997, 2009

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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 the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates 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. 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 IBM's application programming interfaces.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Trademarks

IBM, the IBM logo, and 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

UNIX is a registered trademark of The Open Group in the United States and other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be the trademarks or service marks of others.

Index

Special characters

/etc/filesystems file
  changing entries
    using chfs command 380
/etc/inittab file
  changing records
    using chitab command 401
/etc/vfs file
  changing entries 521
  creating entries 618
.Bidi-defaults keywords 287
.Xdefaults file
  overriding 79

A

ac command 1
access control
  displaying information about 30
  editing 28
  setting for a file 32
accounting
  acctctl command 10
accounting commands
  checking the size of data files 540
accounting system
  charging the users 345
  connect-time accounting 8
  connect-time records
    printing 1
  disk-usage accounting 14
  merging records 16
  merging records into a daily report 16
  process accounting
    using acctcon command 19
    using acctprc1 command 19
    using acctprc2 command 19
  process accounting summaries
    displaying 4
  summaries
    producing 2
  writing utmp records 26
acct/* commands
  chargefee 345
  ckpacct 540
acctcms command 2
acctcom command 4
acctcon command 8
acctctl command 10
acctdisk command 14
acctdusg command 14
acctmerg command 16
accton command 19
acctprc1 command 19
acctprc2 command 19
acctrpt command 20
acctwtmp command 26
aclconvert command 27
acledit command 28
aclget command 30
aclgettypes command 31
aclput command 32
adb command 34
addbib command 36
address resolution protocol 132
addrpnode command 38
addX11input command 40
adfutil command 40
admin command 42
administer disk space 342
aixmibd daemon 47
aixpert command 49
aixpertldap 51
aixterm command 53
  areas 54
  colors
    display 54
  COPY button function 73
  datastream support 57
  escape sequences 54
  menus
    categories 74
  PASTE button function 74
  RE-Execute button function 74
  setting the defaults 85
  WINDOWID environment variable 54
ali command 90
alias command 91
alias conflicts
  searching for (MH) 570
aliases
  defining or displaying 91
alog command 93
alt_disk_copy command 97
alt_disk_install command 101
alt_disk_mksysb command 108
alt_rootvg_op command 112
alter subcommand for the ate command 155
alternate disk, install 101
amepat command 115
anno command 121
ap command 124
apply command 125
ar command 127
arguments
  applying a command to 125
arithmetic
  providing interpreters for arbitrary precision 228
arp command 132
artexdiff command 135
artexget command 136
artexlist command 138
artexmerge command 139
artexremset command 140
artexset command 142

© Copyright IBM Corp. 1997, 2009

as command 144
ASCII characters
  writing strings to standard output 224
Asynchronous Terminal Emulation program 155
at command 149
  removing jobs spooled by the 169
at jobs
  listing 613
  removing 613
ate command 154
  subcommands
    alter 155
    break 159
    connect 159
    directory 160
    help 161
    modify 161
    perform 164
    quit 165
    receive 165
    send 165
    terminate 165
ATE program 155
  alter subcommand 155
  at command 149
  break subcommand 159
  connect subcommand 159
  connecting to a remote computer 159
  directory subcommand 160
  displaying the dialing directory 155
  exiting 165
  help subcommand 161
  interrupting remote activity 155
  issuing commands while using 164
  modify subcommand 161
  modifying terminal emulation settings 161
  perform subcommand 164
  providing help for 161
  quit subcommand 165
  receive subcommand 165
  send subcommand 165
  sending a file 155
  starting 154
  terminate subcommand 165
ATM (Asyncronous Transfer Mode) interfaces
  displaying and modifying address resolution
    using arp command 132
atq command 168
atrm command 169
attachrset 170
audit command 171
audit records 178
  formatting 181
  processing 177
  reading 188
  selecting for analysis 183
auditbin daemon 175
auditcat command 177
auditconv command 178
auditing system
  controlling 171
auditing system *(continued)*
  managing bins of information 175
  processing audit records 177
  reading audit records 188
  selecting audit records for analysis 183
auditmerge
  multiple audit trails 180
auditpr command 181
auditselect command 183
auditstream command 188
authexec command 190
authqry command 193
authrpt command 191
autoconf6 command 194
automount daemon 195
autopush command 198
awk command 200

B

background jobs 251
backsnap command 217
backup command 218
backup format
  creating files in 245
base file names
  displaying 224
batch command 226
battery command 227
bdftopcf command 240
bdiff command 241
bellmail command 242
bffcreate command 245
bg command 251
bibliographic database
  creating 36
  extending 36
bicheck command 253
BIDI mode
  emulating terminals
    using bterm command 285
bidirectional mode
  emulating terminals
    using bterm command 285
biff command 253
bin files
  managing 175
bindprocessor command 256
biod daemon (NFS) 259
BNU
  communicating with another workstation 627
  connecting to another system 680
boot image
  creating
    using bosboot command 269
bootauth 261
bootlist command 261
bootparamd daemon 266
bootpd daemon 266
bootptodhcp command 268
bosboot command 269

bosdebug command 272
Bourne shell
  invoking 283
bsh command 283
bterm command 285
bug reports, mail
  storing 288
bugfiler command
  Mail 288
burst command 291

C

C programming language
  performing file inclusion 594
  reading from standard input 309
C shell
  invoking 619
cache contents 342
cachefsstat command 294
cancel command 299
canonls command 302
captoinfo command 303
cdcheck command 313
cdeject command 315
cdmount command 316
cdromd command 317
cdumount command 318
cdutil command 319
certadd command 320
certcreate command 322
certdelete command 324
certget command 325
certlink command 326
certlist command 328
certrevoke command 330
certverify command 331
cfadmin command 342
cfgif method 333
cfginet method 334
cfgmgr command 335
cflow command 340
change filters 375
change output device 528
change time zone 509
change tunnel definition 506
character classes 75
character strings
  writing in large letters 224
charClass resource
  default table 75
chargefee command 345
chauth command 345
chauthent command
  authentication methods
    changing 347
chC2admin command 348
chCCadmin command 349
chcifscred command 350
chcifsmnt command 351
chclass command 352
chcomg command 357
chcondition command 361
chcons command
  description of 365
chcore command 367
chcosi command 368
chdev command 370
chdisp command 373
checkcw command 701
checkeq command 374
checkmm command 374
checknr command 374
chfilt command 375
chfn command 377
chfont command 379
chfs command 380
chgif method 388
chginet method 390
chgroup command 391
chgrp command 394
chgrpmem command 396
chhbd command 403
chhwkbd command 398
chiscsi command 399
chitab command 401
chkey command 404
chlang command 404
chlicense command 407
chlpclacl command 408
chlpcmd command 412
chlpracl command 415
chlpriacl command 420
chlprsacl command 424
chlv command 429
chmaster command 434
chmod command 436
chmp command 439
chnamsv command 441
chndaf 443
chnfs command 446
chnfsdom 448
chnfsexp command 448
chnfsim command 452
chnfsmnt command 456
chnfsrtd 458
chnfssec 459
chnlspath command 445
chown command 461
chpasswd 462
chpath command 464
chprtsv command 466
chps command 469
chpv command 471
chquedev command 474
chresponse command 476
chrole command 480
chroot command 482
chrsrc command 484
chsec 488
chsensor command 491
chservices command 495

chsh command 497
chslave command 499
chsubserver command 503
chtcb command 505
chtun command 506
chtz command 509
chuser command 509
chusil command 521
chvfs command 521
chvg command 522
chvirprt command 527
chvmode command 528
chwpar command 529
chypdom command 536
ckauth command 537
ckfilt command 537
ckpacct command 540
ckprereq command 541
cksum command 543
classes
selection 75
clogin command 545
clsnmp command 546
cmp command 552
col command 554
colcrt command 555
colrm command 556
columns
extracting from a file 556
comm command 558
command command 560
command path names 560
command usage summaries 2
commands
addrpnode 38
arp 132
as 144
at 149
attachrset 170
bosboot 269
bterm 285
chcomg 357
chcondition 361
chfs 380
chlpclacl 408
chlpcmd 412
chlpracl 415
chlpriacl 420
chlprsacl 424
chnfsexp 448
chresponse 476
chrsrc 484
chsensor 491
chwpar 529
cplv 592
csmstat 621
ctaclfck 629
cthactrl 642
cthagsctrl 643
cthagstune 646
cthatsctrl 647
commands *(continued)*
cthatstune 650
ctlvsd 652
ctmsskf 654
ctscachgen 657
ctsidmck 660
ctskeygen 663
ctsnap 665
ctsthl 667
custom 693
running automatically 611
suppressing shell function lookup 560
commands, running
at a later time
using at command 149
comp command 562
compare_report command 564
compress command 567
comsat command
Mail 569
Configuration Database
merging PS/2 adapter information with 40
configuration file
manipulating 503
configure devices 335
configure IPv6 network 194
conflict command 570
confsetcntrl command 571
confsrc command 576
connect subcommand for the ate command 159
connect-time records 1
convert audit records 178
copying contents of
logical volume
using cplv command 592
cp command 578
cpcosi command 582
cpio command 583
cplv command 592
cpupstat command 598
creating
boot image
using bosboot command 269
crfs command 606
cron daemon 611
cron job files
listing 614
removing 614
submitting 614
cronadm command 613
crontab command 614
crontab jobs
listing 613
removing 613
crvfs command 618
csh command 619
csmstat command 621
csplit command 623
csum command 625
ct command 627
ctaclfck command 629

ctags command 633
ctcasd daemon 635
ctctrl command 636
cthactrl command 642
cthagsctrl command 643
cthagstune command 646
cthatsctrl command 647
cthatstune command 650
ctlvsd command 652
ctmsskf command 654
ctscachgen command 657
ctsidmck command 660
ctskeygen command 663
ctsnap command 665
ctsthl command 667
cu command
  description of 680
curt 684
custom command 693
customized devices object class 370
customizing tool
  starting
    using custom command 693
cut command 699
cw command 701
cxref command 703

D

daemons
  bootpd 266
  ctcasd 635
debug program 34
delete cache 342
delta files
  changing comments 311
  combining 557
Device Configuration Database
  configuring all devices 335
devices
  changing characteristics in 370
  configuration commands
    bootlist 261
devices, configure 335
dialing directory
  establishing a connection with an entry from a 160
digests
  exploding into messages 291
directories
  changing 309
  moving between 309
directory
  changing the group ownership of
    using chgrp command 394
  changing the root 482
directory subcommand for the ate command 160
disk space 342
display
  changing for a low function terminal
    using chdisp command 373
dynamic host configuration protocol
  convert bootp file into a dhcp file
    bootptodhcp command 268
  remove bootp information from a dhcp file
    bootptodhcp command 268

E

editing bitmaps and pixmaps
  picture editor 698
emulates terminals
  in BIDI mode
    using bterm command 285
exporting a directory
  to NFS clients
    using chnfsexp command 448

F

file
  backing up 218
  changing the group ownership of
    using chgrp command 394
  changing the user associated with the 461
  copying into and out of archive storage 583
  copying into and out of directories 583
  displaying access control information of a 30
  displaying block count
    using cksum command 543
  displaying the checksum
    using cksum command 543
  editing the access control information of a 28
  extracting columns from a 556
  select or reject common lines 558
  setting the access control information of a 32
file names
  displaying base 224
file system
  creating 606
files
  comparing
    two 552
  compression 567, 569
  concatenating 305
  copying
    description of 578
  creating
    backup format 245
    cross-refernce tables 703
    preformatted versions 307
  cutting selected fields from 699
  displaying 305
  finding
    differences in large 241
  joining 305
  printing FORTRAN 148
  reading 248
  reading and assembling source
    using as command 144
  receiving from a remote system 155
  scanning 248

files *(continued)*
SCCS
controlling 42
creating 42
sending to a remote computer 155
splitting by context 623
tracking external references 340
files modes
changing 436
filters, change 375
font
changing the default font
using chfont command 379
fonts
converting 240

G

games
arithmetic skills test 130
backgammon 216
blackjack 260
craps 599
group services
control commands
cthagsctrl 643
tuning 646
groups
changing attributes for
using the chgroup command 391
changing the administrators of
using chgrpmem command 396
changing the members of
using chgrpmem command 396

H

help subcommand for the ate command 161

I

incoming mail
notifying users of 569
input extension record
adding 40
install a mksysb image 101
install an alternate disk 101, 335
instantaneous resources
updating 697
Internet Boot Protocol server
implementing 266

J

job control 251

K

keyboard
changing attributes
using chhwkbd command 398
changing the alarm volume
using chhwkbd command 398
changing the clicker volume
using chhwkbd command 398
changing the delay of the keys on
using chhwkbd command 398
changing the repetition rates of
using chhwkbd command 398
enabling/disabling Korean keyboard
using chhwkbd command 398
keyboard map
changing for the Low Function Terminal Subsystem
using chkbd command 403
keys
rebinding 76
keywords
.Bidi-defaults 287

L

language setting 404
libraries
maintaining indexed 127
licenses, change 407
Light Directory Access Protocol (LDAP) 51
line printer
canceling requests to a 299
linefeeds
filtering for output 554
list cache contents 342
locate objects 633
log files
create and maintain 93
logical volume
changing the characteristics 429
copying contents of
using cplv command 592
Low Function Terminal Subsystem
changing the default display for
using chdisp command 373
changing the default keyboard map
using chkbd command 403

M

mail
disabling notification 253
enabling notification 253
listing addressing for aliases 90
mail address
parsing and reformating (MH) 124
Mail commands
bugfiler 288
comsat 569
man pages
keyword searches 126

memorandum macro
checking document formatted with
using checkeq command 374
using checkmm command 374
messages
annotating 121
clearing from the screen 544
composing 562
displaying from system users 242
sending
to system users 242
MH
ap command 124
conflict command 570
mirror pools 429, 439, 471
Modes menu
description 74
modify subcommand for the ate command 161
MultiPath I/O
chpath command 464

N

network config
autoconf6 194
NFS clients
changes options to export directory
using chnfsexp command 448
NFS commands
chnfs 446
chnfsexp 448
chnfsmnt 456
NFS daemons
automount 195
biod 259
bootparamd 266
NIS commands
chkey 404
chmaster 434
chslave 499
chypdom 536
NLSPATH,
secure,
setting 445
nroff command
filtering output for CRT previewing 555
nroff file
checking
using checknr command 374
number of licenses 407

O

Options menu
description 74
output device, change 528

P

paging space
changing the attributes of 469
password
conflicts
searching for (MH) 570
pattern matching 200
perform subcommand for the ate command 164
picture editor
editing bitmaps and pixmaps 698
plotter queue
changing the name of 474
printer
line
canceling requests to 299
printer queue
changing the name of 474
process troff output 302
programs
compiling and interpreting 274
pseudo terminal
creating 54

Q

queue
changing the name of 473
displaying the jobs to be run 168
quit subcommand for the ate command 165

R

RBAC
Role-Based Access Control
using authexec command 190
receive subcommand for the ate command 165
reconfigure
Internet instance
TCP/IP 390
Reliable Scalable Cluster Technology (RSCT)
commands
cthactrl 642
remote computer
connecting through ATE program 159
remote system
interrupting current activity on 155
receiving a file from 155
sending a file to 155
terminating an ATE connection 155
revision levels
verifying availability of software at the
appropriate 541
Role Based Access Control (RBAC) 345, 480
running commands
at a later time
using at command 149

S

SCCS
delta files
changing comments 311
combining 557

SCCS *(continued)*
files
controlling 42
creating 42
SCCS commands
admin 42
cdc 311
comb 557
screen
capturing displays 304
clearing 544
printing messages 224
printing to a file 304
scrollbar
description 74
security files, change 488
send subcommand for the ate command 165
services file
manipulating 495
shells
Bourne 283
C 619
smit command 466
SNMP Enterprise MIB sub-agent 47
Source Code Control System 42
source files
reading and assembling
using as command 144
source files, locate objects 633
SRC
modifying subserver object definition 494
changing owning subsystem example 495
changing subserver type example 495
modifying subsystem object definition 500
changing communication type examples 502
changing subsystem name example 502
SRC configuration commands
chserver 494
chssys 500
standard command-line options 693
standard input
capturing screen dumps 304
status information, system 272
STREAMS commands
autopush 198
STREAMS facility
modules
configuring list 198
subservers
modifying SRC object definition 494
changing owning subsystem example 495
changing subserver type example 495
subsystem
control commands
cthactrl 642
cthagsctrl 643
cthatsctrl 647
group services
tuning 646
topology services
tuning 650
subsystems
modifying SRC object definition 500
changing communication type examples 502
changing subsystem name example 502
syntax checker 253
system
connecting to another system 680
system boot
boot devices
list of 261
system console
redirecting to a file 365
redirecting to a specified device 365
system load level
running jobs when permitted by 226
system resource controller 494, 500
system status information 272

T

TCP/IP
displaying and modifying address resolution
using arp command 132
instances
activating 333
configuring 333
loading and configuring 334
modifying 388
methods
chginet 390
name service
changing configuration of 441
print services
changing configuration of 466
service management 495, 503
TCP/IP commands
arp 132
chnamsv 441
chprtsv 466
TCP/IP methods
cfgif 333
cfginet 334
chgif 388
TCP/IP smit commands
chnamsv 441
chprtsv 466
termcap file
converting to terminfo entries 303
terminal emulation
HFT default 54
initializing 53
terminals
clearing the screen 544
dialing an attached 627
terminate subcommand for the ate command 165
terminfo descriptor files
conversion from termcap file 303
text
filtering forward and reverse half-linefeeds for output 554
filtering reverse linefeeds to standard output 554

time management
displaying calendars 296
writing reminder messages 297
time zone, change 509
topology services
control commands
cthatsctrl 647
tuning 650
translation bindings
actions available 76
default values 77
troff command
preparing
using checkcw command 701
using cw command 701
troff file
checking
using checknr command 374
troff output, process 302
trusted computing base attribute
changing 505
querying 505
tuning
group services 646
topology services 650
tunnel definition 506

U

updating
instantaneous resources 697
user
changing gecos information for
using chfn command 377
changing login shell 497
changing the file 461
users
changing role attributes for 480
changing the attributes for 509

V

virtual printer
changing the attribute values of 527
volume group
changing a physical volume characteristics 471
setting the characteristics 522

W

Workload Manager (WLM)
confsetcntrl command 571
managing time-based configurations 571
workload partition (WPAR) 4, 10, 20, 171, 181, 294, 342, 529, 545, 684

X

X applications
customizing tool for
using custom command 693

Readers' Comments — We'd Like to Hear from You

AIX Version 6.1
Commands Reference, Volume 1, a - c

Publication No. SC23-5243-02

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send your comments via e-mail to: pserinfo@us.ibm.com

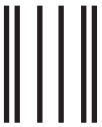
If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	E-mail address

Readers' Comments — We'd Like to Hear from You
SC23-5243-02

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Information Development
Department 04XA-905-6B013
11501 Burnet Road
Austin, TX 78758-3400

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.

SC23-5243-02